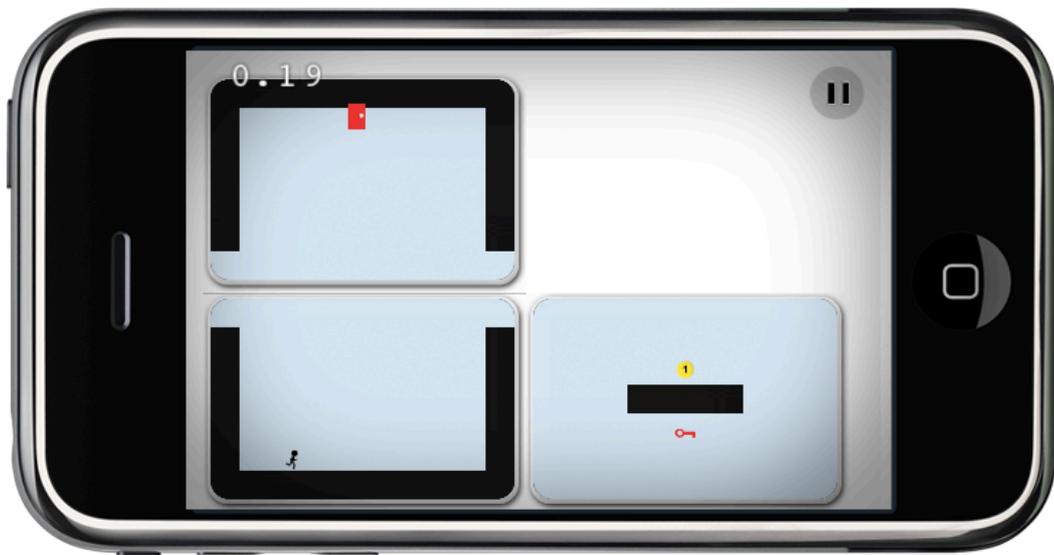


# CHALMERS



## Continuity 2: The Creation of an iPhone Sequel to the Puzzle Platformer Continuity

*Master of Science Thesis in the Programme Interaction Design*

GUY LIMA, JR.  
STEFAN MIKAELSSON

Chalmers University of Technology  
University of Gothenburg  
Department of Computer Science and Engineering  
Göteborg, Sweden, October 2011

The Authors grant to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Authors shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Authors have signed a copyright agreement with a third party regarding the Work, the Authors warrant hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

## Continuity 2: The Creation of an iPhone Sequel to the Puzzle Platformer Continuity

Guy Lima, Jr.  
Stefan Mikaelsson

© Guy Lima, Jr. and Stefan Mikaelsson, October 2011.

Examiner: Staffan Björk

Chalmers University of Technology  
University of Gothenburg  
Department of Computer Science and Engineering  
SE-412 96 Göteborg  
Sweden  
Telephone + 46 (0)31-772 1000

Cover:  
The level "TiltTest4" from Continuity 2.

Department of Computer Science and Engineering  
Göteborg, Sweden October 2011

## **Abstract**

This report presents the design process of the iPhone game Continuity 2, a sequel to the PC puzzle platformer Continuity. The primary goals of this project were to create a sequel that felt tailor-made for iPhone and familiar to, but distinct from, the original. A major focus of the game's development was the design and evaluation of tacit controls, a character control scheme that uses no visual interface elements. The result of this work is a completed design and a feature complete implementation of an iPhone game.

**keywords:** game design, mobile gaming, user interface design

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem . . . . .	2
1.2	Scope . . . . .	3
<b>2</b>	<b>Background</b>	<b>4</b>
2.1	Continuity: The Sliding-tile Puzzle Platformer . . . . .	5
2.1.1	Game Mechanics . . . . .	5
2.1.2	Teaching the Rules of the Game . . . . .	7
2.1.3	Level Progression . . . . .	7
2.1.4	Audio . . . . .	8
2.1.5	Controls . . . . .	9
2.1.6	Release . . . . .	10
2.2	The iPhone Platform . . . . .	11
2.2.1	The Smartphone iPhone . . . . .	11
2.2.2	The Portable Media Player iPod Touch . . . . .	11
2.2.3	The Tablet Computer iPad . . . . .	12
2.3	Casual Gaming . . . . .	13
2.4	Mobile Gaming . . . . .	14
2.5	Character Movement in iPhone Games . . . . .	15
<b>3</b>	<b>Methodology</b>	<b>18</b>
3.1	Generating Ideas . . . . .	19
3.1.1	Brainstorming . . . . .	19
3.1.2	Removing Mental Blocks . . . . .	19
3.1.3	Stating Objectives . . . . .	20
3.1.4	Risk Assessment . . . . .	20
3.2	Market Research . . . . .	21
3.2.1	Questionnaire . . . . .	21
3.3	Prototyping . . . . .	23
3.4	Evaluation . . . . .	25
3.4.1	User Observation . . . . .	25
3.4.2	“Quick and dirty” Evaluation . . . . .	26
3.4.3	Think-Aloud Technique . . . . .	26

3.4.4	Cooperative Evaluation . . . . .	26
3.4.5	Fly-on-the-Wall (FOTW) . . . . .	27
3.4.6	Interviews . . . . .	27
<b>4</b>	<b>Planning Development</b>	<b>28</b>
4.1	Time Plan . . . . .	29
4.2	Iterative Development . . . . .	30
<b>5</b>	<b>Method</b>	<b>31</b>
5.1	Defining Continuity . . . . .	32
5.1.1	Segmented Platformer . . . . .	33
5.1.2	Minimalism and Clarity . . . . .	33
5.1.3	Rewards . . . . .	34
5.2	Assessing Risks and Building a Foundation . . . . .	36
5.2.1	Assessing Risks . . . . .	36
5.2.2	The Foundation . . . . .	37
5.3	Establishing Goals . . . . .	38
5.3.1	Distinct, But Familiar . . . . .	40
5.3.2	Smooth Play . . . . .	40
5.3.3	Interruptability . . . . .	41
5.4	Survey . . . . .	43
5.4.1	Motivation . . . . .	43
5.4.2	Conducting the Survey . . . . .	45
5.4.3	Survey Results and Conclusions . . . . .	45
5.5	Re-planning and Redefining the Project Goals . . . . .	48
5.6	Iteration 1: Touch-to-Stop and Tacit Controls . . . . .	50
5.6.1	Analyzing Continuity's Character Movement . . . . .	50
5.6.2	Gameplay Mechanics as Solutions . . . . .	53
5.6.3	Developing Two Versions . . . . .	61
5.6.4	Designing the Character Movement Systems . . . . .	61
5.6.5	Playtesting . . . . .	69
5.7	Iteration 2: One-mode and Tacit Controls . . . . .	72
5.7.1	Designing One-mode With Direct Controls . . . . .	72
5.7.2	Playtesting . . . . .	72
5.8	Iteration 3: Accelerometer-Based Rotation . . . . .	75
5.8.1	Levers, Wires and Gates . . . . .	75
5.8.2	Rotation Controls . . . . .	77
5.8.3	Teaching Character Movement . . . . .	77
5.8.4	Playtest . . . . .	77
5.9	Iteration 4: Take Home . . . . .	84
5.9.1	Achievement Times . . . . .	84
5.9.2	Level Selection . . . . .	85
5.9.3	Character Controls . . . . .	88
5.9.4	Rotation Controls and Spikes . . . . .	89

5.9.5	Playtest . . . . .	90
<b>6</b>	<b>Results</b>	<b>91</b>
6.1	Gameplay Design . . . . .	93
6.1.1	Keys and Coins . . . . .	93
6.1.2	Rotation . . . . .	93
6.1.3	Levers, Wires and Gates . . . . .	93
6.1.4	Spikes . . . . .	94
6.1.5	Learning Through Playing . . . . .	94
6.2	Out-of-Game Features . . . . .	97
6.2.1	Achievement Time . . . . .	97
6.2.2	Saving Level Progress . . . . .	98
6.3	Controls . . . . .	99
6.3.1	Micro-Mode . . . . .	99
6.3.2	Macro-Mode . . . . .	100
6.3.3	Rotation . . . . .	100
6.4	Software . . . . .	102
<b>7</b>	<b>Discussion</b>	<b>103</b>
7.1	Planning and Work Process . . . . .	104
7.2	Tailor-made for the iPhone . . . . .	106
7.3	Distinct but Familiar . . . . .	108
<b>8</b>	<b>Future work</b>	<b>110</b>
<b>9</b>	<b>Conclusions</b>	<b>112</b>
<b>A</b>	<b>Glossary of Terms</b>	<b>117</b>

# Chapter 1

## Introduction

This report concerns the design and implementation of a videogame, named Continuity 2, for the iPhone mobile device. Continuity 2 is a sequel to the puzzle platformer Continuity, a game that combined mechanics of a 15-puzzle game and a platformer game. The original Continuity was developed in the autumn of 2009 by the authors and two other students. Continuity was released in November 2009 and was generally well-received by both reviewers and players. The game, which is hosted online, has received over 5 million visits. The creation of a sequel was motivated by feedback from many players of the original that communicated their desire for a mobile version of Continuity.

The goal of the project was to create a complete game that would be ready for distribution. Continuity 2 was meant to be recognizable to those that played Continuity as a continuation of the original, rather than an entirely new game. At the same time, the sequel was meant to contain new aspects that would appeal to all players, regardless of whether they were familiar or unfamiliar with the original. Another goal for the project was that Continuity 2 should feel as if it was tailor-made for the iPhone.

The original Continuity was designed and developed to be played using the Adobe Flash Player on a personal computer using a keyboard and monitor. Continuity 2, on the other hand, is played on the iPhone, a mobile device which possesses a multi-touch display and lacks a physical keyboard. Developing gameplay around a different interface was one the main design challenges of the sequel's development. For Continuity 2, a character control system was developed, called "tacit controls", in which the player controls a character without the aid of a graphical user interface.

Work was carried out from January through May, 2010 with much of the work devoted to developing the control scheme. The project resulted in a feature-complete game that marries gameplay mechanics with the iPhone interface. However, there is some work left to be completed before the game will be distributed.

## 1.1 Problem

The overall goal of this project was to create a complete game. Unfortunately, there is no objective measure by which to establish when a game is complete. For this work, the game could be considered complete based on playtester feedback. The game's pacing must be such that playtesters find the game appropriately challenging as they progress. The playtesters should experience Csikszentmihalyi's concept of flow[21], in that the game should be neither too challenging or too simple. Additionally, to be considered complete, the game must not contain any placeholder graphics or sounds and must be generally lacking in software bugs.

There were two primary challenges involved with creating such a complete game. First, the same design solutions that worked well for Continuity would not necessarily work well for Continuity 2, due to the differences between the methods of interaction used on a personal computer and an iPhone. Second, there was a challenge of creating a follow-up to a well-received game that appealed both to returning players and new players.

The problem with switching computing platforms involves accounting for different ways of interacting with the software via a different interface. The original Continuity was designed for a larger, higher resolution display than the iPhone possesses. The iPhone also uses a multi-touch display in place of the monitor, keyboard and mouse used by a PC. Moreover, the iPhone offers a set of other features, such as an accelerometer and digital compass, that have not been traditionally available on PCs. These capabilities and limitations pose both challenges and opportunities for designing intuitive gameplay.

The challenge of creating a sequel concerned exploring what characteristics constitute the core values and mechanics of the original game. A sequel that was too different may alienate those that liked the original. However, if the sequel did not change enough about the original it might feel repetitive to players of the original. Designing a sequel requires finding and developing additional solutions and modifications to the original game's design that results in a balance between the novel and the familiar.

## 1.2 Scope

The primary goal of this project was to create a game that both first-time and returning players enjoyed. This goal was not concerned with the game's marketability or commercial success. Therefore, the business and marketing of Continuity 2 or videogames in general is not discussed.

Furthermore, interaction and game design are the primary focuses of this work. While the importance of narrative, visual and audio aesthetics should not be underestimated as tools to create an engaging gaming experience, these aspects will not be presented.

As the goal of this project is to produce a piece of software, the work consists of both design and implementation. However, the technical and implementation decisions we made had little effect on the design of Continuity 2. As a result, this report does not concern itself with software development or engineering.

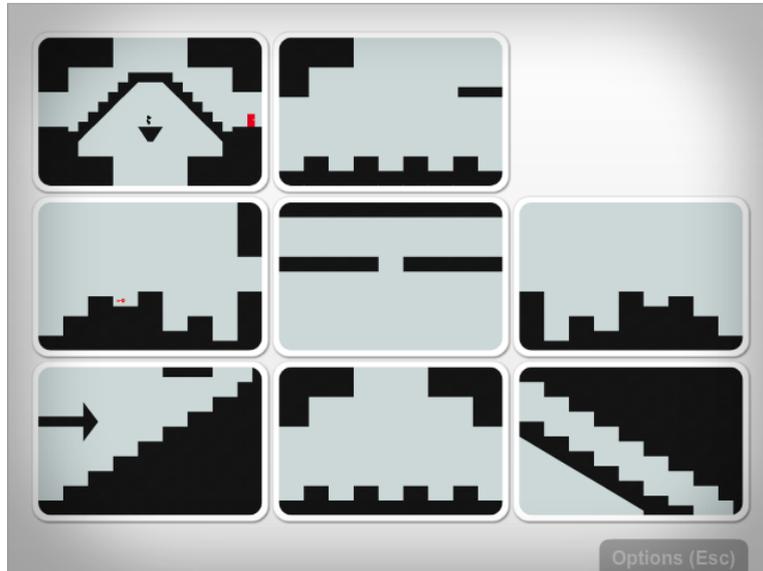
A final important limitation of this report regards the target platform. At the time of thesis initialization the most recent iOS device was the iPhone 3GS and second generation iPod Touch. Neither iPad nor iPhone 4 had been announced nor released and consequently these devices are outside the scope of this report.

## Chapter 2

# Background

This chapter describes the original Continuity, a web game to which Continuity 2 is a sequel. This chapter also provides a technical overview of the iPhone platform, for which Continuity 2 was designed. Last, a summary of common schemes commonly used by iPhone games is presented.

## 2.1 Continuity: The Sliding-tile Puzzle Platformer



**Figure 2.1:** A screenshot of one of the levels in Continuity.

Continuity is a puzzle-platformer that was designed to be played using a keyboard and monitor on a personal computer. Continuity can be played using the Adobe Flash Player at a resolution of 640 by 480 pixels. This section introduces description of the game’s mechanics and controls.

### 2.1.1 Game Mechanics

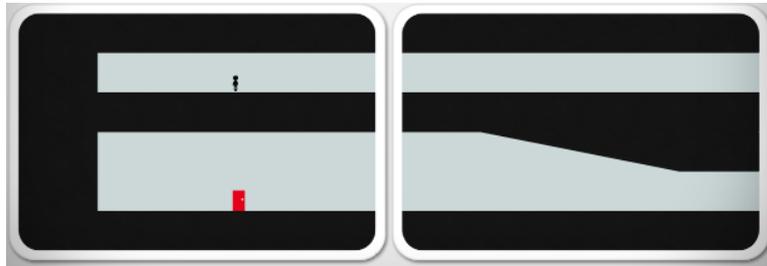
Continuity is a game comprised of 32 levels, arranged linearly. In each level, the player attempts to navigate a character through a 2-dimensional world in order to collect all the keys placed across the level before reaching a door. When the player reaches the door, having collected all the keys in that level, the player completes the level and advances to the next.

Gameplay in Continuity is divided into two distinct control modes. The player must use both modes to complete the levels. In one mode, hereby referred to as the macro-mode, the player rearranges tiles in the same way as one interacts with a 15-puzzle. A 15-puzzle, or  $n$ -puzzle for any number of tiles in general, is a puzzle with tiles placed out in a grid leaving one empty space. The goal of this type of puzzles is to rearrange the tiles so that each tile lies on it’s predefined location in the grid. In Continuity, however, tiles have no predefined location.

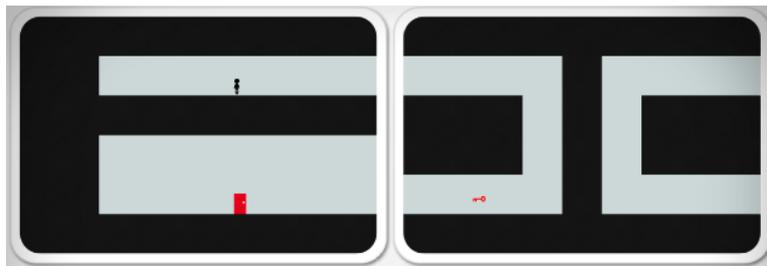
In the second mode, hereby referred to as the micro-mode, the player controls a character as in a platformer. The actions the player can take while in micro-mode are limited to moving the player left or right and jumping.

The player cannot control the height of a jump; the character jumps the same height every time. The game was designed in order to minimize any dexterity and timing requirements. Levels were designed such that the platforms are relatively wide and gaps that the character needs to jump are relatively narrow. The act of moving the character is not meant to be a challenge itself.

When in micro-mode, the character can move across a border between tiles when the images on the given border align graphically. The character cannot move between tiles unless the border forms a match in this manner. See figures 2.2 and 2.3. The character exists only within the tiles and cannot move between or outside the tiles. From the character's perspective, when a matching border occurs, the tile's contents forms a continuous world, with no gaps.



**Figure 2.2:** These tiles match as their borders form a continuous image along the entire border. The character may move across their border, from one room to the other.



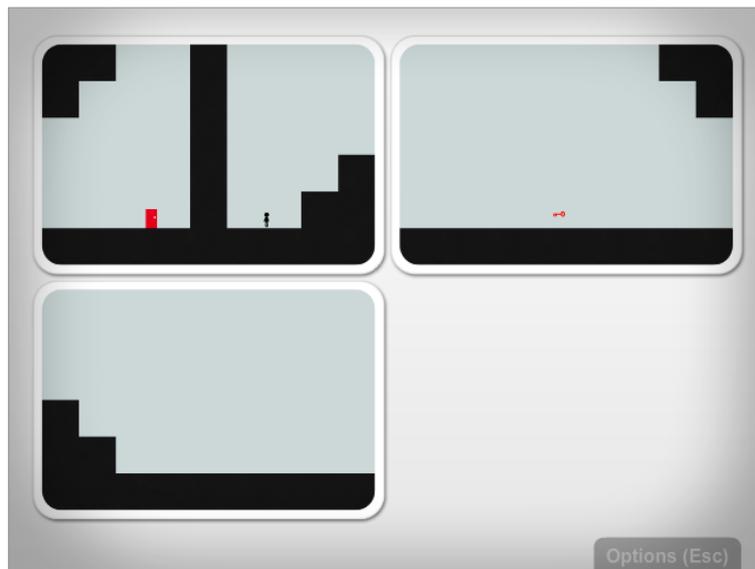
**Figure 2.3:** The side between the two tiles does not form a continuous image along the entire border as the bottom tunnel does not align. The character cannot move across these rooms' borders.

The game is divided into 32 levels, presented linearly. Generally, the difficulty of the levels increases as the game progresses. The goal of each level is for the character to collect all the keys and then reach the level's door. The character collects the keys by colliding with them. Once the character reaches the door with all the keys picked up, the player progresses to the next level.

If the character falls out of the bottom of a tile and there is no matching tile beneath the tile the character currently inhabits, the character is said to have died. See figure 2.5. The one and only consequence of dying is that the character is reset to the position of the last collected key or at the start point, if no keys have been collected in the current level. The tiles remain in the same positions as before death. There is no limit to the number or times the character may die and there is no other consequence.

### 2.1.2 Teaching the Rules of the Game

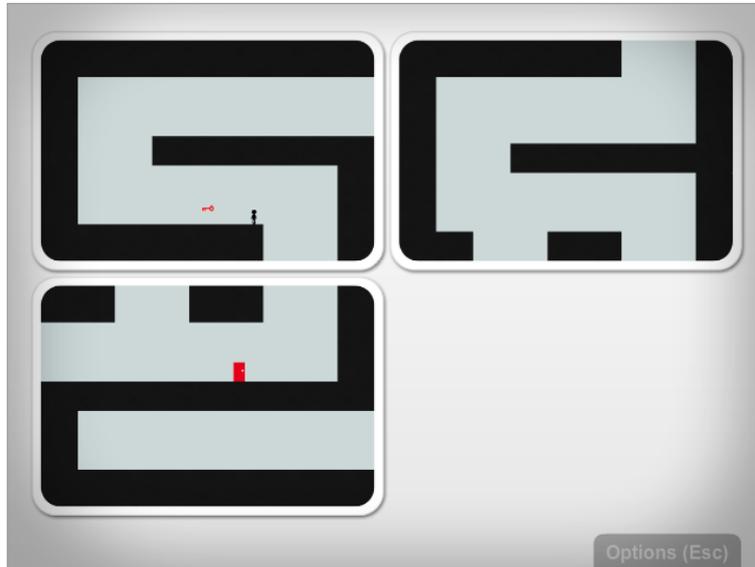
There is no explicit tutorial in Continuity and no help text is available. The initial levels of the game are designed to facilitate the players' learning of the game's rules. Each of these levels focuses on conveying one particular aspect of the game. See figures 2.4, 2.5, 2.6 and 2.7 for examples of such levels.



**Figure 2.4:** Level 4 serves to educate the player about Continuity's tile-matching rules.

### 2.1.3 Level Progression

When a new game in Continuity is initiated, the player is presented with the first level. To continue to the next level the player can either complete the level or choose "Skip Level" from the options menu. From the options menu the player can also choose to restart the current level, although the player can never reach a state that requires restarting.



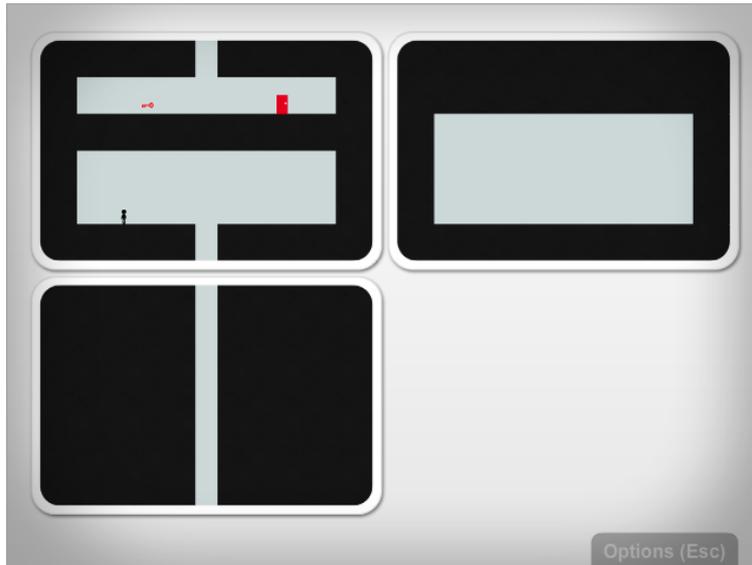
**Figure 2.5:** Level 9 serves to reinforce that the player cannot move across the bottom of a tile if the tile beneath does not match. The initial configuration of this level encourages the player to move the character immediately to the room with the door. However, if the player does this, the character will die.

The game keeps track of which levels have been completed and which have been skipped. Once all the levels have been completed the player is said to have completed the game. If the player completes or skips the last level while still having skipped levels left incomplete, he or she is transferred to the first of the skipped levels. If the player completes the last of those skipped levels, the player is transferred to the first of the remaining skipped levels and so on. When all the levels have been completed the player has completed the game. The mechanism of skipped and completed levels is not transparent to the player. The game does not present to the player which, if any, levels have been skipped nor that he or she will be presented another opportunity to play them.

The only way a player can move to a specific level in *Continuity* is by choosing “Restart game” from the main menu and skip through all the levels until he or she reaches the desired one.

#### 2.1.4 Audio

*Continuity* features two musical tracks, one for each of the modes and a number of sound effects. The macro-mode music track is a soothing piano-based song, while the micro-mode one has a more electronic sound and a higher tempo. The tracks are synchronized so that their melodies are continuous when switching between modes though the musical style changes.



**Figure 2.6:** Level 13 requires that the player understands that the game world is paused while the player is using the macro-mode.

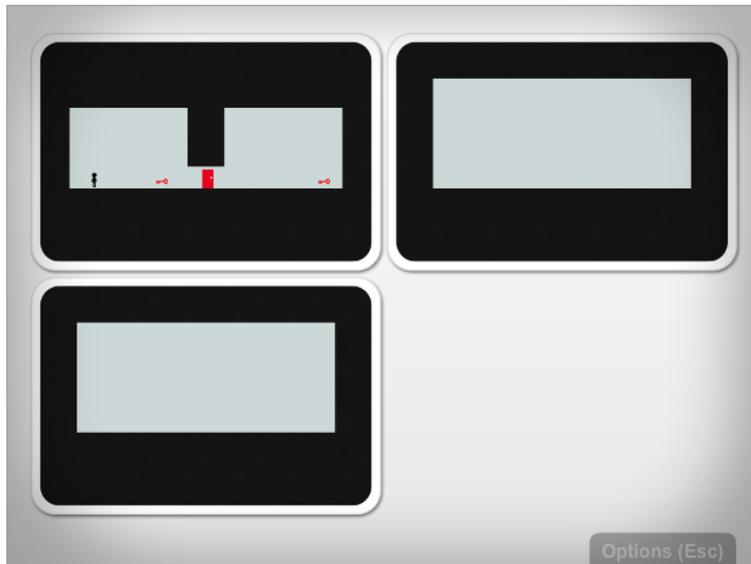
The micro-mode song is actually also part of the macro-mode song, however it is muffled to give the impression that the beat goes on within the game world and is heard from outside.

### 2.1.5 Controls

Continuity uses a keyboard as the primary input device. The player can navigate the menus via the mouse; however, the gameplay can be controlled exclusively by the space bar and the arrow keys. The same keys are used in macro and micro-mode but map to different actions depending on the mode. There are some redundancies built in, as the W, A, S and D keys can be used instead of the arrow keys.

Key	Macro	Micro
Spacebar	Zoom in	Zoom out
Left	Move tile left	Move character to the left
Right	Move tile right	Move character to the right
Up	Move tile up	Jump
Down	Move tile down	No function

**Table 2.1:** Continuity's keyboard controls.



**Figure 2.7:** Level 17 was created to demonstrate to the player that all the keys in a level must be collected to complete the level. To assist this point, the character must move through the door after having collected one key in order to attain the other.

### 2.1.6 Release

Continuity was made publicly available online on the 15th of November 2009. At the same time, the game was submitted to the 2009 Independent Games Festival (IGF). IGF is an international competition held annually since 1998 that aims at promoting the best independent game developers [1]. Continuity was announced as the winner of the Best Student Game award at the IGF on the 11th of March 2010 and, at the time of writing, the game has had over 5 million visits in total.

## 2.2 The iPhone Platform

As of May 2010, Apple, Inc. offers three groups of mobile devices that runs its operating system, the iOS: the smartphone, iPhone; the portable media player, iPod Touch; and the tablet computer, iPad. All three of these mobile devices feature a multi-touch display and are able to run applications developed by third-party developers. The iPhone and iPod Touch share the same form factor with a screen size of 3.5 inches while the iPad is bigger, using a 9.7 inch screen [5, 6, 3]. All share a common hardware and software platform, hereby referred to as the iPhone platform.

The term “iPhone” will hereafter refer to all the devices sharing the iPhone form factor, i.e. all the versions of iPhone and iPod Touch.

### 2.2.1 The Smartphone iPhone

There are three models of the iPhone: the iPhone, the iPhone 3G and the iPhone 3GS. The first generation iPhone was released on June 29th, 2007, the second, iPhone 3G, on July 10, 2008; and the third generation, iPhone 3GS, on June 9, 2009 [10, 9, 7]. From the first release in 2007 through the second fiscal quarter of 2010 a total of 51.2 million units had been sold. All generations of the iPhone have a multi-touch display and lack physical keyboard. The screen size measures 3.5 inches and has a resolution of 480 by 320 pixels. All generations also have an accelerometer, a camera and a microphone. The most significant difference between the first and second generation iPhones is the addition of 3G network connectivity and GPS in the second [12]. Between the second and the third generation iPhones, the main difference is the addition of a digital compass and a 3.2 megapixel camera that can be used for recording video [4].

### 2.2.2 The Portable Media Player iPod Touch

Each of the three iPhone models have a corresponding iPod Touch model with approximately equivalent computing and graphics performance [16, 15]. The iPod Touch runs the same operating system, the iPhone OS and has the same multi-touch 3.5 inch display as the iPhone models. The main difference between the iPhone models and the iPod Touch models is that the iPod Touches are not mobile phones. Also, the iPod Touch has no GPS capabilities or camera. The iPod Touch models are slimmer and lighter than their corresponding iPhone version. The third generation iPhone, for example, weighs 4.8 oz and the third generation iPod Touch weighs 4.1 oz [5, 6].

### **2.2.3 The Tablet Computer iPad**

The iPad was announced on January 27th, 2010 and is similar to the iPhone/iPod Touch models since it uses the same type of multi-touch display and runs the iPhone OS. It also has many of the same features such as accelerometer and WiFi networking capabilities. The main difference between the iPad and the iPhone/iPod Touch models is that it has a 9.7 inch multi-touch display that supports a resolution of 1024 by 768 pixels. The iPad weighs 1.5 lb and there are two versions: one with 3G connectivity and one without [8].

## 2.3 Casual Gaming

The definition of casual gaming is expressed in many, sometimes contradictory, ways. One approach to defining this category of games is to contrast them to games that require a great deal of devotion to be rewarding, so called hardcore-games [27]. J Juul describes casual games as games that provide a meaningful gaming experience in short time, but doing so without preventing longer time devotion. Juul also describes five components of casual game design that helps analyzing to what degree or in what aspect a game is casual.

1. Emotionally positive fictions dominate over negative in casual games.
2. Casual games tend to exhibit high usability and are easy to understand.
3. Casual games exhibit interruptability, which is the facilitation of players playing in short bursts.
4. Casual games often become very difficult as the game progresses but punishment for failing is low.
5. Excessive positive feedback, or juiciness, is common in casual games.

In a survey conducted in 2008 by Juul, 81% of the respondents reported that interruptibility was “Very important” when playing downloadable casual games. The same study indicated that telephone calls were the primary source of gameplay interruption[27].

Based on a similar view on casual games as Juul’s, A. K. Kultima describes four design values for casual game design[29]. These design values are acceptability, accessibility, simplicity and flexibility. Acceptability refers to the ability to attract a wide heterogeneous audience. Ways to address acceptability include avoiding offensive topics and endorsing positive values. Accessibility denotes making it easy for people with varying backgrounds and abilities to grasp the game concept as well as physically reach the game. Simplicity regards removing or hiding complexity. Using single-button control schemes, automating actions or gradually introducing game elements if they are numerous can achieve this. The fourth design value described, flexibility, constitutes making a game that allow for use in multiple contexts, exhibits the aspect of interruptibility from Juul’s definition and can be played in multiple modes [29].

## 2.4 Mobile Gaming

A mobile game, for the purpose of this report, is simply a video game that can be played on a mobile device such as a cellphone, smartphone or portable media player. This definition does not include video games for handheld gaming devices such as Nintendo Gameboy, Nintendo DS or Sony PSP. Mobile gaming is a recent topic which has seen significant growth in recent years [2]. In 1997, Nokia included a re-make of the classic arcade game Snake on its mobile device Nokia 6110. This became something of a starting point for mobile gaming [2]. Today Snake and its successors have been made available on more than 350 million devices. It has been suggested to be the most-played mobile game ever [17, 28].

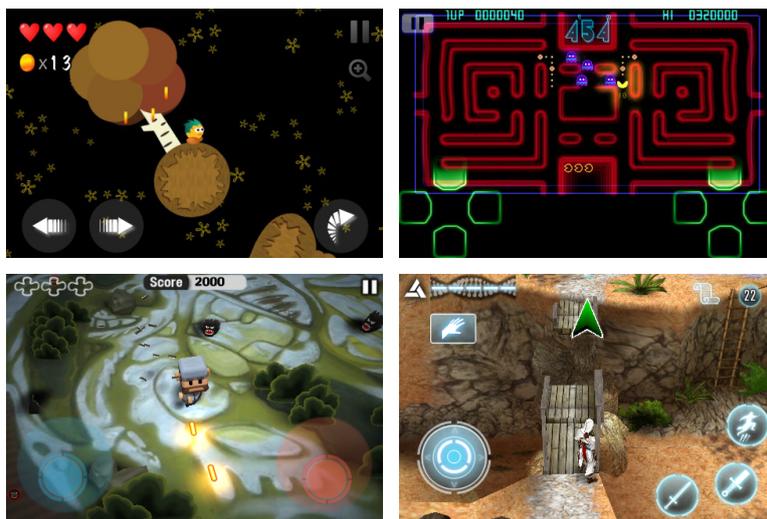
The capabilities of the mobile devices have seen a radical change since the release of the Nokia 6110 adding significantly larger color displays, accelerometers, touch-input, cameras, digital compasses and a new level of connectivity with technologies such as WiFi and BlueTooth. Reports from 2006 indicate that there are over 2 billion cellphone users worldwide [2]. In addition, new distribution channels such as the Ovi Store, the Android Market and the App Store have made downloadable games easily accessible from many places. Currently, mobile gaming is a multi-billion dollar market and it has been speculated that in 2010 mobile gaming will overtake both console and PC games in market value [2, 37].

In a study with over 1,800 participants performed by Nokia in 2006, researchers found that 80% of the mobile gamers played mobile games more than once a week and 34% played every day. The average session length was 28 minutes [11]. Another study conducted in 2004 with 752 participants reveals an average play session time of 17 minutes. Among these 752 users, 60% played games once a day or more. More than 60% reported that they typically play games at home [14].

## 2.5 Character Movement in iPhone Games

There seems to be a lack of consensus about how to control characters through 2D environments in iPhone games. In games for videogame consoles, character movement is usually primarily accomplished via a directional pad or thumbstick. Since the iPhone lacks any physical artifact to manipulate, other than itself via the accelerometer, game creators have little convention to draw from when creating games.

Many games use virtual buttons to control characters through 2D environments. The player presses the display where these buttons are drawn to control the character. One common realization of this paradigm is to make the iPhone display mimic a gamepad. The player holds the iPhone horizontally and uses his or her thumbs to press the virtual buttons or manipulate virtual thumbsticks. Unlike with an actual button or thumbstick, the virtual representations provide no tactile feedback.



**Figure 2.8:** Many iPhone games mimic a gamepad by displaying on-screen buttons. Clock-wise from top-left: *Soosiz*, *Pac-Man Championship Edition*, *Minigore* and *Assassin's Creed: Altair's Chronicles*.

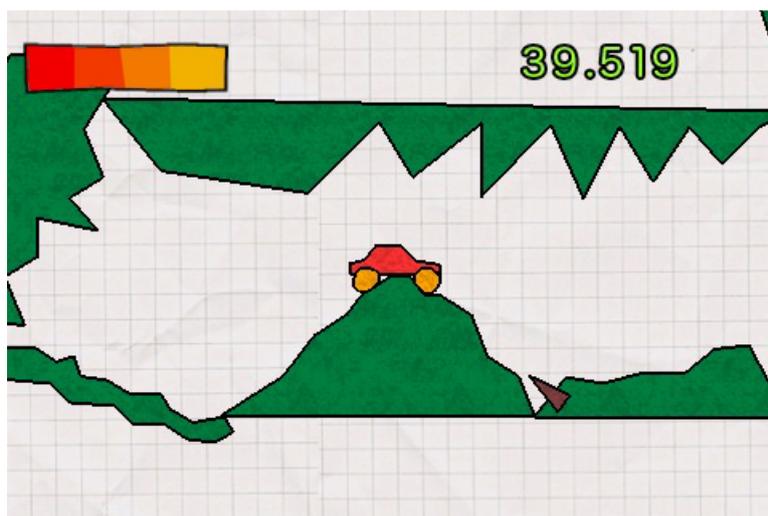
Other games have attempted to move away from the gamepad conventions. Many games use gestures to move the game character. For example, in the game *Kami Crazy*, the player swipes a finger left or right, which causes the character to move to its left or right, respectively. The character will continue moving in the indicated direction until another swipe occurs. The character cannot be stopped. The player can flick a finger upwards on the screen to make the character jump.

In games like *Spider: The Secret of Bryce Manor* and *Jelly Car* the player controls a character directly without the use of on-screen UI elements. In both games, the player touches the screen and the character



**Figure 2.9:** A screenshot of the iPhone game *Kami Crazi*. The player interacts with the game via touch-based gestures rather than virtual buttons.

moves towards the location of the touch. As the avatars move, the games' virtual cameras follow so that the characters stay relatively centered in the iPhone's display. In *Spider: The Secret of Bryce Manor*, the player swipes across the spider, the player's avatar, and the spider jumps along direction of the swipe. Players can play these games holding the iPhone as a gamepad and using their thumbs, or by supporting the iPhone with one hand and using the other hand's index finger for input.



**Figure 2.10:** Games featuring direct-control of an avatar. In both the player touches the screen in the direction he or she desires the avatar to move. **Above:** *Spider: The Secret of Bryce Manor*. **Below:** *Jelly Car*.

## Chapter 3

# Methodology

In the 1960s there was a movement among design theorists towards systematizing the act of designing. Many designers worked to develop a framework for the design process that would lead to a more efficient workflow. Some even had the ambition to fully automate the design process by enabling computers to act as designers without human intervention [18]. While these ambitions fell short, the focus resulted in a number of design methods that can be used to structure the design process. A design method is, by J.C Jones' definition, any action that is performed when designing [26].

From the viewpoint presented in Jones' *Design Methods*, each design method is useful in one or more phases of the design process. He divides designing into three stages: divergence, transformation and convergence.

In brief, the divergence phase constitutes broadening the view by exploring the design situation. Transformation follows the divergence phase and is mainly concerned with structuring the design situation, fixing objectives, determining critical aspects, identifying constraints and so on. Convergence is the phase where the work initialized in the divergence and transformation phases are completed and a final design solution is formed.

The design process in whole does not transition from divergence to transformation and then to convergence without revisiting the divergence or transformation stages. The idea of these stages is rather that of dependence. Work in the transformation phase cannot commence before sufficient work in divergence phase has been carried out and the same applies to advancement from transformation to convergence [26].

This section presents methodologies relevant to this thesis from the fields of design theory, interaction design and game design.

## 3.1 Generating Ideas

### 3.1.1 Brainstorming

The term “brainstorming” was originally described in *Your Creative Power* in 1948 as a design method where one uses the brain to attack a problem [33]. Brainstorming is perhaps the most commonly used design method and one that many people are familiar with. However, Kelly suggests in the paper *The Perfect Brainstorm* that brainstorming is a skill that needs to be learnt and reflected upon to be effective. The general idea of brainstorming is that focusing on producing many ideas, regardless of their quality, may result in a few very good ideas. Jones describes brainstorming as a group activity whereas Fullerton finds the method suitable to perform alone. Brainstorming is a general method that comes in many forms and can be applied to all stages of the design process.

Brainstorming is described as a way to address a design problem and is appropriate to use when the problem can be stated in a straightforward way [26]. Fullerton also points out the importance of having an aim with a brainstorming session where more specific goals are better [23]. The key to a successful brainstorming is to keep focus on quantity and not to abandon or criticize ideas even though they may seem infeasible at first glance [26].

### 3.1.2 Removing Mental Blocks

The objective of this design method is to find new directions when the design work has come to a halt or the flow of feasible ideas is weak [26]. Jones describes three categories of methods with slightly differing approaches to removing mental blocks.

The three categorizes are:

1. Apply transformation rules to an existing unsatisfactory solution or to parts of it.
2. Search for new relationships between parts of an existing unsatisfactory solution.
3. Re-assess the design situation.

Applying transformation rules denotes changing some property of the unsatisfactory solution. For example, properties of the design can be magnified or minified, consolidated, re-arranged and substituted.

Searching for new relationships consists of dissecting the unsatisfactory solution into parts and then investigating the relationships between these pieces. Parts may be presented in a list or arranged in a matrix to facilitate this task.

Categories 1 and 2 focus on improving unsatisfactory solutions. Another approach consists of focusing on the problem. Methods for category 3 are numerous and aim at identifying limitations in the formulation of the problem that may unnecessarily restrict the spectrum of possible solutions. An example of such a method is *The Why? Why? Why? Method* outlined in *Design methods*. Using this method contradictions are searched for by repeatedly asking, “Why,” in regards some part of the problem.

### 3.1.3 Stating Objectives

Stating objectives is a method that resides in the convergence phase by Jones’ definition. Its aim is to help obtain a clearer view of the problem itself. The steps of this method are the following:

1. Identify the situation within which the design is to operate.
2. Identify features of this situation with which the design needs to be compatible.
3. Ensure that statements identifying the objectives are compatible with each other and with information that becomes available when designing.

For designing a new transport system, for example, step 1 may involve finding out where it will be used and the unique circumstances of that location. Step 2 involves stating the essential objectives, identifying resources available and defining the client’s expectations. The last step involves reviewing the network of objectives produced in step 1 and 2 and making sure that they are internally consistent [26].

### 3.1.4 Risk Assessment

This method forces the designer to think pessimistically as the aim of it is to find out all the things that could prevent the product from being realized or fail to live up to expectations [35]. This method involves two steps.

First, the designer lists all the conditions under which the concept cannot be realized. For example, in the context of a videogame, the target platform must be able to render a certain number of objects at a certain frame rate.

Second, the designer investigates how those situations can be avoided. In the example of rendering objects for a videogame, the designer would evaluate optimizations or compromises that could be made in order to achieve adequate rendering performance.

## 3.2 Market Research

Market research involves collecting and analyzing information about human behavior to help in decision-making [24]. Online surveys are one particular method for market research that can be used to find out more about people's desires, needs and beliefs. In 2006, a third of all market research was performed online [24]. Besides being more cost effective than conducting face-to-face interviews or telephone surveys and being faster to perform, it has also been suggested that online methods provide more honest data [24]. In 2006, a third of all market research was performed online [24].

Among the concerns with online methods is how well the respondents represent the targeted population. For long surveys that may possess an additional problem in that the quality will decline as the survey loses the respondents' attentions. [24].

### 3.2.1 Questionnaire

One way to query users, regardless of whether this is performed online or offline, is to create a questionnaire for users to complete [22]. Due to the fact that the questions are fixed in advance, questionnaires may not provide an as rich picture of the respondents as other query techniques, such as interviews. Questionnaires can however be administered with greater ease and reach a wider group of respondents.

Because the evaluators generally are not present when users complete the questionnaire, it is crucial that it is properly designed. Using other user query techniques evaluators may have the opportunity to clarify the questions in case of confusion [22]. Performing a pilot survey to evaluate the questionnaire is something to consider due to the importance of deciding on questions, type of answer alternatives and how the questions are formulated [26].

There are numerous question and response formats that can be used in a questionnaire: open-ended, multiple-choice, ranked, etc. In general, using closed questions is preferable, as they require less effort from respondents. For measuring opinions, attitudes and beliefs, something called Likert scales are frequently used [36]. A Likert scale question consists of a statement and a number of answer alternatives. By choosing an alternative, the respondent specifies to what extent he or she agrees with the statement. The scale is commonly 1–5 or 1–7; however, 1–3 and 1–9 are also used. Deciding on a scale is important for getting valuable responses, however, the opinions about the benefits and drawbacks of using large versus small scales differ. Clearly, having a 1–3 scale the respondents have no way of specifying to what degree they agree, or disagree, with the statement as the alternatives become only disagrees, neutral and agrees. It is argued that with too large of a scale people have a problem with picturing the difference between two

points [36].

### 3.3 Prototyping

In the context of game design, Fullerton describes a prototype as a playable model of the game idea focused on core game features [23]. In a broader sense, a prototype can be described as a representation of a design solution created before the final product is available for evaluation [19].

Prototyping is of great importance for creating interactive computer systems and has become something of a common practice within the field of human-computer interaction [19]. The importance is further emphasized by Fullerton, who states that prototyping is a key activity in the process of game creation [23].

The level of abstraction between prototype and the product envisioned can vary greatly. A cardboard box or a piece of wood can at times serve as valuable prototypes of a complex software system [36]. According to the definition presented by Preece et al prototypes that do not closely resemble the final product are low-fidelity prototypes. High-fidelity prototypes, on the other hand, are prototypes that may look and feel the same way as the final product or can be interacted with in a similar way. For games, Fullerton, keeps a similar categorization but divides game prototypes into physical and digital prototypes [23].

The advantages and disadvantages of low-fidelity and high-fidelity prototyping are subject to debate [34]. Preece et al nevertheless promote the use of low-fidelity prototypes, arguing that they are good for investigating conceptual flaws and cost effective. High-fidelity prototypes need to be more carefully designed to give valuable feedback and are often expensive to produce. Preece et al also point out that when using low-fidelity prototypes it is more or less obvious to users that compromises have been made. With high-fidelity prototypes, the user might confuse the prototype with the actual system, which distracts from the focus of the test. Helander stresses this even further and suggests that clarifying the purpose and limitations of a prototype to users is crucial to the outcome [30].

The reasons for creating a prototype also vary. Preece et al express that the general purpose of prototyping for interactive systems is to investigate a design solutions feasibility and appropriateness [36]. Prototypes can however also be used as a way of exploring and generating ideas [19]. In addition, prototypes can be used to communicate a design solution more effectively than via words or static images [20].

More concretely, Fullerton, describes four categories of games prototypes, each targeting one aspect of a given game. She stresses the importance of knowing what question needs to be answered or clarified so that the prototype can be designed to properly address that. The four categories are game mechanics, aesthetics, kinesthetics and technology.

In *The Art of Game Design*, Shell outlines how to practically prototype games. Schell provides eight brief tips on how to do productive prototyping

for games [35]. Below is a summary of the advice presented.

1. Ask a question: Make sure you know what you want to find out.
2. Forget quality: Create a rough version, partly because quality takes time and partly because quality may defeat the purpose.
3. Don't get attached: To rapidly create a prototype you need to cut corners. Plan to throw away.
4. Prioritize the prototypes: Prioritize prototypes if there is a need to prototype multiple features or aspects of the game.
5. Parallelize prototypes productively: Prototypes that target separate aspects of a game, such as art and technology, may be worked on simultaneously.
6. It doesn't have to be digital: Low-fidelity are beneficial even for experience-focused artifacts such as games.
7. Pick a Fast-loop game engine: Adopt a software development cycle that allows for quick and easy modifications.
8. Build the toy first: Leave game rules out at first and create something that is fun to play with without any specific goal.

## 3.4 Evaluation

Evaluation lies at the heart of interaction design as one of the basic activities performed during the entire course of development [36]. Playtesting, within the field of game design, refers to the activity of exposing a user to a game and collecting information about the user experience. In this field, play testing is the predominant form of evaluation and has been argued to be critical to creating a good game [35]. It has also been suggested that playtesting is required for a game to reach full potential [23].

Designers and developers extensively use a prototype or system to a great extent when developing and so become experts at their own system. The designers will learn how to use the system and possibly find it intuitive because they are accustomed to using it [32]. A concrete example of this problem can be observed in the paper *Using “tilt” as an interface to control “no-button” 3-D mobile games* in which playtesters preferred a solution that was deemed counter-intuitive by the researchers. The power of empathy is a great tool, but no substitute for observational techniques [32].

There are a number of other approaches to evaluating interactive systems that do not involve users or depend on the evaluator’s empathy. Expert analysis methods are examples of such, however, the primary goal of these are not to assess the actual use of the system but rather to ensure that the system adheres to certain usability principles [22].

On a practical level, evaluation specific to mobile applications poses a problem as mobile applications are typically used in small bursts throughout a day on different locations [25].

The participants of such an evaluation clearly have an impact on the outcome of a user test. Finding participants that are representative of the target group is essential but the importance varies with the intention of the testing. If the intention is strictly to uncover usability issues, the importance of finding a representative user group is lower [30]. Choosing participants is considerably easier if you know your target group. Designing for everyone is a difficult task, as “There is no such thing as an average person”, as Norman states in *The Design of Everyday Things*.

The sample size is also of significance but it has been shown that even with a single participant one will be able to find a third of the usability problems [31]. The added return diminishes with the number of users and Nielsen and Launder claims that there is little to gain from testing with more than five. Dix et al adds that five may be sufficient to find problems but may not provide information about the extent of the problems [22].

### 3.4.1 User Observation

There are a number of approaches to observing users with the most significant difference being the level of interaction between the evaluator and the

user. Outlined below are a few methods that ask users to reflect and analyze their behavior and a number of others where users are observed in a manner that more closely resemble the setting of actual players.

Establishing the goals of the activity is key to a successful user study. Dix et al suggests that a hypothesis should be formulated before testing commences and so the aim is to prove or falsify this hypothesis [22]. Another way of defining the aim of the observation is to establish a number of questions to be answered. A proper set of questions is of great importance as general questions like, “Is the game fun,” are difficult to answer from limited testing [35]. Schell proposes that the value of playtesting rises the more specific the questions are.

### **3.4.2 “Quick and dirty” Evaluation**

This is an informal technique where there is an ad hoc setup and evaluators briefly watch and talk with the observed users. It may be performed as field study, where the evaluator visits a school, office or other place where users can be met and observed in an unconstrained manner for a limited time [36].

### **3.4.3 Think-Aloud Technique**

When strictly observing users there is little insight to what the users are trying to achieve through their actions or what their impressions of things are. Using the think aloud technique users are instructed to express verbally what they are thinking during the whole session [22]. This way the evaluators get a better grip of what users attempt to do. Dix et al acknowledge, however, that the technique does not give a full insight because users may state what they are trying to do, but not why. For example, users may mention that they want to undo an action but not what the reason for the desire of undoing an action comes from or what alternatives have been considered before the decision to undo an action was made.

Another potential drawback of this method is that the act of describing what one’s doing may change the way one does it. When users experience unexpected events silences often occur, probing the evaluator to intervene. Intervening may be required to follow protocol but potentially alters the behavior of users even more. A solution to facilitating the practice of the think aloud technique is to let two people collaborate and so they are encouraged to communicate with speech [22].

### **3.4.4 Cooperative Evaluation**

Cooperative evaluation, as described by Dix et al; encourages the user to criticize the system and allows the evaluator to interrupt with questions like “why” or “what if”. This method is less constrained but induces a different

mindset where the goal of the user is not only to accomplish a task but also to actively critique the experience of accomplishing a task [22].

### **3.4.5 Fly-on-the-Wall (FOTW)**

Using this technique there is no interaction between the user and the evaluators. The user is observed using the artifact in a regular setting and time frame. An important benefit of using this method, in contrast to for example interviews, is that information gathered is largely about what they did, not what they say that they did [13].

### **3.4.6 Interviews**

Interviews provide a direct and structured way of gathering information about the user experience where focus lies on the questions “why” and “what if”. This is effective for a higher-level evaluation and may provide a more holistic view than some other methods such as *Fly-on-the-Wall*—. It may shed light on the preferences of the users, the impressions with using the artifact and attitudes in general. This may reveal issues that had only been discovered by chance if stumbled upon during session. A major negative aspect of this technique is that users will provide a rationalized view of what did occur rather than an accurate one.

## Chapter 4

# Planning Development

We anticipated that the development of Continuity 2 would go quickly and smoothly based on our experience developing the original Continuity. The original Continuity was developed over a period of two months by four people, each working half time. When we began to plan the development of Continuity 2, our belief was that over a period of three months, two developers working full time would be able to complete a similar task. We anticipated that we would be able to reuse much of the design of the original. Further, since we worked out many technical problems while implementing the original, we believed we would be able to avoid many such problems in the sequel. We believed that our experience creating Continuity would allow us to be able to produce Continuity 2 within a matter of several months.

## 4.1 Time Plan

Our initial plan was to dedicate six weeks to do everything required to implement a high-quality prototype. During this period, we would modify Continuity’s design through conceptualization and low-fidelity prototyping. We intended to then use playtests to evaluate our design decisions so that by the end of these six weeks we would have produced a playable prototype that contained all of the gameplay mechanics that would be present in the final game.

We created this schedule based on three assumptions we drew from our experience implementing the original Continuity. First, we assumed that we had already overcome most of the implementation problems that we were likely to encounter while developing Continuity and that we would be able to quickly re-implement the same or similar, solutions for Continuity 2. Second, we believed that we would be able to directly reuse some work from the previous game such as the level editor and some of the art assets. Thirdly, we believed that we would quickly be able to create interesting, fun modifications to the initial game’s design so that we could spend much of the time polishing the game and designing the levels.

After we had developed this prototype, we intended to thoroughly test it and then spend another three weeks making modifications based on player feedback. This would be the last opportunity to evaluate player feedback to gameplay features. During this time, mechanics might be modified or removed, but no new features or mechanics would be added.

According to the plan we created, after 10 weeks of work, all game and software features would be complete. We would then spend the remaining three and a half weeks fixing bugs and otherwise fine-tuning the game. For example, we would spend this time creating and modifying levels, tweaking animations, modifying the character’s movement speed and testing different menu layouts.

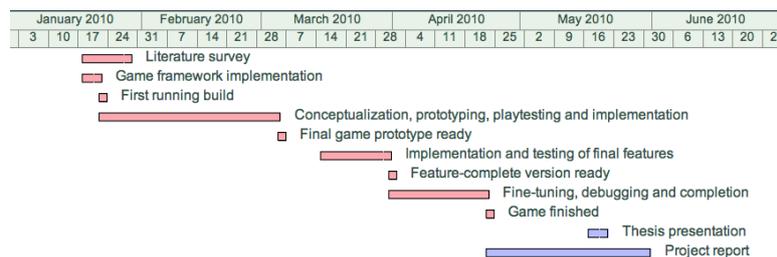


Figure 4.1: A Gantt chart representing the initial time plan.

## 4.2 Iterative Development

Rather than using a waterfall process, in which we would perform all design work before performing any implementation, we opted to take an iterative approach. This had worked well for us when designing *Continuity*. We would not specify the entirety (or even majority) of the design before beginning implementing. Further, we would not fix a hard schedule that would define certain periods for certain methods. Instead, we would include methods in the process as we believed them to be useful to the immediate situation.

We intended to conduct regular playtests throughout the game's development as a way to evaluate the game. We had used such playtests successfully during the development of *Continuity*, mostly as a way to evaluate aesthetics, level design and level progression. For *Continuity 2*, we intended to use playtests for these purposes as well as a fundamental way to evaluate game mechanics. For each playtest we would modify the playable product to highlight some specific aspects that we were attempting to evaluate rather than simply trying to get overall feedback of the game as a whole. Playtesters would not be informed of the purpose of a given test before or while playing so as to not draw too much of their attention to the aspect under tested.

We decided to primarily use *Fly-on-the-Wall* playtesting as a way to evaluate the players' experiences. We preferred this method of playtesting as we did not want playtesters to become overly conscious of their playing experiences while they played as might happen when using a method such as *Think Aloud*. For these playtests, we would provide the players with the minimum amount of information required to test what we wanted them to test and then allow the players to try to work through the game by themselves. In general, we would only interject when the player became confused due to errors of implementation (e.g. faulty animations) or was simply ready to give up.

During these playtests, we would not ask questions of players while they played because we believed that when players are asked to answer questions while playing, the act of forming a reasonable response about their actions causes them to analyze the game in a different manner than if they were left to themselves. Since a goal of ours was to create a game that teaches itself to players at an appropriate pace, we wanted the player to consider the game at the level that they normally would.

Our intention was to create levels throughout the entire duration of the development. At every state of development we intended to evaluate not only the gameplay mechanics, but also the levels we had created. We would then create new levels or modify existing levels to fill perceived problems with the way the game's education of the player or the game's pacing.

# Chapter 5

## Method

This chapter describes the design, prototyping and evaluation work carried out during the course of this project. The work is presented in chronological order and occurs in all of the three phases of the design process described by Jones [26]. Naturally, some work in the divergence and transformation phases have already been performed as this project builds on an established foundation.

The first four sections (*Defining Continuity*, *Assessing Risks and Building a Foundation*, *Establishing Goals* and *Survey*) focus on the divergence phase of development, aiming at investigating the design problem. The goal of these first steps was to establish a clearer view of the challenge at hand. It is also during this part of the project that possible design solutions are developed.

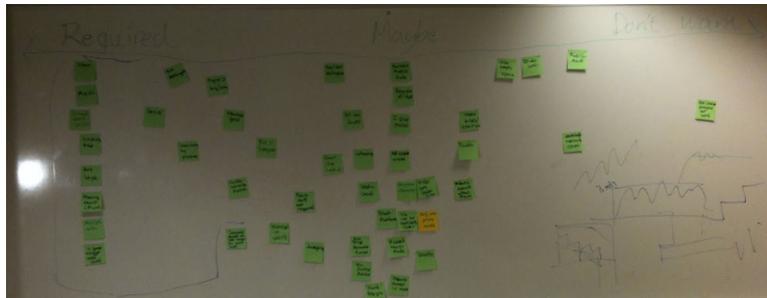
The work following those four chapters focuses on iterative development, prototyping and evaluation of the better of the design solutions brought fourth during the initial phase. These work done in these sections corresponds to the transformation and convergence phases of the design of Continuity 2.

## 5.1 Defining Continuity

We did not want to assume that Continuity 2 would simply be a superset of the original. In other words, we did not want to automatically conclude that a sequel would contain all the gameplay mechanics and features in addition to some new ones.

We feared that adding or changing too many elements would make the sequel unrecognizable to players of the original. On the other hand, the game might feel completely repetitive if it was not changed enough. One of the first tasks we engaged in was defining what elements are necessary for a game to feel like a sequel to Continuity. The goal of this task was to define the gameplay elements and aesthetics that must be present in order for a game to feel like a Continuity game.

In order to complete this task we tried to list all aspects of the original game. We wrote each element onto a sticky note. A spectrum was drawn on a whiteboard representing how critical elements were to the concept of a Continuity game. The spectrum was not intended to represent how important aspects were in relation to the original Continuity, but to a larger sense of Continuity as a series of games. We placed the sticky notes on the spectrum according to how important each feature was to this overall concept.



**Figure 5.1:** Game elements from the original Continuity were plotted along a horizontal axis. Elements to the left must be in a sequel. Elements to the right must be removed from a sequel. The vertical axis is not relevant.

After creating this range, we grouped these elements into categories as a way to more easily discuss and quantify the importance of each element. Five categories were created and we assigned each element from its continuous position on the plot into one of the five discrete categories.

1. Elements that must not be in Continuity 2.
2. Elements that should not be in Continuity 2.
3. Elements that might be in Continuity 2.

4. Elements that should be in the Continuity 2.
5. Elements that must be in Continuity 2.

The most important elements that we identified are presented here.



**Figure 5.2:** Elements from the original game were categorized based on their importance to the sequel. Category 5 is composed of elements that must be in the final game. Category 1 contains elements that must not occur in the game.

### 5.1.1 Segmented Platformer

We asserted that the manipulation of the segmented world was integral to the game. We believed the game should retain some sense that the player directly manipulates a segmented game world as well as some character in order to achieve some goal. From the player-controlled character’s perspective, the level should only consist of what is drawn onto these segments, with no level existing outside or between the segments. We further decided that, given the name “Continuity”, the segments of the world should seem to fit with one another and not simply be arbitrary shapes.

We believed that given that the  $n$ -puzzle seems like an easily recognizable and manipulated physical puzzle, that the segmented pieces should form some sort of  $n$ -puzzle. While we could have left the possibility open for other types of puzzles, such as jigsaw puzzles, we felt like the simplicity of sliding tiles was consistent with the minimal design we desired. As such, we stated that the world manipulation aspect of the game should take the form of an  $n$ -puzzle. We did leave open the possibility that the puzzle would differ in some way from standard  $n$ -puzzles such as by having more than one empty location or tiles of varying sizes.

### 5.1.2 Minimalism and Clarity

Many of the elements that we found to be necessary in a sequel concerned retaining a sense of minimalism that was present in the first game. We believed one of the great successes of the game was the way in which players learned the rules of the game through playing as opposed to through explicit

instructions or tutorials. The original Continuity also possessed a minimalism of actions. This lack of complexity made the game more casual, and made it more appealing to a wide array of players. Players only had two meaningful inputs to the game: moving something and toggling what they were able to move. We believed that players should learn the game primarily through playing in a mostly language-agnostic manner. This would add to the casual nature of the game, as the game would have a built in tutorial and would not rely on previous gaming experience.

We also concluded that the game should only possess a few, interesting, reusable game objects and rules. We decided it was essential that players could easily understand how to interact with every game object, regardless if they could necessarily immediately understand all uses of the object. For example, in the original game, the act of switching to the macro-mode paused the movement of all game objects that existed within the tiles. However, the player was not required to understand the full effects of switching modes until the game's eleventh level. We felt it was important for Continuity 2 to surprise players by allowing them to discover new consequences to previously discovered actions.

This degree of clarity should also be present in the levels themselves. We believed it was important for the player to be able to easily identify all elements in a level. Upon beginning a level, the player should be able to easily determine the primary objective of the level, although he or she should not as easily be able to discern how to accomplish achieving the objective. We held that presenting the player with all of the information about a level from the beginning enhances the puzzling aspect of the game.

Since we set forth that the game would be, primarily, a puzzle game that deemphasized dexterity or mechanical skills, we believed that the game should encourage players to think things through and should not require experimentation. We determined that dynamic elements must behave predictably so that once a player understands the kind of element, he or she is no longer required to guess or experiment.

### 5.1.3 Rewards

We found that the original game's lack of rewards was a problem that should be alleviated in a sequel. The original game did not reward players frequently or thoroughly enough. Based on data we collected from players, some levels took, on average, of over 80 minutes for players to complete. Players also were more likely to quit the game when playing these long levels. When players did complete levels of the original game, they were simply presented a screen informing them of how many levels were left in the game without any extra consideration as to whether they had been especially skillful in completing the level.

The original Continuity contained many levels that had non-obvious so-



**Figure 5.3:** This screen is displayed when the player has completed all the levels in Continuity. It provides little reward to the player.

lutions, which could be solved in a matter of seconds once one discovered the solution. We found these levels to be some of the most interesting, rewarding levels in the entire game. We believed the game should possess examples of such levels that could be solved quickly once the player came to terms with some aspect of the game. We felt that players were emotionally rewarded when they quickly transitioned from bafflement to success with a moment of comprehension. In the original game, we found these moments encouraged players to continue playing.

We believed it was very important to increase the explicit and implicit rewards for Continuity 2. Adding a greater sense of reward would increase the sense of juiciness and hopefully make the game more appealing to casual players.

## 5.2 Assessing Risks and Building a Foundation

Upon beginning development of Continuity 2, we believed that it was important to first determine what were the largest risks to the development. We decided to use risk assessment and prototyping to ensure that we would be able to implement solutions to fundamental requirements of the game's design. Our concerns were grouped mainly into three categories: inexperience with the iPhone programming environment, a lack of familiarity with the technical capabilities of the devices and uncertainty about how the form of the device would affect the gameplay.

We decided that the best way to mitigate these risks was to prototype solutions. This would allow us to prioritize implementing the requisite aspects of the game, since if we were unable to achieve adequate solutions we might have been required to drastically rethink the game's design.

In order to minimize the amount of work that would need to be thrown out later and to allow us to more easily prototype other gameplay concepts, we decided to also create the initial foundation of the game at this time. The objective of this was to be able to create a rough prototype of the game that allowed for touch input be able to move tiles and then for on-screen buttons to be able to move some physics-based character within and between tiles. This prototype was also supposed to enforce the tile-matching rules that were found in the original Continuity.

### 5.2.1 Assessing Risks

We assessed several technical risks, however, all of these turned out to be solvable without compromising any aspect of the limited design that Continuity 2 currently had. We decided to use the Cocos2D software library as it would help us immensely in being able to present graphics on the iPhone's display and play both music and sound effects. Based on this decision, we were able to determine that the iPhone was capable of using graphics and sounds similar to those used in the original Continuity.

When analyzing how the game might look on iPhone, we realized that perhaps one of the greatest risks was the screen size of device. All iPhones possessed a 3.5-inch display with a resolution of 480x320. The original Continuity operated at a resolution of 640x480, double that of the iPhone's display. We quickly determined that we could not simply scale the game down by half to fit the iPhone's screen.

When the player is in the macro-mode, it is important for him or her to be able to identify the location of the character and all other objects in order to be able to plan a route through the level, across many tiles. As a test, we displayed scaled-down screenshots from Continuity on the iPhone's display. We concluded that the character was simply too small to be easily visible from the macro-perspective.

In the original *Continuity*, the character height was  $1/14$  that of each tile. This allowed for very complicated levels that contained many vertical platforms at many different heights. By making the character bigger, relative to the tile-size, the levels effectively became smaller and could not be as complicated. Our desire was to work to find the minimum character size that players could easily recognize from the macro perspective so that the character would not be too large for each tile.

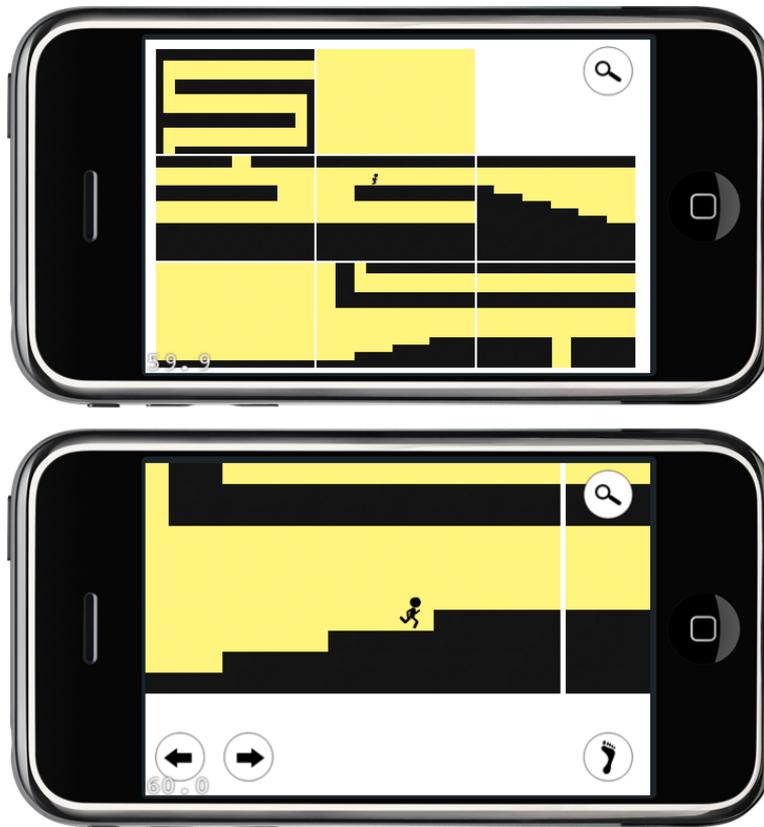
After creating a series of mockups, we found that a character height that was  $1/8$  the height of each tile was a reasonable compromise. This allowed the character to be clearly visible when playing in micro-mode and identifiable as possessing a human-like form when playing in macro-mode.

### 5.2.2 The Foundation

The foundation, which we would build upon throughout development, demonstrated solutions to all the major risks that we predicted. It also served as a competent basis for further development. It featured two modes of play as in *Continuity*. In the macro-mode, the player could move tiles by dragging them with his or her finger. In the micro-mode, the player could move the character around the screen using on-screen buttons. The player could switch between the control modes by pressing a magnifying glass button that was present in both modes.

The foundation was capable of loading and playing a series of levels from files exported from a level editor. Although, it was not capable of saving progress, a framework was in place to facilitate adding such functionality in the future.

This phase of development was intended to only take three weeks to create a very rough game prototype and use it to assess the identified risks. Unfortunately, this was overambitious scheduling and this task took closer to six weeks. The extra time was largely due to unfamiliarity with the programming environment and an overambitious appraisal of the ease of implementing the solutions.



**Figure 5.4:** Screenshots of the foundation prototype. **Above:** The macro-mode, in which the player can move tiles by dragging them. **Below:** The micro-mode, in which the player can use the buttons to make the character run and jump.

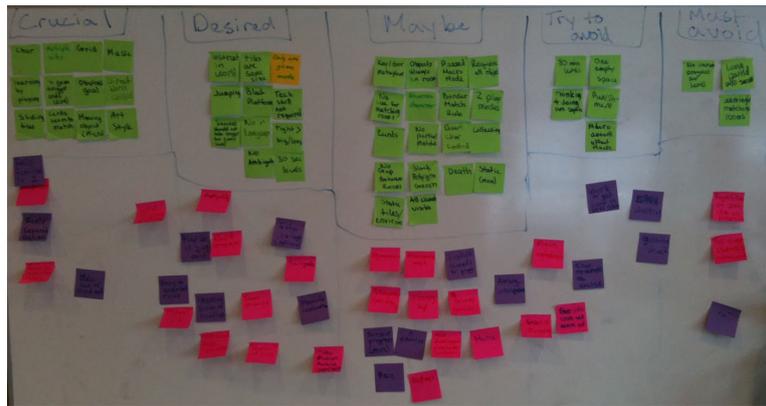
### 5.3 Establishing Goals

Since we did not want to assume that Continuity 2 would be a strict superset of the original Continuity, we began design work by appraising the game and setting goals for the development process. Using the *Stating Objectives* method [26], we created a set of core gameplay aesthetics that we wanted the sequel to possess. These concepts and values would guide the rest of the development process.

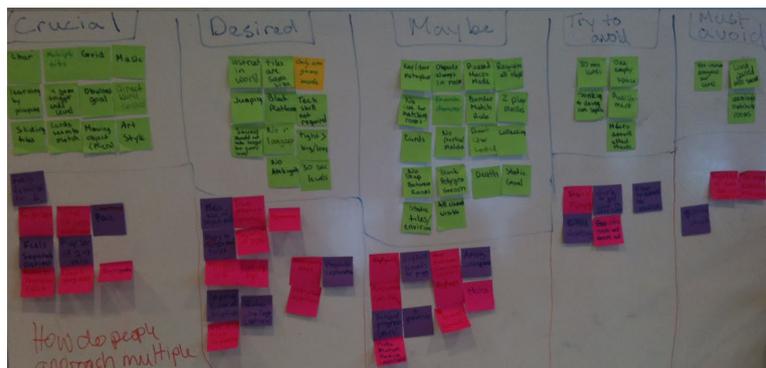
For this activity, we analyzed particularly successful and unsuccessful aspects of the original Continuity. Whereas while defining the game (Section 5.1) we tried to identify what was required for any game in the Continuity series, for this exercise we attempted to identify what aesthetic values we desired specifically for Continuity 2 on iPhone. Rather than simply establishing goals to either retain or eliminate gameplay elements, we dug further to identify the underlying aesthetic values that made the game elements so successful or unsuccessful. We approached this task mostly from our own

perspectives based upon our experiences playing the game, observing others playing the game and feedback we had received from others who had played.

We conducted this method in a manner similar to how we defined Continuity. First, we drew a spectrum onto a white board. Then, we wrote concepts onto sticky notes. We plotted each sticky note onto the board's spectrum based on our perception of how important it was for Continuity 2. Again, after creating a continuum, we grouped the elements into five categories for ease of later discussion.



**Figure 5.5:** Objectives for Continuity 2 were written on pink and purple sticky notes and plotted on a horizontal continuum according to how desirable they were for Continuity 2. The green sticky notes are from the *Defining Continuity* exercise (Section 5.1) and served as a reference.



**Figure 5.6:** After all objectives had been plotted, they were grouped into five categories corresponding to how much we desired their presence in Continuity 2.

While discussing the grouping of the objectives, we realized that there were several themes reoccurring across the objectives. After we finished grouping the objectives individually based on desirability, we clustered them based on related themes they shared. From discussing these clusters we

discovered a set of abstract aesthetic ideals that we wanted to appear in Continuity 2.

Some of the most important goals are discussed here.

### 5.3.1 Distinct, But Familiar

The first, probably most clear, goal we determined was that the game must be both distinct and familiar to players of the original Continuity. We had implicitly held this goal previously, but in this exercise we asserted it explicitly.

We understood that creating something that was both novel and recognizable would not be an easy task, but we set it as the primary objective. We wanted to create a game that both first-time and returning players would enjoy. Since the game was to be a sequel to Continuity, rather than some other, arbitrary game that possibly shared some mechanics, we felt that players needed to feel like the sequel was related to the original. However, since we were attempting to create a sequel rather than simply an iPhone port of the original game, we felt that the game needed to possess enough differences that players would not feel like they were retreading the original game.

Accessibility was also a concern in regards to balancing the new and the old. When faced with new gameplay mechanics, players who had and had not played the original Continuity would be on similar footing in terms of performance. However, for content that was similar to the original Continuity, the players of the previous game would likely perform much better than players that did not play the original. We understood that balancing the difficulty of the game for players of different experiences would prove challenging.

We identified surprise as a key component of the success of the original Continuity. Players are surprised when confronted with puzzles that they cannot solve without expanding their understanding of the game. We decided surprise should be one of the key components of Continuity 2 and that levels should introduce the player to surprising scenarios rather than repeating similar-seeming puzzles. This concern seemed to motivate the addition of new rules and elements to the game because new features would more easily surprise new and old players alike.

### 5.3.2 Smooth Play

One of the primary problems with the original Continuity is the manner in which the player's interaction with the game is broken and fragmented. Our experience in observing players play Continuity informed us that players generally only use the micro-mode of play for very brief periods. They spend the majority of their time examining the level from the macro-perspective.

When the players do take control of the character in the micro-mode, the control session is brief as they generally only move the character a short distance from one tile to the next. After moving the character to an adjacent tile, they almost immediately switch control back over to the macro-mode in order to reappraise the character's location in the level. It was very rare to see a player move the character across two borders without first using the macro-mode to affirm that the character was moving the correct direction.

Players never voiced this concern to us directly, but based on observation we became aware that they quickly lost track of the character's position as soon as they took control of the character in the micro-mode. Our major concern was that this disjointed control of the character made the whole experience of moving the character feel like excise. This feeling of excise is reinforced by the lack of rewards and the players inability to carry out challenging tasks while controlling the character.

We decided that one of the major goals for Continuity 2 would be to reduce the feeling of excise in the character movement. We wanted to create a smoother play experience for the player where he or she did not need to transition between play-modes very frequently and could spend more time in each.

### 5.3.3 Interruptability

One element that is important to both casual and mobile games is that of interruptability. The original Continuity does not accommodate interruption, which makes it less accessible to casual players that may not play for long play sessions. The player's progress is only saved between levels. The player cannot elect to play levels arbitrarily. Most importantly, many levels take a great deal of time to complete. Some levels have a mean completion time of over an hour. If the player quits the game at any point within a level his or her progress within that level is lost and he or she will have to restart the level from the beginning. We believed the granularity of the original game's save system was far too coarse to truly suit casual or mobile gamers.

Even if players were able to save arbitrarily within the original Continuity, the game would likely still not be interruptable enough. Many of the levels in the original Continuity present the player with long sessions between achieving any sort of measurable progress. If the original Continuity was playable in a mobile environment, where players play for brief play sessions, a player might play an entire session without making any discernible progress as the only indicator of progress was the number of keys the player had collected in each level. However, collecting a key is often a major milestone that takes occurs fairly infrequently. This lack of reward would likely be compounded as players might spend much of a play session attempting to remember what they were trying to accomplish during the last play session.

We felt that Continuity 2 should possess more and clearer subgoals as

a way to allow players the ability to feel rewarded even within a short play session. More subgoals would also allow players the ability to resume play more easily when beginning a new play session as they would be able to more clearly set a goal and then attempt to achieve it.

## 5.4 Survey

After clarifying the intent of the project by setting explicit goals, we still felt there were many questions left unanswered. Some of these related to what players especially liked about the original game and wanted from a potential sequel. Other questions related to how players approached playing the game.

This section discusses the creation and carrying out of an online survey.

### 5.4.1 Motivation

We previously defined what it meant to be a Continuity game (Section 5.1) as well as set goals for what we desired from Continuity 2 (Section 5.3) for iPhone. As we began considering modifying the game mechanics of the original Continuity, we realized that much of our analysis was based directly on our own opinions.

We were concerned that without getting outside input, we might take emphasize or remove entirely the wrong parts from the original game. Before starting down the wrong path, we wished to test the validity of some of our assumptions and probe for further information.

### Level Variety

One of the primary motivating factors for conducting a survey was trying to understand to what extent fans of the game were satisfied with the variety of levels of the previous game. Since we created the levels for the original Continuity, we did not obtain the same experience as other players when playing the game. Further, we had played and observed the game so many times that we had lost some perspective on what aspects were surprising and fun.

We believed that the shorter levels that were overcome primarily through a flash of understanding were the best levels in the original Continuity. The longer levels that players solved incrementally appealed to us less and seemed to be less exciting to players. However, these longer levels do not provide such a singular moment of discovery, so it is possible players enjoy the entire long experience but have no moment around which to outwardly express their enjoyment.

We thought these long, visually complicated levels felt too similar to one another while lacking surprise and variety. It was necessary to consider that perhaps players felt differently. We wanted to discern whether players felt such levels became repetitive and boring or whether the players were excited to play more of these maze-like levels. Of course, we also needed to account for the possibility that players liked all levels in moderation and appreciated a variety of level types.

During development of the original *Continuity*, we found these long, complicated, maze-like levels proved easy to create in relation to other difficult, yet visually simpler levels. If players indicated that they simply wanted more intricate mazes for *Continuity 2*, our time developing the sequel could focus around trying to design such levels as opposed to modifying the design in any radical manner. This would be an easier approach that we did not want to overlook before learning more about players' opinions on *Continuity's* level variety.

### **Play Styles and Character Movement**

We never truly discussed or explored how players attempted to solve levels during the development of *Continuity*. It seemed to us that most players were hesitant to move the character until they were assured that they could move it somewhere meaningful. Usually, this meaningful place would be the location of either a key or a door, but might also be some other place that they somehow (perhaps incorrectly) established would somehow lead the character to reach a key or a door.

We believed that by better understanding how players set goals for themselves, we would be able to create better levels. In the original *Continuity*, most of the structurally complex levels that contained multiple keys could be optimally solved without backtracking. If the player left this pre-defined path, he or she was punished by needing to spend a lot of time to return the character to a previous location. Many of the levels in the original *Continuity* offered the player many opportunities to set themselves back by making the character take such a wrong turn. The player would then be required to spend much time to return the character to the location of the wrong turn.

In order to enhance the experience of moving the character, we wanted a better understanding of how players felt about moving the character. We suspected that the punitive nature of making a wrong turn discouraged players from moving the character until the moment the players were absolutely sure they could navigate to a goal. However, we desired more insight into how players formed and attempted reaching such subgoals.

### **Redefining *Continuity***

We wanted to use this opportunity to further clarify some of our assumptions about what aspects of the original game players felt were intrinsic to the experience of a *Continuity* game. Our previous defining of the core concepts was based almost exclusively on our own feelings. We worried that there was a great risk of focusing on the wrong parts of *Continuity*. The two major concerns we had with our original definition related to the game's minimalism and the game's aesthetic values.

To us, the minimalistic nature of *Continuity* was a core aspect of the

game. We had established that we believed that the original game’s minimalist gameplay and appearance were a requisite part of creating a Continuity game. Upon further reflection, we decided that this might not be the case. Perhaps players desired more activities. Perhaps players wanted a high-pace, dexterity-based platformer within the context of an  $n$ -puzzle. We desired a better understanding of how much minimalism players wanted from a Continuity sequel.

We also wanted to learn about the gameplay aesthetics that players identified as inherent to Continuity. Perhaps players identified aspects of the game, which we failed to recognize and believed these aspects were especially relevant to their enjoyment of the game.

#### 5.4.2 Conducting the Survey

We created a survey consisting of 16 questions and an open-ended question, soliciting any extra comments or advice. We wished to keep the survey brief to encourage responses. Of the 16 questions, 14 were multiple choices or multiple selections. Only two of the questions and the space for extra comments encouraged the respondents to actually come up with their own answers.

The survey was placed on the webpage on which the original game was hosted. The survey was placed directly beneath the game on the page. The survey was available for five days (from February 26, 2010 to March 2, 2010). During this time, 1,165 respondents participated.

It is important to note that in addition to those who simply happened to visit the game’s webpage during the time of the survey, we used Facebook and Twitter to encourage people that had likely played and enjoyed the game to participate. We realized that the people who viewed our Facebook and Twitter pages would likely have played and enjoyed the original game and therefore their input would provide insight into the specific group of players that enjoyed the original.

During the surveying period, the page that hosted the game and survey received 49,510 visits. Some of these were likely visiting and participating in the survey directly as a result of our requests via Facebook and Twitter. As a result, it is likely that the survey has a more positive bias than it would have if the respondents were simply those that happened to visit the game during that same period.

#### 5.4.3 Survey Results and Conclusions

The vast majority of the 1,165 respondents indicated that they enjoyed their experience playing the game, however 37% indicated that they only played for between 0 and 15 minutes. The positive responses to inquiring to what extent players enjoyed the game might be biased by the colloquial phrasing

of the options. Players were asked, “What do you think of Continuity,” and then prompted to respond with a score from one to five. A score of one represented, “It’s shit,” whereas five represented “It’s the shit.” We used the informal tone to encourage responses to the survey, but it clearly might have further biased the results.

In general, the survey was helpful but not overly so. Our analysis of the results left many of our questions unanswered.

### **Level Variety**

We were surprised to find that very few players felt that the game’s levels became uninteresting as the game progressed. We perceived that the levels at the beginning and end of the game were fairly repetitive. A large majority of players, however, found that the original game’s levels remained interesting. Further, very few respondents indicated that they actually skipped levels or quit the game due to boredom or a lack of interest in the levels.

Many players indicated that they enjoyed the complex mazes, but that they were not what players enjoyed the most. Respondents seemed most fond of the trick or gimmick-based levels. From this we concluded that the sequel should not focus strictly on complex mazes or avoid them completely. Instead, the game design should allow for many trick-based levels with some complicated mazes mixed in sparingly.

### **Play Styles and Character Movement**

The survey results reaffirmed our belief that very few players enjoyed moving the character. However, there was no true consensus from the results as to what the primary problems were with moving the character. It seemed that many people have many different things they liked least about moving the character. This question should have been preceded by a simple question asking the player to rate the character movement experience.

We received a fair amount of feedback regarding the implementation of the character controls. We already believed that there was a problem with the implementation and that the character did not possess enough friction with the ground and was too “slippery”. As a result, we were more interested in the concept of character movement, in general, rather than the actual specifics of how the player controlled the character.

We were not able to draw any strong conclusions regarding players’ opinions about moving the character. Perhaps every person found part of the character movement problematic or perhaps few people found every aspect problematic. Without any clear evidence that people specifically enjoyed the character movement, we decided to operate under our previous assumption that the problems we previously identified with character movement (the slowness, the lack of meaningful activity, the ease of getting lost and

the excessive penalty of moving the character the wrong way) were all valid concerns as some people identified each as problematic.

The fact that players did not especially enjoy moving the character was reinforced by their play styles. Only a very small minority claimed that they used the character movement as their approach to completing levels. 43% of respondents indicated that they primarily analyzed the level from the macro-mode while 53% indicated that they used both modes equally.

We were also unable to determine any consensus in terms of how players identified subgoals. It was clear at least, that a majority of players do create subgoals rather than simply having the character wander through the level until it gets close to some key. Only a minority, however, tried to plan a path through the level past all keys. From these results we concluded that levels in Continuity 2 should not try to only force players down a singular path to solve levels.

### **Redefining Continuity**

Players indicated that they enjoyed the minimalism of the game. They also indicated that they would either enjoy or otherwise not mind additional gameplay elements being added to the game. We believed this served as validation of the game's minimalism as being important but allowed for the Continuity 2 to be somewhat less minimalistic than the original.

The survey results also indicated that a significant number of players enjoyed the dual gameplay modes, however, it was not overwhelming. Given that few indicated that they enjoyed moving the character, we felt that it was possible that a game might still feel like Continuity, without the player directly controlling the character or otherwise without two mutually exclusive control modes.

Lastly, we found no meaningful results in terms of gameplay aesthetics. In order to discern what aesthetics players enjoyed, we asked the survey takers to list other games that provided similar types of enjoyment as Continuity. The results for this question were inconsistent. Many people simply listed other "indie games", web games or games that were visually similar to Continuity, but which had no other meaningful gameplay relationship.

We had avoided asking players directly to list the aesthetic values that Continuity appealed to, because we felt players would likely not have such a vocabulary. Further, we did not list possible options so as not to bias the result. Our hope was that players would list games that affected them similarly and that we would be able to analyze those games and determine the similar appeal. Unfortunately, given the lack of common responses, this was not possible.

## 5.5 Re-planning and Redefining the Project Goals

By the time we began designing for the first playtest, half the time allotted for the project had already passed. We were already well behind our schedule as it took us much longer to implement prototypes to validate our technical concerns and also produce a working software foundation for the rest of the game’s development. During this phase we became aware the time remaining would not be sufficient for us to create a finished game while still exploring the design space.

In addition to needing to design and evaluate new gameplay mechanics, a major obstacle that we predicted would stand in our way was the creation of enough high-quality, unique levels. Another major impediment was the designing and development of out-of-game interfaces and menus.

Given that we felt the quality of the final game was more important than its timeliness, we decided to modify the focus of our work. Since we felt we would not be able to create a high-quality game within the remaining time, we redefined our objective for the project. Instead of trying to create a completed game by May 1st, 2010, we would try to produce the best prototype that we could by that date.

Since we no longer needed to have anything in particular completed by May 1st, we decided that trying to create an all-encompassing time plan would be unwise. Instead, we decided that we would simply engage in development iteration based on playtesting. This process contains five steps:

1. Generate potential solutions to the game’s primary risks and problems
2. Evaluate and refine these potential solutions
3. Prototype some of the suggested solutions
4. Use playtesting to evaluate the design decisions
5. Repeat a new iteration from 1 using data collected from the playtests.

Such a system allowed us much more flexibility, as it would allow us to accommodate unexpected problems or successes. We were concerned that if we had re-planned the remaining development time that poor game design decisions would be cemented into the final design due to a lack of time to attempt alternative solutions.

This workflow is in contrast with how we developed the original Continuity. During Continuity’s development, we preferred producing a finished game to a more feature-rich or otherwise possibly superior prototype. In that context however, we had no specific intention of resuming work after the project deadline. This effectively only allowed us modify the previous work for the game, rather than trying radically different things. For Continuity 2, we wanted to allow for the possibility to completely throw away

work and more fully explore the design space, which is something we never did in the original. We felt that progressing on a fixed time plan would hinder our ability to implement alternative solutions to problems as they arose.

## 5.6 Iteration 1: Touch-to-Stop and Tacit Controls

After conducting the survey, we found that the character movement was still the largest risk. The foundation that we created used on-screen buttons to switch between the micro-mode and macro-mode as well as move the character. While this worked, in that it allowed someone to play the game, we were not satisfied with the solution.

To develop a better movement system we first analyzed the problems with Continuity's movement system. We then brainstormed what other actions could be given to the player or character to improve the character movement. Finally, we tested two different movement systems.

The first prototype used the touch-to-stop system, in which the player controls both the character and tiles, simultaneously from a fixed perspective. The intent of this system was to encourage smoothness by allowing the player the ability to quickly switch back and forth between control modes.

The second prototype used the two-mode tacit controls movement system. This system was a two-mode system similar to the original Continuity. When in the micro-mode the player could use gestures to move the character without using any visible user interface elements. This prototype also allowed for the player to rotate the device in order to modify the gravity direction relative to the game world.

### 5.6.1 Analyzing Continuity's Character Movement

After analyzing the character movement system from Continuity, we came to four major conclusions:

1. Moving the character not allow enough interesting options and controlling the character does not influence the game world.
2. The process of moving the character is done in too many brief stints because the player constantly shifts perspectives determine the character's position in relationship to the other tiles and the rest of the level.
3. There is too little enjoyment present in character movement considering the great amount time spent engaged in the action.

Additionally, we desired that moving the character across a single tile would pose a challenge. However, this goal was dropped upon further analysis of restraints of the iPhone's display. The resolution was simply too limited to place a variety of obstacles across a tile and still have them all be visible while zoomed out.

## **A Lack of Actions**

We felt that the character movement was not fun because the player has no incentive to make any interesting decisions or the ability to demonstrate any skill while moving the character. Since the levels in the original *Continuity* were designed such that making a single wrong turn could set the player back by several or more minutes of progress, the player was discouraged from exploring alternate paths or making any meaningful decision or action while in the micro-mode.

Since the character's only abilities were running and jumping, players were left simply carrying out pre-determined moves in the micro-mode with little room for improvisation. This generally took the form of periodically pressing the jump button while simply holding another button to make the character run in a certain direction.

One conclusion we drew from the game *Paper Cakes*, was that the player is still more engaged when directly controlling the character than when waiting for the character. In *Paper Cakes*, the player draws a route for the character to move. If the route is valid, the character carries out the move while the player observes. If the route is invalid, the player immediately receives feedback notifying him or her and the character does not move. While the character is moving, the player can only modify or cancel the movement, but can perform no other actions.

Our belief was that this aspect of waiting on the character to complete a move that is already guaranteed to succeed resulted in a slower, less engaging experience. Even though the player could not affect much while moving the character in *Continuity*, we felt that the direct control kept the player more engaged. In *Continuity*, the player's ability to make the character jump, even if it accomplished no real gameplay objective, kept the player more involved with the game than the player was in *Paper Cakes*' scheme of directing and waiting.

We concluded that to keep the player engaged, he or she required some ability to constantly control some aspect of the game. We thought a direction or routing system similar to *Paper Cakes* could work as long as the player was able to interact with the game in some other way while the orders were being carried out.

## **Disjointed Movement**

As has been discussed previously, a major concern of ours was that, in *Continuity*, the player finds him or herself unsure about the relative location of the character in the constantly rearranging world. This became a worry to players because the risk of moving the character to the wrong place was too great. Players were constantly checking the character's position as they could effectively lose minutes of progress by making the character take a

single wrong turn. This constant reexamination of the character's location expressed a paranoia that we found undesirable.

*Closure* is another game that has players repeat character movements, which are not rewarding by themselves and afford little room for improvisation. In *Closure*, the player places light-emitting orbs into orb-holders, which then may move. As these orbs move, the level's physical structure changes as different areas are illuminated. Many levels require that the player undertake a certain action when these orbs are at specific points along their paths.

Oftentimes, if the player errs in controlling the character as the orbs are moving, he or she must start the orb movement process again. This process contains two types of waiting. First, the player must wait for the orb-holder to move back to the starting position. Second, the player must again wait for the orb to reach the critical point along its path in order to retry the failed action. Since the player can fail due to mistiming or incorrect input, the player might retry a specific action many times with a high time penalty for each failure.

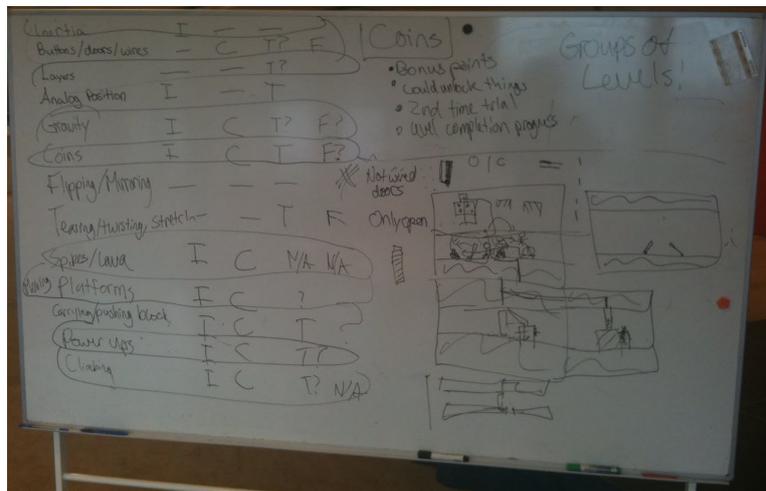
In both *Continuity* and *Closure*, this punishment of needing to repeat actions becomes tiring as players find their characters setback to previous positions. Since both games are based more upon the puzzling aspects than rewarding dexterity, it becomes frustrating when repeating previously taken actions. We hypothesized that the main cause of the disjointed, constantly zooming play style was this punishment. Without the threat of having to repeating sections of the level, players would feel more free to explore and less fearful of controlling the character for longer periods.

### **Taking Too Long**

The consequences of players' disjointed, constantly reappraising play style and inevitable setbacks were that players took a long time to move the character from one place to another. Given that *Continuity*'s character movement was not inherently enjoyable, the time required to move the character from one place to another was too long. This was not necessarily a problem of the time it took to move the character, but of the density of enjoyment over the required time.

Since the player does not get to make interesting decisions while moving the character, the engaging moments of the game occur before and after movement occurs. It is at these points that the player needs to imagine the character moving through the level, establish a new subgoal and try to create a new plan. *Continuity*'s movement system placed these opportunities too far apart.





**Figure 5.8:** From the large list of gameplay elements and mechanics we sifted out the best ideas for further evaluation.

The concepts that we found to be the best and went on to prototype are presented in the remainder of this section.

### Bonus Items

One way we thought we could improve the character movement experience was to simply provide additional rewards and “juiciness.” This would have the added effect of making the game appeal more to casual players. There were two main ways we considered rewarding the player: providing more clear subgoals that the player could achieve through moving the character and making the action of moving the character more rewarding.

We considered optional bonus items to be a good way to provide such extra rewards. In the context of *Continuity*, we decided that these items could take the form of coins, as we predicted that players would quickly recognize them from games such as *Super Mario Bros* as not being required to complete a level. In *Super Mario Bros*, players are rewarded with an extra life if they collect 100 coins. For *Continuity 2*, we sought some similar explicit way to reward for players for collecting the bonus items.

We did not determine any specific gameplay reward that *Continuity 2* would provide for players that collected bonus items. Instead, we considered various ideas. Perhaps the player could unlock new levels or parts of the game by collecting the coins. Alternatively, in addition to level progress, the player could track his or her collection of bonus items. This would present optional difficulty for players, hopefully making players better able to customize the game to their own desires.

In addition to any explicit reward, we wanted the collection of any bonus item to possess an implicit reward. First, the action of collecting the coin

should produce a “juicy” sound. In *Super Mario Bros*, the pleasant sound of the coin being collected makes the experience of collecting it more rewarding. Second, we thought that collecting these bonus items, as well as the mandatory keys, should contain a similarly juicy visual aspect. In the original Continuity, the keys simply disappeared, leaving a shadow, when collected. *Crayon Physics Deluxe*, on the other hand, provides players with a pleasing animation of the star (the goal item) rising upwards and fading out when collected. Lastly, we thought that by simply placing these elements slightly out of the way of the character’s normal path through the level (perhaps so that it was required to jump) the player would have some more interesting tasks while moving the character.

In addition to the keys, these bonus items could also function as checkpoints. This would help alleviate some of the game’s punishment as providing players more checkpoints would help prevent players from losing as much progress whenever the character died.

### **Levers, Wires and Gates**

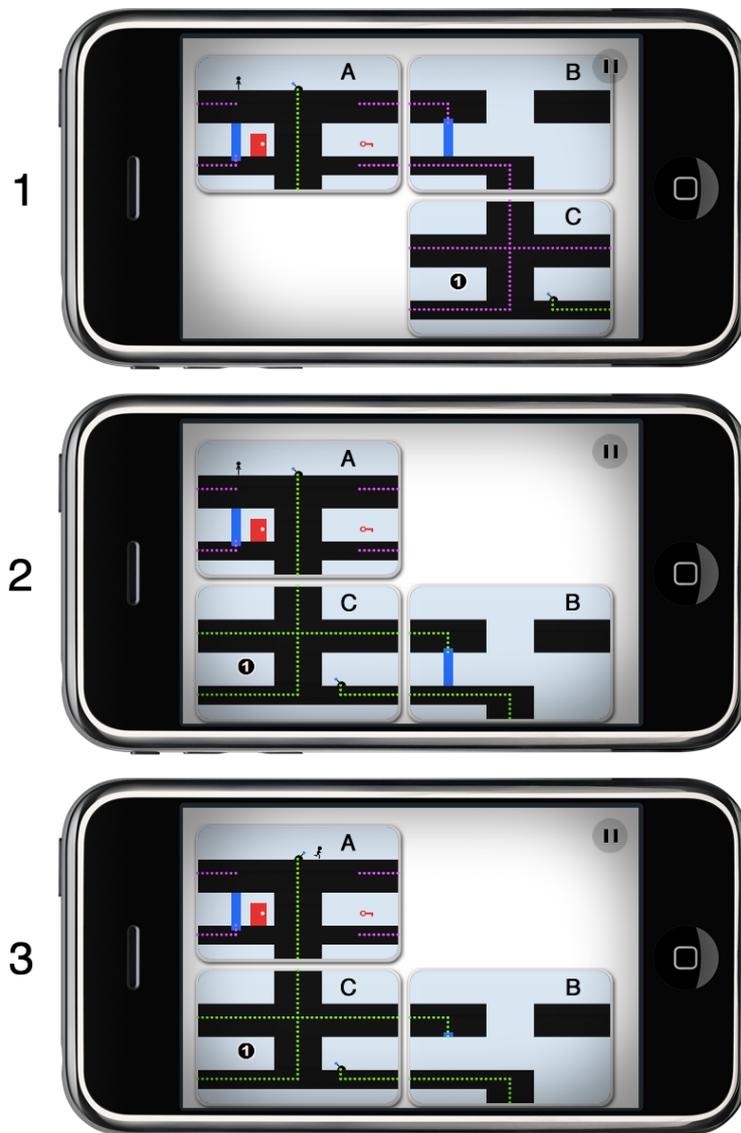
In the original Continuity, the character had no ability to influence the game world. Controlling the character felt like a secondary concern, as the player could modify the game from the macro-mode but not the micro-mode.

During development of the original Continuity, we had discussed the concept of gates and some sort of trigger to open them. These gates would serve as a wall or floor segment that could be opened and closed and so were distinct from doors, which serve as a goal location. We considered gates a simple way to increase the influence the character had on the game world.

For Continuity 2, we wanted for the triggers to not work universally on all gates. Further, as we had established previously, we wanted the player to be able to know, without experimenting, which trigger would activate a given gate. Therefore, the triggers and gates needed some way to denote their pairing.

We conceived that wires could be a way to literally connect the trigger to the gate. Expanding upon this idea, we found that such connections would allow for a new aspect of puzzling.

In this proposal, wires segments would connect to triggers, gates and other wire segments. Any wire segment that would be attached by a path of other wire segments to a trigger would be energized. When a gate would be attached to a trigger by a path of wire segments, the gate would be energized. When the player activated the trigger, the energized gate would open or close. These rules allowed for a many-to-many relationship (i.e., each trigger could connect to any number of gates and each gate could connect to any number of triggers). At any point, a wire segment could connect to any number of other wire segments, so a trigger might energize any number of gates simultaneously. For an example, view Figure 5.9.



**Figure 5.9:** An example of the levers, wires and gates from the third prototype. The player needs to rearrange the tiles in order to energize the gate in tile *B*. Once the gate in tile *B* is connected to the lever by a series of wire segments, the character can flip the lever to open the gate.

We found two major benefits from this system of triggers and gates. First, as discussed, it allowed the character to influence the game world, which we hoped would make character movement feel more important. Second, the wire connection rules made the game feel more like an  $n$ -puzzle. In the original *Continuity*, a player never needed to be concerned about the position of more than two tiles at once. This system of wires connecting triggers to gates would allow for us to design levels that required the player

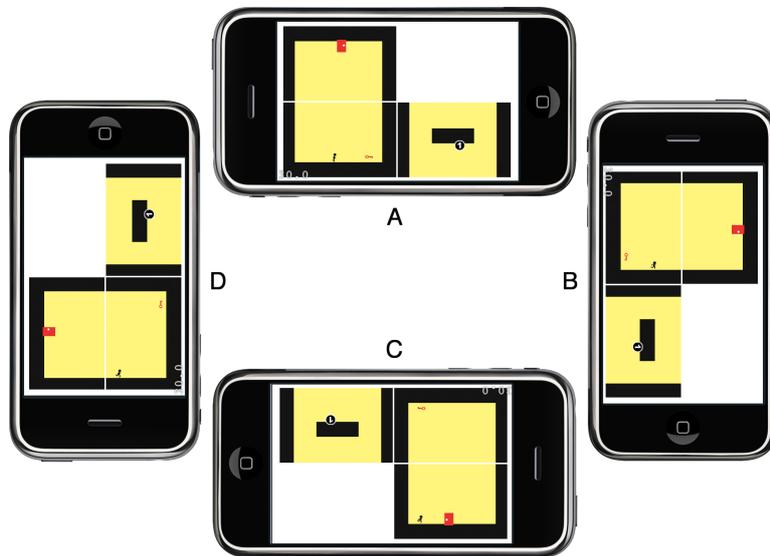
to consider the location of several tiles, simultaneously. Our hope was that this would strengthen the  $n$ -puzzle analogy.

It was decided that the triggers would take the form of levers. These levers would only activate gates when they were flipped. The player could flip a lever by moving the character across it. Gates would change state between open and closed whenever a connecting lever was flipped, regardless of the direction of the flipping. The manifestation of these rules lead to many further puzzle scenarios.

### Rotation

Another gameplay concept we had considered, but ultimately not pursued, while developing Continuity was that of rotating tiles or the entire level. On the iPhone's display, tiles were simply too small to allow each to be rotated via an intuitive finger-twisting gesture. However, using the iPhones' accelerometers we supposed players could rotate the entire device to rotate the game world's gravity. We were apprehensive of how players might feel about such physical interaction.

Players would be able to rotate the device into four different orientations as in Figure 5.10. The tiles and game world would stay fixed, relative to the device. However, the gravity of the game world would change such that it approximated down in the real world. That is to say, by holding the device upside down, the character would fall onto what used to be the ceiling.

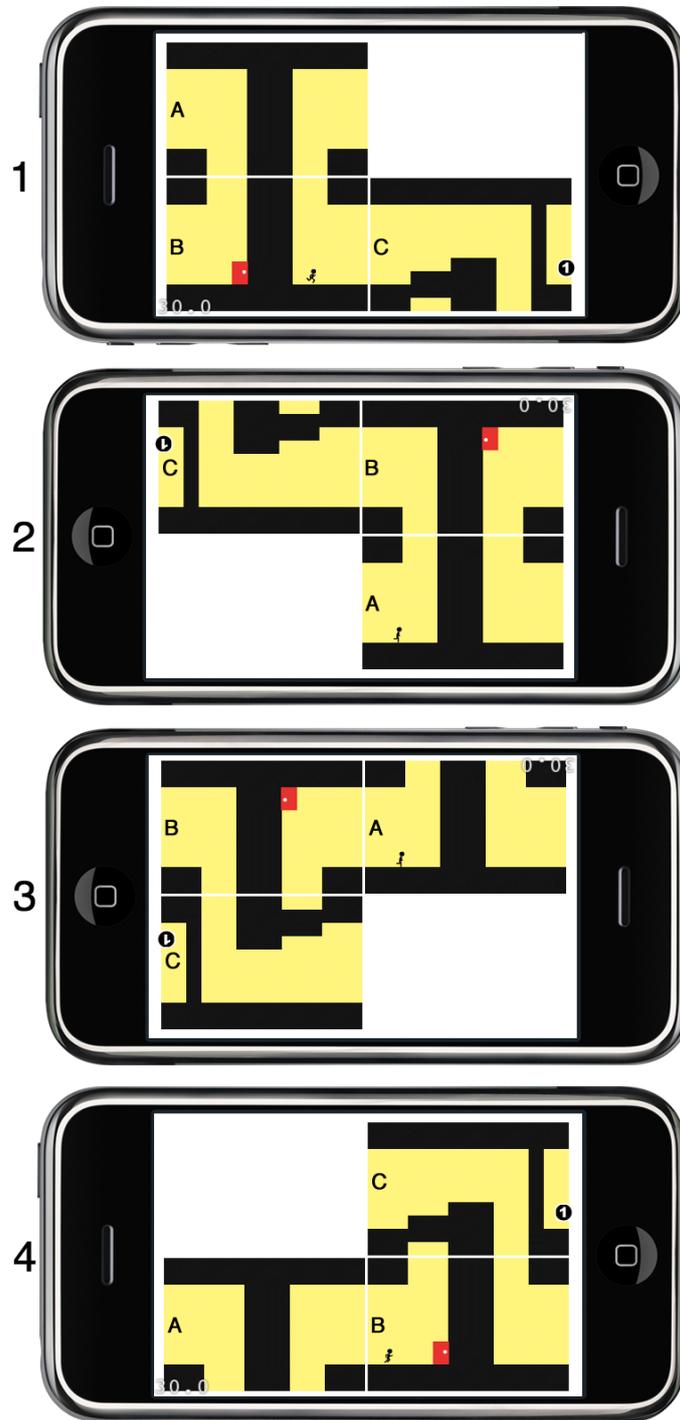


**Figure 5.10:** The rotation controls allowed for gravity to change depending on the orientation of the device. **A.** The device is held in the standard orientation. **B.** The iPhone has been rotated 90° clockwise. **C.** The iPhone has been rotated 180°. **D.** The iPhone has been rotated 270° clockwise.

We only allowed for four directions of gravity, as one of our primary concerns with including such a feature was that players might cease to view the game as a platformer. Instead, if the player could instantly change gravity to any arbitrary direction, players might play the game as a dexterity-based, marble maze game.

Even given only four possible gravity directions, the player was freer to move the character through level. Without constant gravity and a limited jump height, many of the levels from the original *Continuity* would be much easier. We felt this easiness was a double-edged sword as making traversing spaces easier would lessen the punitive aspects of taking the wrong path. For example, if the character fell down a shaft and the player later wanted to undo this action, he or she could simply rotate the device and have the character fall back down that same shaft to where it originally started. Our hope was that we would still be able to create levels with non-obvious paths.

From the beginning we desired the player to discover his or her ability to rotate the game after having played for some time. In the original *Continuity*, players seemed to truly enjoy the moment that they discovered that they could pause the game world by switching control to the macro-mode. We wished for returning players to experience a similar sense of discovery when they learned about their ability to change the game's gravity.



**Figure 5.11:** An example of using rotation to solve a puzzle. **1.** The character needs to move to the door. **2.** The player rotates the iPhone 180° and the character falls into tile A. **3.** The tiles are rearranged. **4.** The iPhone is rotated back to the standard orientation and the character reaches the door.

## Deemphasizing Controlling the Character

One of the more radical ideas we generated was to remove Continuity’s dual perspectives. By using the *Removing Mental Blocks* method [26], we minimized the character control as part of the game. Instead of having mutually exclusive control modes, the player could instead control both the character and the tiles simultaneously from a fixed perspective. This one-mode control system would hopefully allow for smoother play as the player could more easily keep track the character and its relationship to other tiles. The player would also be able to instantly switch between controlling the character and moving the tiles instead, so players would be able to play at a greater pace.

In this one-mode control system, the player would still move the tiles by dragging them. Some other means, such as on-screen buttons, would be used to control the character.

We considered removing the player’s control of the character entirely. In such a concept, the character would simply move given some simple rules and the player would attempt to rearrange the environment in front of the character so that it did not die. *Lemmings* also used a similar environmental manipulation mechanic where the player tries to assure the safety of autonomously moving units. Ultimately, we decided we wanted to preserve the player’s ability to control the game on two levels.

It was not hugely important to us that the player directly controls the character. One version of the one-mode control scheme we developed was called touch-to-stop. In this version, the player simply directs the character to move left or right. The character continues this action until the player commands it to stop moving or move in the other direction. The player can also cause the character to jump at anytime.

The touch-to-stop concept was developed to increase the pace at which players play and to allow for experienced play. Such experienced players would direct the character to begin moving and then rearrange the tiles such that the character would uninterruptedly move through the level. Such uninterrupted character movement would encourage near constant thinking and planning, as the player would need to constantly mind both the character’s ever-changing position and the layout of the tiles. We believed the possibility of this experienced level of play would allow for speed runs and increase replayability. In order to prevent this constant pressure from being overwhelming, the game world would pause whenever the player touched a tile. If the player simply wanted more time to think, he or she could just touch and hold on any tile.

### 5.6.3 Developing Two Versions

We possessed uncertainties with how players would react to the tacit controls and the rotation mechanics. We thought that these concepts were not necessarily complimentary as one-mode gameplay did not provide the player a safe mode from which to rotate and view different orientations of the game. In one mode play, if the player were to rotate the device, the character would immediately fall due to the change in gravity. With two-mode mechanics, the player could always zoom out to macro-mode (where the game world is paused) in order to rotate the iPhone. Although we could have designed further to integrate rotation mechanics with one-mode gameplay, we decided to get to prototyping quickly, to see if players were even interested in the concept of one-mode gameplay.

For this design iteration, we decided to pursue touch-to-stop instead of one-mode direct control. Touch-to-stop was riskier than the direct control one-mode concept and a greater departure from the original game. We also thought it could be the better solution as it would provide a distinct experience to returning players while still maintaining enough aspects of the previous game. However, players might find the concurrent nature of controlling both the character and tiles to be overwhelming.

We also thought that the rotation controls allowed for a great number of possibilities if players demonstrated that they enjoyed the interaction. In order to give the player some feeling of safety, we added the rotation mechanic to the two-mode controls of the original Continuity.

Rather than only developing one of these concepts, we conducted work on both in parallel. This allowed us to make a more informed decision about whether to further pursue one-mode or two-mode alternatives.

We felt that both of these options were highly dependent upon their implementations. Specifically, our primary concerns were how players would react to the real and virtual interfaces required to control these game mechanics. We decided to pursue prototyping these options on the iPhone using the foundation we previously developed rather than using paper prototyping or other methods that would not accurately reflect the actual interaction.

For both prototypes, we decided to implement coins, as they could be quickly and easily added to the foundation. The gates, wires and levers were reserved for a future iteration, as they would require more complicated logic and art to prototype.

### 5.6.4 Designing the Character Movement Systems

We felt that the interface was highly coupled to the gameplay. In the original Continuity, ideas were abandoned simply because we could not produce elegant interface designs for them. For Continuity 2, we wanted to focus on ensuring that the gameplay and interface suited each other. That is, we did

not simply want to design gameplay mechanics without considering how the player could interact with them. Therefore, we set about creating separate interfaces for both the two-mode control version and touch-to-stop version of the game.

The UI for the two-mode version of the game is notable in that the player is only presented with a single pause button. This system is called tacit-controls. Rather than pressing or otherwise interacting with visible elements on the display, the player touches regions of the display and uses gestures to control the character.

The GUI for the touch-to-stop mode uses on-screen buttons for character movement. Two buttons are used to cause the character to begin walking in a given direction or stop. Another button is used to make the character jump.

In both game versions, the player moves the tiles by dragging them.

### **One-mode: Touch-to-Stop**

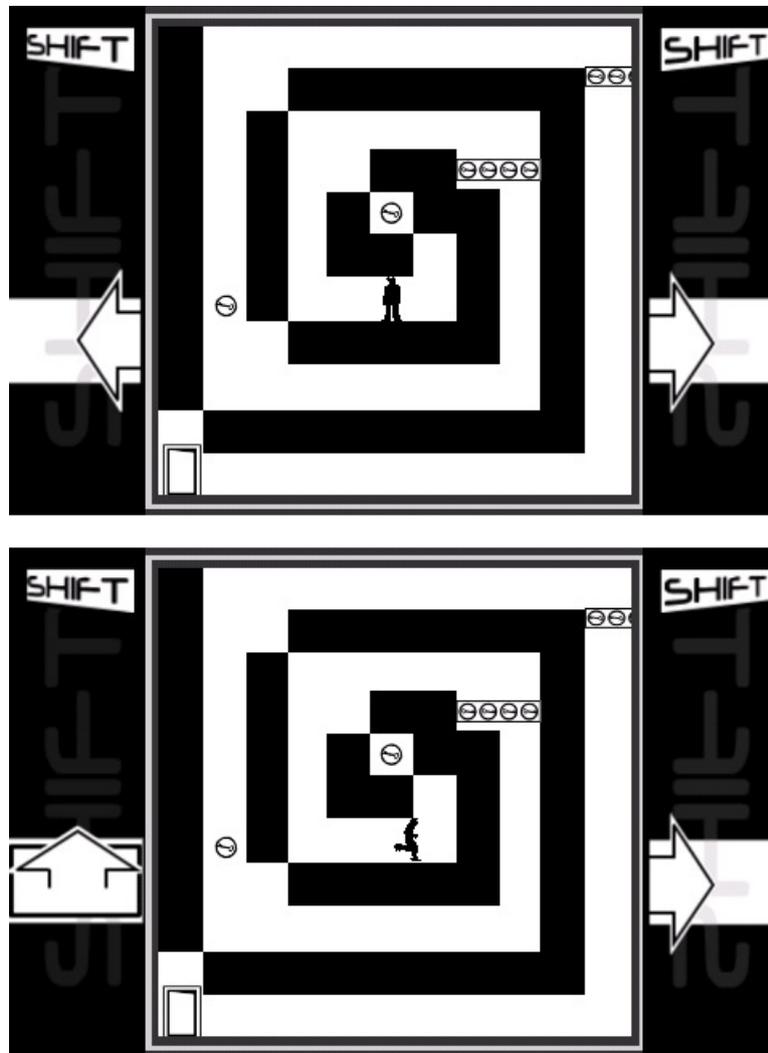
The main problem with one-mode gameplay was that the player needed a way to unambiguously input a command to move either the character or the tiles. Since we wanted the game to feel like a player was holding a physical  $n$ -puzzle, moving the tiles needed to be accomplished in a simple, direct way by physically dragging the tiles.

*Kami Crazy* uses such an intuitive gesture-based control scheme. However, there were problems incorporating such gestures into *Continuity 2* since touching and moving fingers over the tiles is already reserved for tile movement. The game could have recognized a multi-touch gesture as pertaining to the character, however we wanted to allow the player to hold the iPhone as if it was a gamepad and use only his or her thumbs if he or she preferred. It is challenging to perform multi-touch gestures with only one's thumbs on the iPhone.

Since these gesture-based control schemes did not work well for both character and tile movement, we decided to use a virtual button-based control scheme for the character movement.

The iPhone game, *Shift*, uses virtual direction buttons for character control, although it places them beside the game world. The player controls the character directly using these buttons. One important aspect of *Shift's* buttons is that they are context-sensitive. When the player holds a button to move the character a certain way, the other movement button changes into a jump button. Pressing this button will cause the character to jump in the direction it is currently running. This system saves more space on the screen. When the player is moving the character to the right by holding the right button, the left button would be of no use. By changing the other movement button to a jump button, the UI uses less space and is more visually balanced than if three buttons (left, right and jump) were always

displayed. This design works well as the character is never required to jump while standing still.

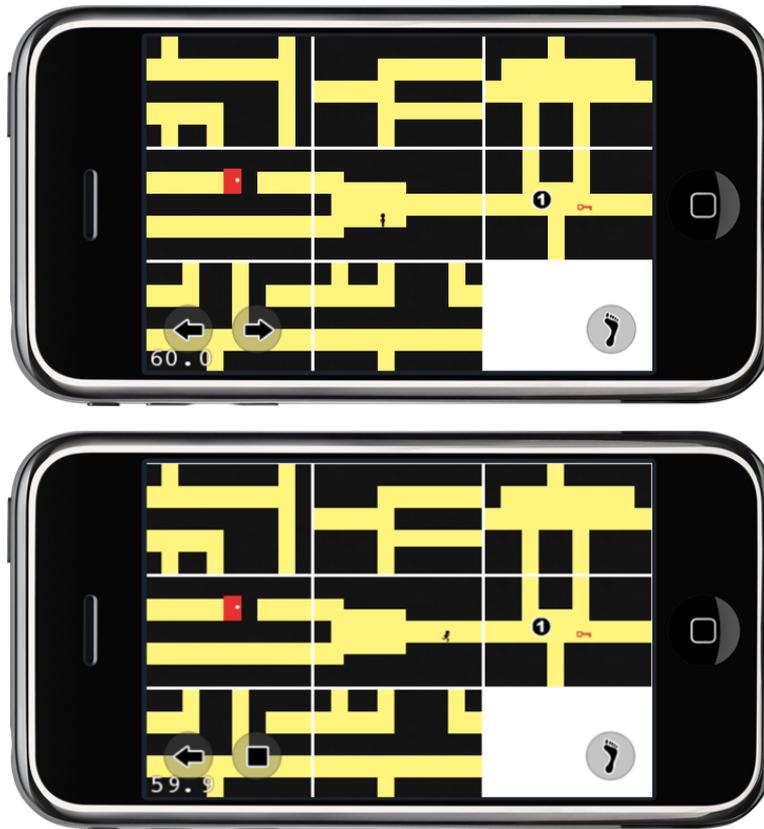


**Figure 5.12:** Images of the iPhone game *Shift*. **Above:** The character is standing still and buttons are presented to move in either direction. **Below:** The player is pressing the right arrow to cause the character to run to the right; the other button now causes the character to jump.

We decided to borrow from the virtual gamepad pattern and use aspects of *Shift's* context-sensitive buttons for the touch-to-stop prototype. The touch-to-stop prototype placed the buttons along the bottom of the device with three buttons. Two buttons along the left side controlled horizontal movement. Another button on the right side controlled jumping.

When the character stopped, the two left-side buttons took the form of

arrows. When either was pressed, the character would move in the respective direction until given other directions. Once moving, the movement button corresponding to the character's movement direction would become a stop button. If the player pressed the stop button, the character would halt immediately. If the player pressed the other movement button, the character would change directions and the buttons would change accordingly. The jump button would always be available to the player and the character would simply jump in the direction it was already moving, or straight up if the character was standing still.



**Figure 5.13:** An example of character movement in the touch-to-stop prototype. **Above:** The character is standing still. Buttons are presented for left and right movement. **Below:** The character is running to the right. The right running button is changed to a stop button.

To better facilitate the player switching focus between controlling the character and controlling the tiles, the game world paused whenever the player touched a tile. These touches paused the character's movement, without commanding the character to stop. The game world would pause for 500 milliseconds after such touches ended. The delayed resuming allowed players to rearrange many tiles without having to worry about the character

moving. Of course, a player could simply hold a touch on any tile if they desired more time to think. Our hope was that this would minimize the player's button presses and result in a smoother experience.

### **Two-mode: Tacit Controls and Rotation**

The foundation that we built contained a two-mode GUI. A button containing an image of a magnifying glass was used to switch modes and alter the game's perspective. When zoomed out, the player would drag the tiles directly via touch. When zoomed in, the player controlled the character using two buttons that controlled its running direction and another that was used for jumping. These controls worked adequately as a player could competently play.

While a player could competently play with these on-screen buttons, we felt that another approach might be more optimal. Games such as *Spider: The Secret of Bryce Manor* and *Jelly Car*, used interfaces that only contained diegetic elements. These sparse interfaces more closely matched the visuals of *Continuity*. Additionally, these interfaces worked for users that wished to hold the iPhone as a gamepad as well as for those that wished to support the phone in one hand and use the index finger of their other hand. For these reasons, we believed that the character controls should use as few visual, non-diegetic elements as possible. We refer to these interfaces as tacit, as the player does not explicitly interact with any visible UI elements.

For the tacit controls prototype, we used a scheme similar to *Jelly Car* for the horizontal movement. Essentially, in this system there are two invisible "buttons" responsible for moving the character in opposite directions, each taking up half the screen. The character will move in a set direction at a set speed regardless of where the player touches within such a "button." The player directly controls the character, so the character moves only as long as the player touches the screen.

While the jumping in *Spider: The Secret of Bryce Manor* is both rewarding and intuitive, we deemed such a system unnecessary and overly complicated for *Continuity 2*. In *Spider: The Secret of Bryce Manor*, the player requires precise, directed control over the spider's jumping. In this game, the player swipes across the spider character and the spider jumps along the vector of the swipe. In *Continuity 2*, however, we did not want to encourage such dexterity-based jumping segments. As such, we focused on repeatability over precision and provided only two pre-determined jumping vectors, as opposed to many.

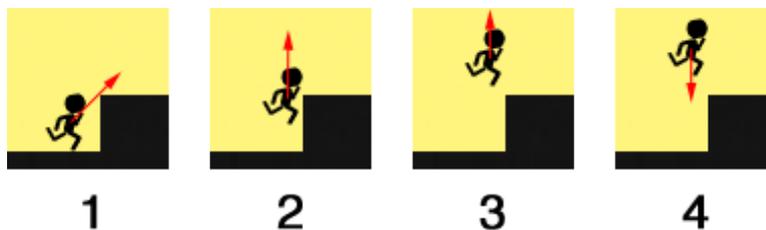
For this two-mode prototype, the player could swipe upwards on either side of the character in order to command the character to jump. Unlike *Spider: The Secret of Bryce Manor*, the character did not jump along the vector of the swipe. Instead, the character jumped at a set velocity towards the side of the screen on which the swipe occurred. The velocity of the

jumping was fixed so that the player could easily predict what would occur and repeat the action.

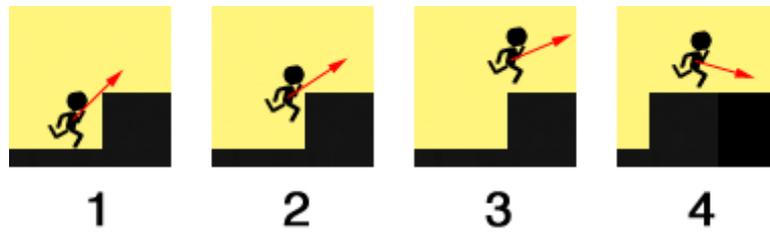


**Figure 5.14:** A visualization of the invisible “buttons” of the tacit interface. If the player touches anywhere in the red region, denoted by “L”, the character will move left. If the player touches anywhere in the blue region, denoted by “R”, the character will move to the right. Swiping upwards in these regions will cause the character to jump in the respective direction.

A major problem we encountered when developing this jumping mechanic was that it was difficult to jump over objects. The character would collide with the side of the object and the character would lose its horizontal velocity. Players had a tendency to lift their finger from the screen when swiping, so the character ceased trying to move to the side. To rectify this, the game constantly reapplies the horizontal velocity until the player changes the movement direction or until the character lands on the ground (the game’s physics system prevents the character from moving through other objects). This allows the character to pass over the object once the character is no longer contacting the obstacle’s side.

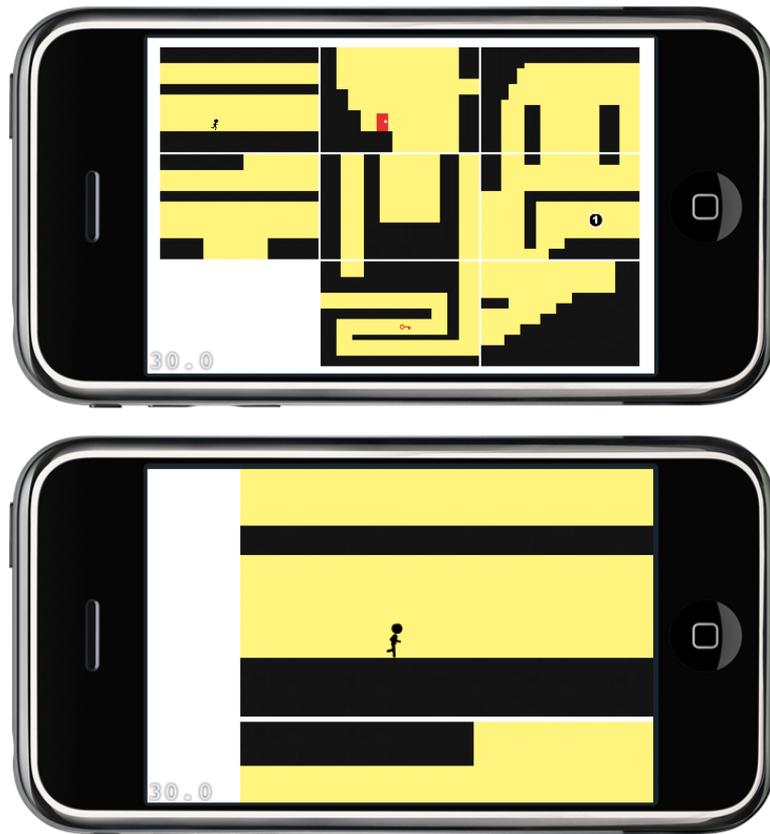


**Figure 5.15:** A problem with jumping over objects using tacit controls. **1.** The character begins a jump with an initial velocity. **2.** The character collides with the side of the object. The character no longer has any horizontal velocity. **3.** The character clears the side of the object, but has no horizontal velocity. **4.** The character begins to fall and will not pass above the object.



**Figure 5.16:** The solution to jumping over objects. **1.** The character begins a jump with an initial velocity. **2.** The character collides with an object, but maintains his horizontal velocity. The physics system prevents the objects from overlapping. **3.** The character clears the side of the object and begins to move over the object. **4.** The character continues to move over the object while descending.

To further minimize the presence of visible interface elements, the magnifying glass button was removed. In either mode, the player simply double-taps anywhere on the screen to switch between the micro-mode and macro-mode.



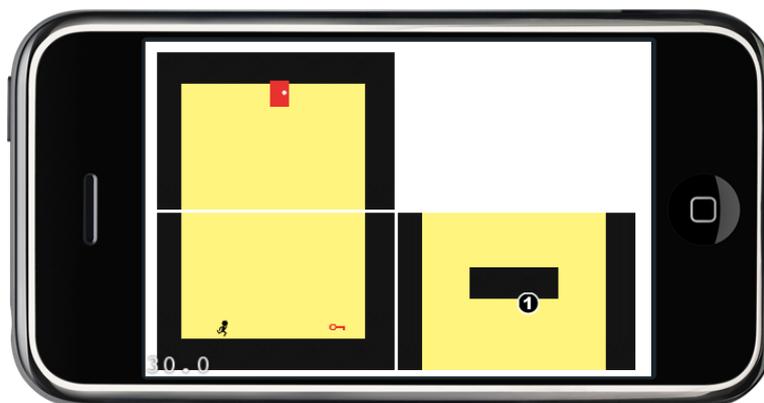
**Figure 5.17:** The prototype of tacit controls. The player can switch between modes by double tapping anywhere on the screen. **Above:** The macro-mode. **Below:** The micro-mode.

We feared that, if the player's abilities to change the gravity of the game were too potent, the player would cease to play the game as a platformer. For this reason, the gravity in this prototype could only be modified to one of four directions, each of equal strength.

Since we wanted to surprise the player with his or her ability to modify the game's gravity, we decided that the game should prevent accidental rotations from modifying the gravity. The operating system of iPhone is capable of notifying applications when the orientation of the device has changed to one of the four orthogonal orientations. Many applications use these notifications to rotate themselves and realign their interfaces. This notification is not immediate, but takes longer than a second. For this prototype we decided to use this operating system notification for several reasons:

1. The timing of the rotation of the game would be consistent with other iOS applications.
2. The delay in notification made the gravity rotation effect less prone to accidental discovery or use due to players' incidental tilting.
3. The delay prevented the player from manipulating gravity often, in quick succession, as this would be too powerful.
4. These notifications made prototyping rotation easier than monitoring the accelerometer data directly.

When the game was in the macro-mode, no visual clue was presented for the player to know that the gravity had changed. Our hope was that this would also discourage players from accidentally discovering rotation. When the game was in the micro-mode, once the gravity was changed, the character would rotate such that its feet pointed "down," towards the direction the gravity pulled. Since the player was expected to not discover the ability to rotate gravity through normal play, levels were created that we hoped would facilitate this discovery.



**Figure 5.18:** A level that was created to motivate the player to discover rotation controls. Without rotating the gravity, the character cannot reach the door.

### 5.6.5 Playtesting

Six playtesters played both the touch-to-stop prototype and the rotation controls prototype. Three of the playtesters had played *Continuity*. We used Fly-on-the-Wall observation combined with interviews after they finished playing to gather information.

Players drastically preferred the two-mode prototype. Players expressed frustration over their inability to directly control the character in the touch-to-stop prototype. In general, players were constantly stressed by the semi-autonomous movement in the touch-to-stop prototype and were unable to multi-task by rearranging tiles while the character moved. Although the touch-to-stop's controls were universally disliked, some players expressed that they liked the one-mode gameplay.

In both prototypes players understood that the coins were optional, bonus items without explanation. Some players ignored them or only collected those that were along the path the character took. Other players sought out the coins. These results solidified that such coins were a good way to reward players and allow for player-customizable difficulty.

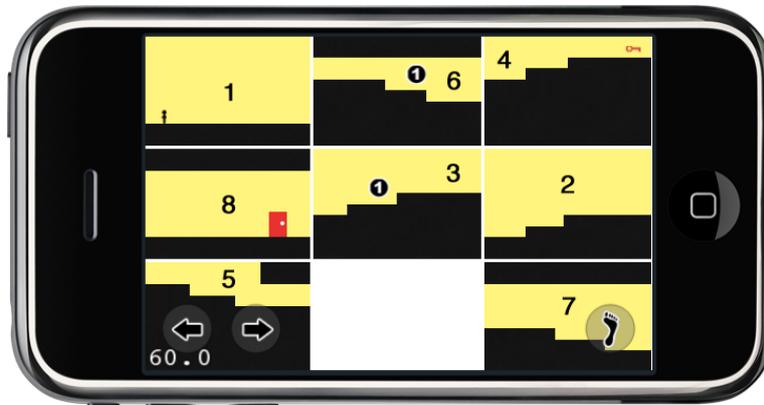
#### **Touch-to-stop**

Players unanimously disliked the touch-to-stop prototype. The primary problem was that players felt anxious about their lack of direct control of the character. Players were constantly worried that the character would end up dying or otherwise in the wrong place.

In general, this stress about the character movement left players uninterested in attempting to rearrange the tiles while the character was moving. Instead, players played in much the same manner as they did in the original *Continuity*, locating a path, moving the character and then reassessing.

Over the course of playing the five levels available, no player rearranged tiles while the character was moving for a smooth, constantly engaged experience. Instead, players were left idle, observing the character moving without participating. This left players even less engaged than in the original *Continuity* with two-mode controls.

All players held the device as a gamepad when playing this prototype. For the most part, players immediately understood the interface. However, players encountered two main problems due to the interface. First, players had troubles selecting and moving the tiles in the bottom corners. Second, players mistimed jumps often, pressing the jump button either too soon or too late.



**Figure 5.19:** A level in the touch-to-stop prototype. The player can solve this level by moving the character to the right through the tiles in the numbered order. The character cannot die in this level, as there are no pits. Even without any threat, players still alternated moving the character and tiles, instead of doing both simultaneously.

### Tacit Controls and Rotation

We were surprised by how well players enjoyed the rotation controls of the game. Our expectation was that players would feel that rotating the device was awkward, gimmicky or both. While players looked awkward in their motions, few stated any problems with the rotation mechanics and in fact expressed that they enjoyed them.

One of the primary problems with rotation was that players were unaware that they could rotate the device in  $90^\circ$  increments. Some players believed that the gravity rotated only  $180^\circ$ . The level that was meant to facilitate the player learning about rotation required only a  $180^\circ$  rotation. On later, complex levels, these players were stumped as the levels required  $90^\circ$  rotation.

Despite certain players not realizing that they device could be rotated in any of the ordinal directions, the complicated level that required rotation

still proved difficult for most players. While the difficulty curve of the prototype was probably too steep, we were relieved to see that rotation-based solutions were non-obvious.

Players were split in terms of how they held the tacit controls prototype. Players that had less experience playing iPhone games tended to use their index finger to play. More-experienced iPhone gamers tended to use their thumbs. This greater accessibility to a wider arrange of preferences seems like a strength of the interface.

Some players had difficulty understanding how the character was controlled. Some seemed to try to drag the character along the screen. Others tried to drag the world around the character, as if one was scrolling through the Safari web browser. Eventually, however, all players learned to control the game properly and competently. Jumping also proved difficult at first for some players that tried to flick the character. Given time, players seemed to come to terms with the jumping controls.

## 5.7 Iteration 2: One-mode and Tacit Controls

In the previous playtest, players widely disliked the touch-to-stop prototype. The version of the game induced stress in players and they played very slowly, which is exactly what the mode was supposed to alleviate. However, some players did state that they enjoyed the perspective of the one-mode play as it made navigating the levels easier.

For the second iteration, we decided to test a one-mode prototype in which the player directly controlled the character. The user interface was similar to the touch-to-stop interface. The two-mode prototype was left unchanged from the previous iteration.

This iteration occurred rapidly as it consisted primarily of slight modifications to the controls and interface of the touch-to-stop prototype in order to create a one-mode prototype with direct controls.

Playtesting was undertaken with a new group of players that were offered both the new one-mode and old two-mode prototypes.

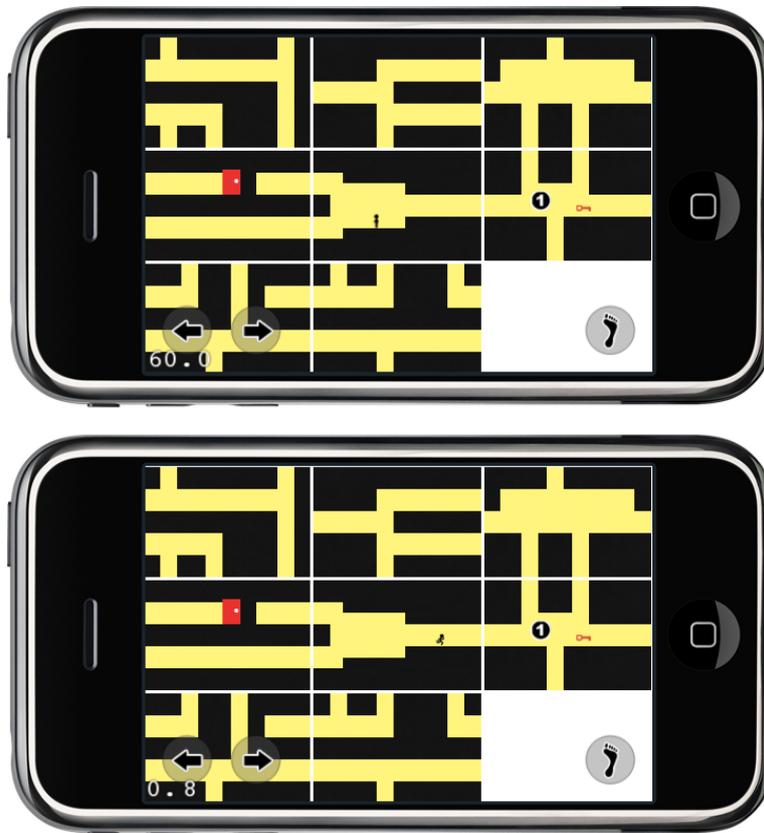
### 5.7.1 Designing One-mode With Direct Controls

Players did not seem to have a problem with either the configuration or visuals of the buttons in the touch-to-stop UI. Therefore, for one-mode controls, we simply altered the icons and functions of the buttons. In this one-mode prototype, the character would only move as long as the player pressed a virtual movement button. Since there was no need for a stop button, as in touch-to-stop, the movement buttons always retained the same icons and functions.

Since the levels designed for touch-to-stop seemed to work well for the one-mode concept, we did not spend any time creating or redesigning levels for the prototype. Other than the interface and character movement, all other aspects of the prototype were taken from the touch-to-stop prototype. We did not include the rotation mechanics in this prototype since, as the game world is never paused, the player had no way to safely rotate the device. If the player were to rotate the device, the character would likely oftentimes fall into a pit and die, resulting in lost progress and frustration for the player. If playtests indicated that players enjoyed the one-mode controls, we could later introduce a method for rotation mechanics in the one-mode system.

### 5.7.2 Playtesting

The goal of this playtest was to determine whether players enjoyed the one-mode controls more than the tacit controls and the rotation mechanics. New playtesters were presented with the new one-mode prototype as well as



**Figure 5.20:** Screenshots from the one-mode prototype. **Above:** The character is stopped. **Below:** The character is running to the right.

the tacit controls prototype from the previous playtest. Players were given instruction regarding the goals and controls of the game.

Again, players tended to prefer the tacit interface and the rotation mechanics.

### One-Mode

Players expressed that they liked the direct controls in the one-mode prototype much more than the previous players enjoyed the semi-autonomous character movement of the touch-to-stop prototype. However, players still experienced problems. In general, performance was poor as players frequently failed to press buttons correctly. Players missed what should have been fairly simple jumps repeatedly by pressing the jump button either too early or late. Most players did not state that the controls were difficult to use; however, the lack of tactile feedback likely accounted for the observed lack of competency.

In addition to having difficulties controlling the character, the players

also demonstrated difficulties moving the tiles. The buttons obscure a fair amount of the two tiles in the bottom corners. Players were often unable to reliably and quickly drag a tile if it was in such a corner.

Overall, players expressed fairly positive opinions about the mode despite their poor performance. Many of the testers were not owners of iPhones or other similar smartphones that possessed a multi-touch display. We believed that some of their interest in the virtual buttons was because they were more familiar in that they resembled a gamepad.

### **Tacit Controls and Rotation**

Players again had some difficulties learning the tacit controls. We continued to observe that players tended to try to directly manipulate the character by touching it. However, after a brief period, they proved fairly proficient at controlling the character.

Despite some of the same physically awkward motions that players demonstrated when rotating the device, they again insisted that they enjoyed the extra challenge added by rotation. Additionally, they were able to perform the actions competently. From the follow-up interviews, we discovered that a problem players experienced with the interaction was due to the delay in the changing of the gravity's direction. Players felt that the delay between rotating the device into a new orientation and the rotating of the game world's gravity was too long.

Players expressed that they preferred the gameplay mechanics and cleaner visuals of the tacit interface.

## 5.8 Iteration 3: Accelerometer-Based Rotation

Playtesters had enjoyed both the one-mode and tacit, two-mode prototypes. However, based on observation, we found that players performed more poorly with the one-mode system.

While players enjoyed the one-mode mechanics, the buttons overlaying the tiles caused problems. Players frequently struggled to drag tiles that were partially obscured by the buttons. The game world could have been shrunk and moved up, creating an interface bar on the bottom of the screen dedicated to the virtual buttons. We had already established that the character should not be any smaller on the device’s screen in order to remain clearly visible. Reducing the size of the game world would have been equivalent to meant making the character even larger relative to the game world—the character was already twice as tall, relative to the game world, as in the original *Continuity*.

Given that we were already concerned that the tiles were too small relative to the character’s size, we decided to abandon the one-mode mechanics in favor of the two-mode mechanics featured in the original game. We believed that by maintaining the tiles at a relatively large size, we would be able to create more interesting levels. Further, players seemed to especially enjoy the rotation mechanics that we felt would not be suited to one-mode play.

Some players did have problems with the tacit interface, so we were unwilling to commit to it. For the next prototype we would try to refine the tacit interface and focus on teaching it to players through the game. If players were unable to gain proficiency with the interface, we would then reconsider displaying a virtual gamepad in the micro-mode.

This prototype also included the levers, wires and gates that we had previously designed. Based on play feedback, we also increased the responsiveness of the rotation of the game’s gravity to the player rotating the iPhone.

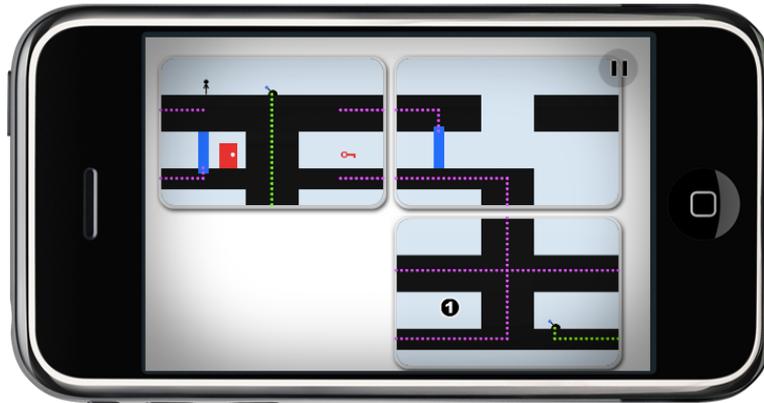
The visual of the game were also improved, making the game appear visually more similar to the original *Continuity*.

### 5.8.1 Levers, Wires and Gates

For this prototype, we implemented the mechanics of levers, wires and gates that were discussed previously (Section 5.6.2). Adding these elements to the game required creating placeholder art assets that were comprehensible. We also designed levels around teaching these concepts to players.

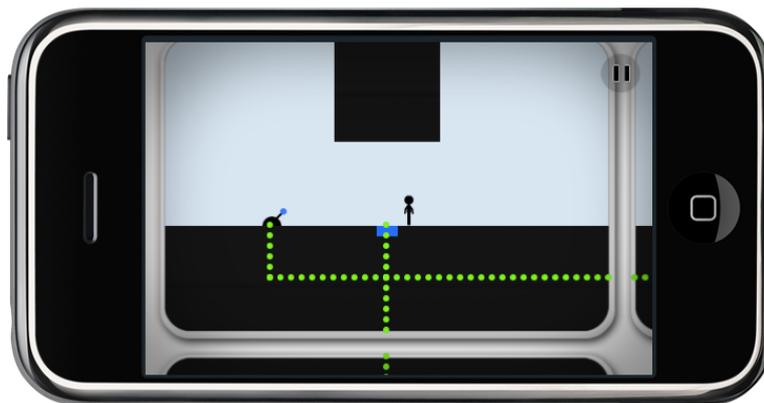
The primary problem with the wires, from a usability standpoint, was that players needed to be able to clearly determine a given wire segment was energized by being connected to a lever. Originally, we conceived of three states for each wire: not energized; energized, but not connected to a

gate; and energized and connected to a gate. In the end, we decided on just two states: energized and not energized. This solution proved to be easier to implement and still allowed players to identify whether a gate or wire segment was connected to a lever.



**Figure 5.21:** Energized wire segments are colored green. Non-energized wire segments are colored purple.

The other major concern of ours was that players were able to distinguish gates from the physical structure of the game world. For this prototype, we used blue rectangles to represent gates. Additionally, players needed to be able to tell where open gates were located. Therefore, rather than completely receding into the level geometry, gates receded into a stub, which is always visible.



**Figure 5.22:** Gates recede into stubs so that players know where open gates are. The stubs are visible from both micro- and macro-modes.

We tried to find interesting, reusable level design patterns while developing the original Continuity. When first adding levers, wires and gates, we too tried to identify potential patterns. Each of these patterns served

as a puzzle that would provide an interesting learning experience for the player. These patterns could hopefully be reused and recombined across multiple levels. By creating and playing levels, we found several of such patterns. Examples of these patterns include “the unwanted connection” and the “ $n$ -puzzle” (Figures 5.23 and 5.24, respectively).

### 5.8.2 Rotation Controls

The major concern players expressed with the rotation controls was the delay between when they would rotate the iPhone and when the game world’s gravity would change. In order to remedy this complaint, we decided to directly use the accelerometer to determine the orientation of the device instead of waiting for the notifications from the operating system. In this system, there is no delay between the rotation of the device and modifying the game world’s gravity.

### 5.8.3 Teaching Character Movement

The original Continuity simply displayed the game’s five input keys to the player during the first level. The game left for the player to determine what effects the inputs caused. However, since the inputs to the game consisted of only five keyboard keys (the four arrows and the space bar), there were few issues for the player of discovery or repeatability.

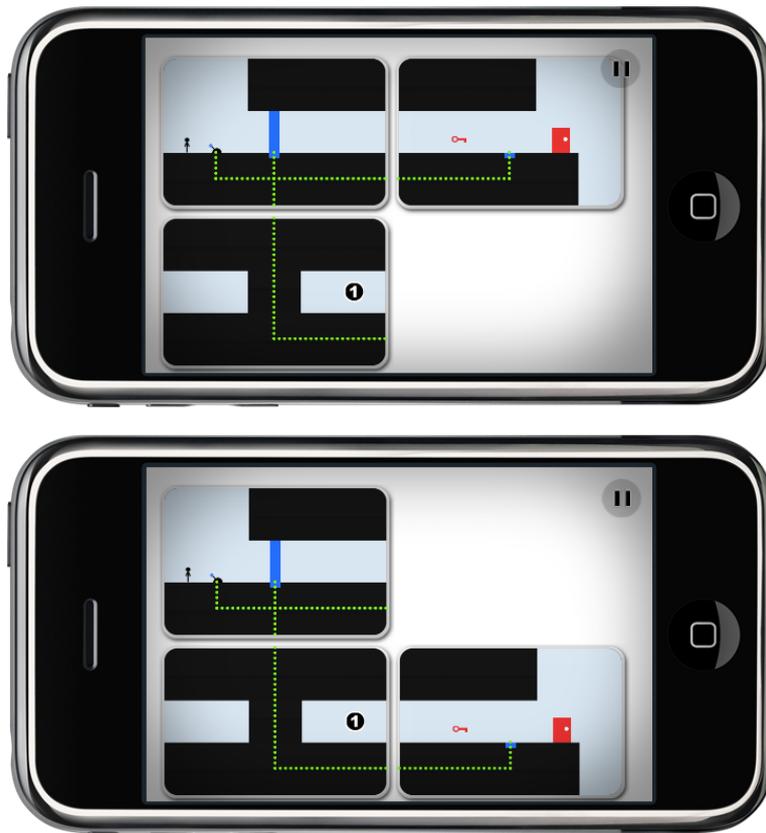
There are many more possible inputs on a touchscreen device than on a keyboard. There is little precedent for a common control scheme for 2D games such as Continuity 2 and various iOS applications and games accept a multitude of different types of touches and gestures. We understood that requiring the player to simply discover the controls might prove difficult.

Our first attempt was similar in nature to that used in the original game: the tile backgrounds were used to display information. While we could have used a clear, written statement such as, “Double tap the screen to zoom in or out,” we wished for the game world to remain language neutral as the original had. Therefore, we designed a system in which fingerprints would be used to indicate how and where the player should touch the screen. Our hope was that the players would touch in the appropriate places, view the response, and develop a mental model of the cause and effect relationship. See Figures 5.25 and 5.26 for examples of these teaching levels.

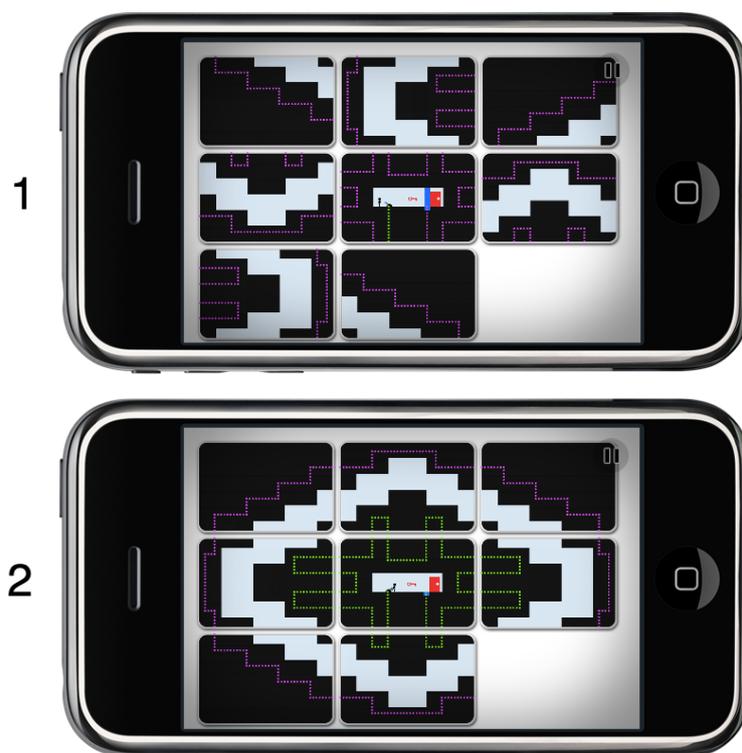
### 5.8.4 Playtest

Another group of playtesters were given the prototype. They were provided no initial instruction. In general, players enjoyed the prototype.

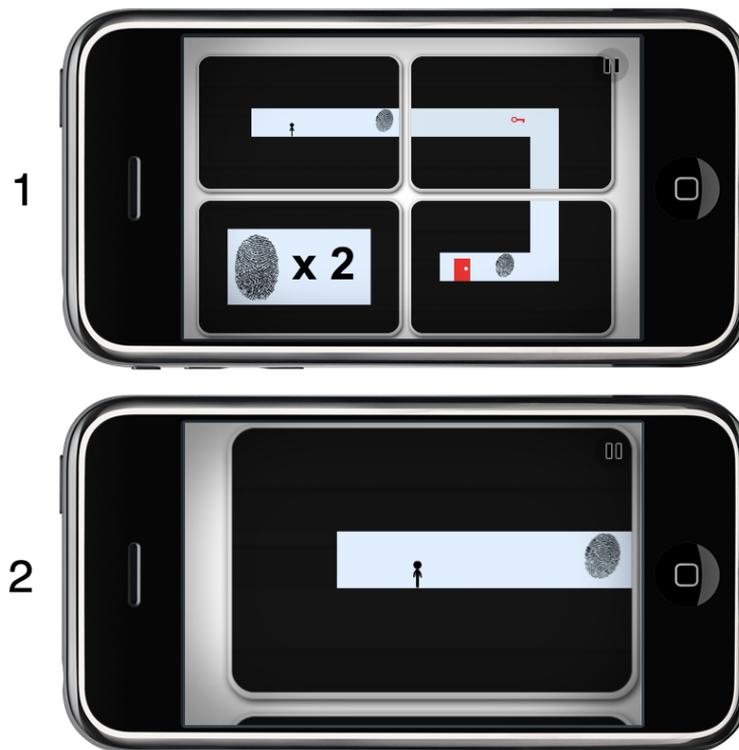
The instructional graphics in the first levels did not prove helpful for players. Players that had played the original Continuity and those who had not played the original quickly learned and understood the lever mechanics.



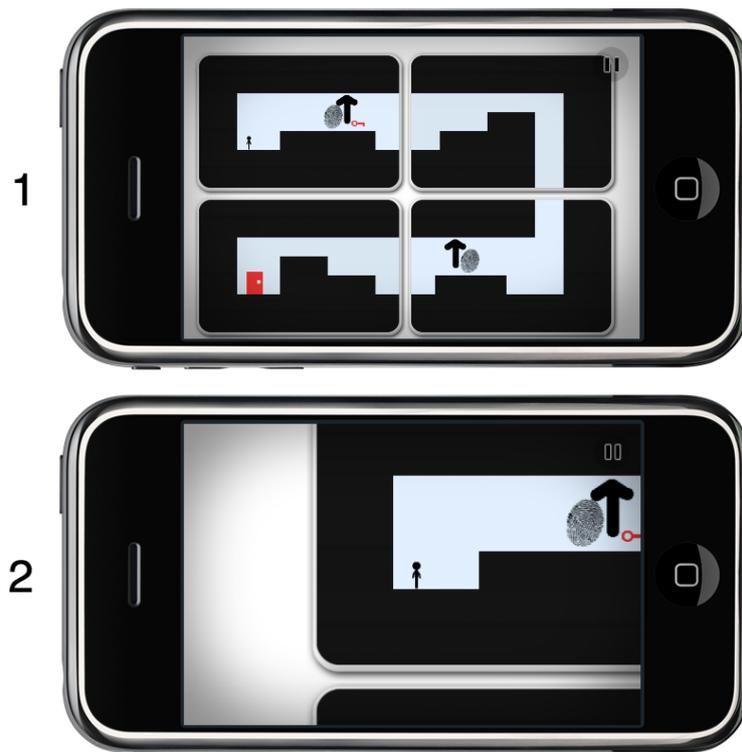
**Figure 5.23:** The “unwanted connection” pattern. In this pattern, opening one gate will close another. The player must arrange the tiles to break the connection to the currently open gate



**Figure 5.24:** The “ $n$ -puzzle” pattern. To connect a lever to a gate, each tile must be placed in a certain position.



**Figure 5.25:** The first level of the third prototype attempts to teach switching modes and making the character run. **1.** The initial view of the player. The player cannot move any tile. The player is encouraged to double tap the display. **2.** Once the player double taps the screen, the perspective changes. The player is encouraged to press on the right side of the screen, which will move the character to the right.



**Figure 5.26:** The second level attempts to teach jumping. Once the player double taps the display to zoom in to the micro-mode, the player is encouraged to swipe a finger up on the right side of the screen, which will make the character jump to the right. After the character falls into the bottom-left tile, the player will be encouraged to swipe upwards on the left side of the screen to make the character jump to the left.

Players also demonstrated a greater competency with the rotation controls, however some players relied on them too much.

### **Levers, Gates and Wires**

Players experienced few problems in learning how to operate the wire systems. In fact, some of the levels seemed too trivially easy for most of the players. The prototype's difficulty curve seemed too flat when introducing these mechanics over a series of levels. Later levels that featured levers, gates and wires approached the proper difficulty level. Specifically, a level that required the use of both levers and rotation mechanics seemed like an appropriately difficult level, without being too challenging.

### **Teaching Character Movement**

The instructive images in the first two levels worked less well than we had hoped. One of the primary problems seemed to be that in the first level, players did not immediately focus on the tile encouraging them to double tap the screen. Instead, they focused on the smaller fingerprints that were supposed to instruct the player about character movement. Placing the double tap tile in the upper-left corner likely would have drawn the players' attentions more immediately. Once players discovered how to zoom in, they learned how to move fairly quickly.

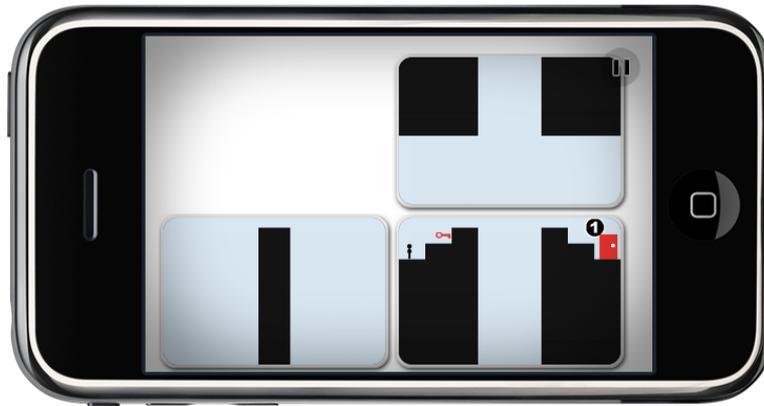
Players experienced further problems discovering how to jump. One player, who did not immediately understand how to make the character jump, rotated the device. When later questioned about this, the player explained that he was aware that the iPhone's accelerometer was used in many other iPhone games and he thought it might do something in the prototype.

A problem that still inhibited players' learning of the character movement is that they still attempted to directly touch and manipulate the character. Although unpredictable, these arbitrary touches and swipes would have some effect and the character would move or jump. Some players started to develop the wrong mental model, believing they needed to swipe horizontally to make the character move.

### **Rotation**

With the quick response of the new rotation mechanics, players experienced very few difficulties in executing rotation. Some players felt so comfortable with the rotation controls that they used them for almost all character movement. Rather than having the character run along a hallway, the players would rotate the iPhone such that the character would fall down the hallway and then rotate the device back to bring the character back onto the ground.

We viewed the potency of the rotation controls, as a concern as some players seemed to think the solution to all puzzles involved rotation. Regardless of whether rotation-based solutions were the easiest to enact, players seemed to only consider rotation as a way to overcome puzzles.



**Figure 5.27:** During the playtest of the third prototype, some players used a variety of dexterity based paths through this level using rotation controls.

## 5.9 Iteration 4: Take Home

All of the previous playtests had taken place in single sessions, under our supervision. These single-session playtests worked to determine how players interacted with some aspects of the game, they did not work well for others. Specifically, we wanted to test if players were encouraged to partake in speed runs and whether they found these challenges rewarding. As we were growing more confident with the game's main gameplay and interactions, we decided to focus on these out-of-game elements.

This prototype added a new level selection system to the game in which the levels were divided into worlds. Menus were added to allow players to view details about their achievements for each level. This system allowed players to see a better overview of their progress through the game.

This prototype also contained spikes as a game world object that served to limit the potency of the rotation controls. Using the spikes along with previously created gameplay elements, we created, removed or modified many levels for this prototype.

The playtest for this prototype was installed on players' phones. Rather than playing the game while we observed, they played the game over a period of two weeks as much or as little as they desired.

### 5.9.1 Achievement Times

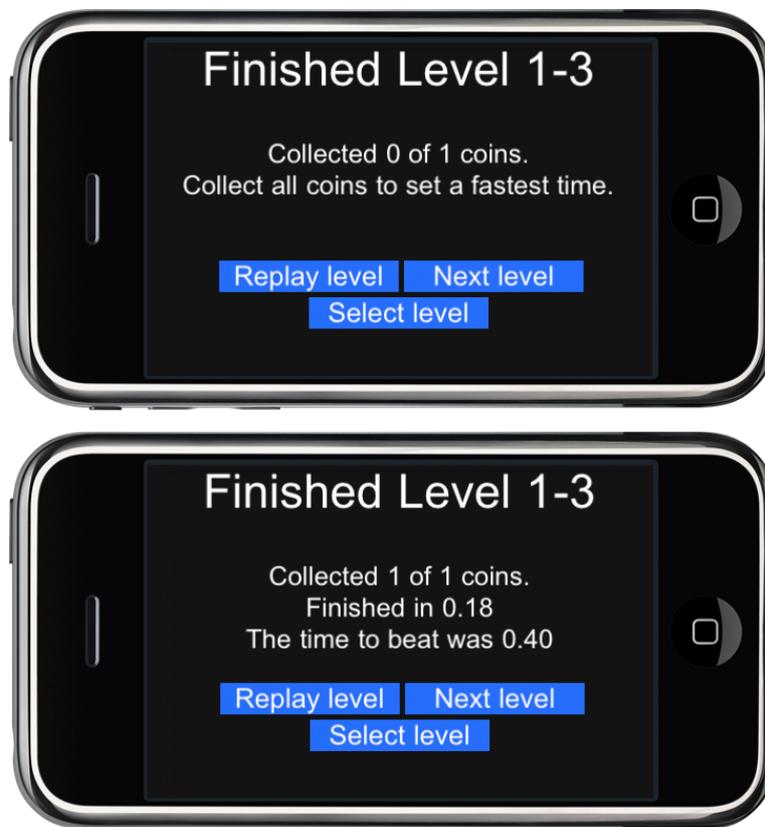
One goal we established was for Continuity 2 to be more rewarding. Speed runs were one way to present the players with more, optional rewards. One pattern to provide players with rewards is the achievement system. In such a system, a player that accomplishes a specific, preset goal earns an achievement. In Continuity 2, we established that each level would have an achievement time. Players would try to collect all of the coins in a level and complete it faster than a given time. For each level, the game would track the best such time so players could strive to improve upon their previous efforts.

We felt that such achievement times facilitated casual play, as players would in no way be required to participate in the system. As with the coins, players could safely ignore the achievement times and still progress through the game without missing any content. Only players that desired the added challenge needed concern themselves with the achievement system. Unlike an extra difficulty mode, in which the player plays every level of the game in some more difficult manner, players were not required to make a long-term commitment. Players could choose to participate in the achievement time system on a piecemeal, level-by-level basis.

The prototype takes two steps to encourage players to attempt achievement times. After every level is completed, the player is presented a debriefing screen informing them about their completion of the level. First, if the player completes a level without collecting all the coins, the player

is notified that he or she must collect all the coins to set an achievement time. Second, if player did complete the level with all the coins collected, the player's completion time is presented in comparison to the preset achievement time. In either case, the player is able to choose to replay this level from the debriefing screen. However, the player can ignore this information and can continue onto the next level or elect to choose an arbitrary unlocked level.

In order for the player to know how long they have spent in a given level, the game displays a timer during levels in both the micro-mode and macro-mode.



**Figure 5.28:** The level debriefing screen after completing level 1-3. Players can choose to replay the level, play the next level or choose which level to play next. **Above:** The player completes a level without all the coins. **Below:** The player beats the level, with all of the coins, faster than the achievement time.

### 5.9.2 Level Selection

In Continuity, players were unable to select the level they wished to play. Players encountered levels in a preset order. The game allowed players to skip a level with the ability to encounter it later; however, players had no

way to know they would be able to play it later. After completing a level, the player was notified only of the number of the level that was completed and how many levels remained.

For Continuity 2, we wanted the player to feel more empowered and rewarded in selecting and playing levels. We believed that by allowing players to select any level they wished, they would feel freer to only play the levels that interested them. This would allow players to play only a minimal number of levels and ignore those that posed too much of a challenge or that they found uninteresting. Players that especially enjoyed the game or otherwise who wanted to experience the game in its entirety were able to play all the levels. Further, we believed viewing an overview of all the levels, with some visual clue as to which had been completed and which remained uncompleted would feel more rewarding to players as they progressed and “checked off” levels.

The system we developed to grant players these freedoms and encourage a sense of reward consisted of dividing levels into worlds. When a player starts playing the game, all worlds except for the first are locked. Levels in these locked worlds are unavailable to the player. The player is free to select any level from an unlocked world that he or she desires. When a certain number of levels have been completed in a given world, the next world is unlocked. Hopefully the player feels a sense of reward when a new batch of levels is made available to him or her.

The levels are named according to the world and the order they occur within that world. For example, the fourth-listed level of world two is named “2-4.” Although players can play the unlocked levels in any order, the levels within a world tend to become more difficult as their number increases. Our hope was that if players play the levels strictly in order they will have an experience with an appropriate, generally increasing, difficulty curve. New concepts are introduced in the first levels of a world. The remainder of the world builds upon these concepts.

For this playtest, we did not focus on the graphical design of these menus. We left placeholder art in order to discourage playtesters from evaluating the graphical design. We rapidly developed a prototype with an adequate information architecture that presented the requisite information. For this prototype, the level selection screen takes the form of a scrollable display. Each row in the display presents information about a specific level. If a level has been completed, a red key icon is displayed. If a level has been completed with all of its coins being collected, a yellow coin icon is also shown. The menu always displays the achievement time for each level. The menu also displays the player’s best time completing a level while collecting all of its coins.



**Figure 5.29:** Level selection screens from prototype 4. **Above:** All the levels of World 1 have been completed and all the coins in each have been collected. The player can play any level in World 1 or transition to World 2. **Below:** The player has only completed the first two levels of World 2. The player can play any level in world 2 or transition to World 1. World 3 is still locked and is unavailable.

We did not want to force players to navigate these menus. Players that wished only to play the levels, in order, should be encouraged to continue playing. We felt the game should afford such players the ability to transition quickly, with little effort, from one level to the next. Therefore, after each level is completed, the player can elect to either return to the level selection screen or continue to the next unlocked and uncompleted level. To entice players to seek extra challenges, such as collecting all the coins or obtaining an achievement time, the player is also able to immediately replay the previously completed level.

In order to allow for interruptability, the prototype also saves the current level's state when the player exits the application or the player exits the level. The player is never required to take any action to explicitly save his or her progress. When the application is resumed later, the player may continue exactly where he or she left off. The timer for the level resumes exactly where it left off when the level was exited. This allows the player to better

deal with unexpected occurrences such as receiving a call or planned, brief sessions of play. The player may restart a level from the beginning using the pause-menu if he or she desires.

### 5.9.3 Character Controls

Clearly communicating the character controls to the player was found to be a primary problem in the previous playtest. For this prototype, we tried to reduce some of the ambiguity in communicating the controls.

Players had previously tried to interact directly with the character by touching it. This resulted in odd-seeming effects, as the character would move to the left or right, depending on whether the player touched slightly to the left of the character or slightly to the right, respectively. In order to clarify that players were supposed to touch to the sides of the character, rather than the character itself, we added a “dead space” to the micro-mode controls.

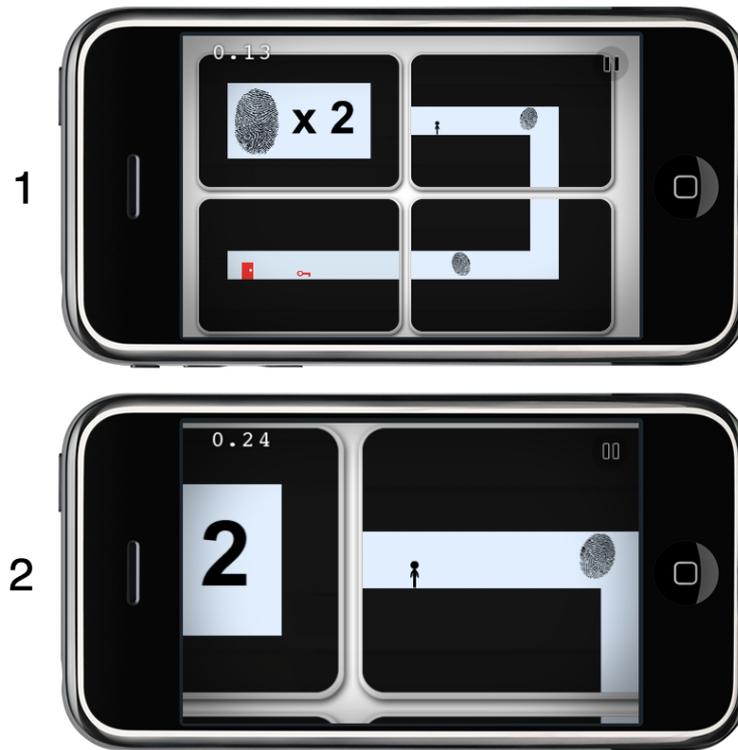
This dead space spanned the 100 pixels in the center of the screen. Touches that occurred within this space were ignored for the purposes of the character’s horizontal movement. Players could swipe upwards in this space to cause the character to jump straight up. Our hope was that since touching in the middle of the display had no effect, players would be encouraged to try touching towards the sides of the display. By forcing touches outwards, we hoped players would more quickly understand that they were meant to touch towards the side of the display to which they wished the character to run.



**Figure 5.30:** A visualization of the tacit controls with the dead space in the center. Touching in the left area, denoted by “L,” will cause the character to run to the left. Touching in the right area, denoted by “R,” will cause the character to run to the right. Touching in the center has no effect.

We also revised the instructional graphics we had created for the first levels of the game in the previous prototype. Previously, players tried many

other actions before double tapping the screen. We believed that this was because the symbol representing double tapping, was placed in the bottom-left corner. As the playtesters were accustomed to reading, left-to-right, top-to-bottom, it seemed that placing the symbol for double tapping in the upper-left tile might make players focus on it more quickly.

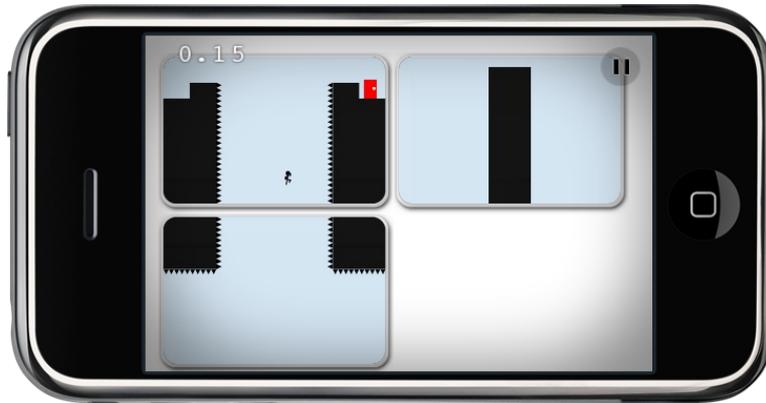


**Figure 5.31:** The revised first level with instructional graphics. The tile encouraging the player to double tap has been moved to the upper-left corner of the screen. **1.** The view the player is presented with when starting the level. **2.** The view the player is shown after double tapping the screen to switch modes.

#### 5.9.4 Rotation Controls and Spikes

In the previous playtest, some players used rotation nearly every time they needed to move the character from one place to another. We felt that the game needed to discourage players from only using rotation, so that players used other resources.

We added spikes to the game as a way to discourage players from having the character traverse certain surfaces. The spikes were represented by triangles and could be placed along any surface. If the character collided with a spike, it would die. By strategically placing spikes in levels, we found that we could limit the options available to the player.



**Figure 5.32:** Before spikes were added the player could rotate and walk along the sides of the tunnel.

### 5.9.5 Playtest

Players had the game installed on their iPhones and were encouraged to play it as much or as little as they wished. These players were located through Continuity’s Facebook page, and had all played the original game. We interviewed the players in person to collect feedback about their experiences.

Some playtesters expressed that they found the achievement time mechanics to be motivating and rewarding whereas others stated that they ignored the mechanics entirely. No player claimed to find the mechanics a detriment to the game.

Players were generally pleased with the interruptability of the game. Players indicated that they often quit and resumed levels without being lost or confused.

All players seemed to have figured out how to control the game during their time with it and, by the time of the interview, were able to move the character competently. None of the players seemed to have discovered the rotation mechanics before the level that first requires them to be used.

## Chapter 6

# Results

This chapter describes the state of Continuity 2 at the end of this project. The primary focus in this chapter is the game and interaction design; however, a brief outline of the technical components relevant to the implementation is also included.

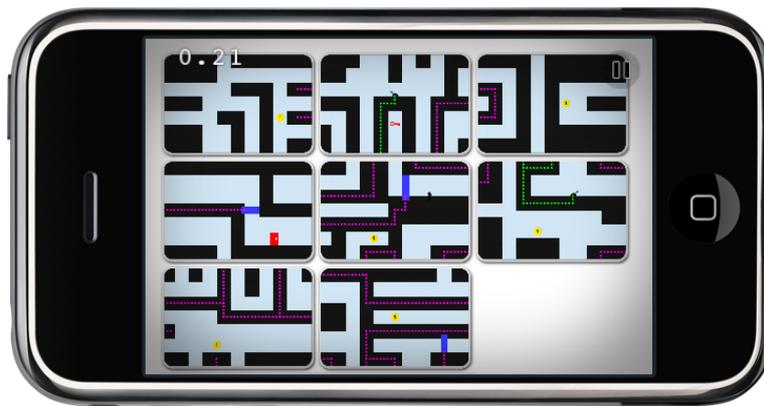
Continuity 2 is a fully-functioning iPhone game with 27 levels. However, the game requires art and audio assets as well as additional levels to be created in order for us to consider it a finished game. Minor software changes and bug fixes are also necessary.

The goal of each level is the same as in Continuity, to navigate the character to a number of red keys and then to a red door. This is complicated by the fact that the world is divided into a number of tiles and the character can only transfer between tiles when the edges of two tiles match perfectly. There are only 3-puzzles and 8-puzzles (levels with 3 and 8 tiles, respectively) in Continuity 2 whereas in Continuity there were 3-puzzles, 5-puzzles, 8-puzzles and one 15-puzzle.

In addition to the actual gameplay, the application includes a menu system that allows the player to browse and choose which level to play at any time. From this level selection screen the player is able to see which levels have been completed and the current fastest completion time for each. The menu system is currently a prototype and features temporary graphics and artwork.

A player is able to quit the game and resume exactly where he or she left off at anytime. Re-opening the application puts the player back in the level he or she was playing in less than five seconds on a second-generation iPod Touch. Overall the application has acceptable performance running at more than 30 frames per second on a second-generation iPod Touch.

The visual representation of Continuity 2 is similar to that of Continuity. The main difference between the two is that the graphics of the tiles have been reworked in Continuity 2 to resemble tiles of an actual  $n$ -puzzle rather than playing cards, as in the original Continuity. The character animations,



**Figure 6.1:** A level from Continuity 2.

music and a sub-set of the sound effects from Continuity have been re-used in Continuity 2. Currently, Continuity 2 uses the same two music tracks from Continuity; one is played while in the macro-mode and the other is played in the macro-mode.

In addition to the iPhone application, a level editor that runs on a PC has been developed to facilitate the creation of levels.

## 6.1 Gameplay Design

Continuity 2 reuses many of the same design solutions found in Continuity in addition to some new mechanics. In both games, players complete a level by moving the character through a 2D world, divided onto sliding tiles. The character must collect all the keys that are placed in the level and then move to the door to complete the level.

In Continuity 2, as well in Continuity, the player is presented with two modes of control. In micro-mode the player controls the character and in macro-mode the player slides the tiles. The actions that the character can perform are limited to moving left or right and jumping. When switching from micro-mode to macro-mode the camera zooms out to show all the tiles and the game world is paused. When transferring from macro to micro-mode the opposite occurs: the camera zooms in on the character and the game world is unpaused.

### 6.1.1 Keys and Coins

In Continuity 2, there are two items the character can collect: keys and coins. The character is required to collect all the keys in order to complete a level. The coins are bonus items and the player is never required to collect any of them. If a level has been completed with all the coins collected, a yellow coin symbol is added next to the level name in the level select screen as an achievement. The character must collect all the keys and coins in order to set an achievement time.

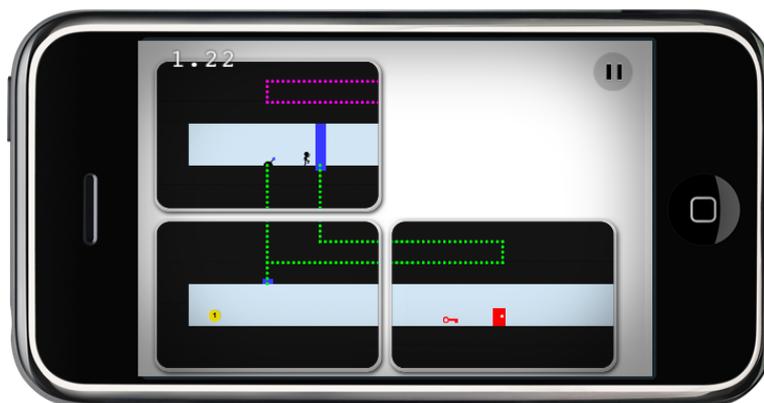
### 6.1.2 Rotation

The player is able to rotate the game world's gravity in Continuity 2. This allows the character to run along surfaces that were previously walls or ceilings. Gravity can only be changed in 90° increments.

The gravity vector can be changed at any time, in either micro- or macro-mode. Changing the gravity has no effect in macro-mode, as the game world is paused.

### 6.1.3 Levers, Wires and Gates

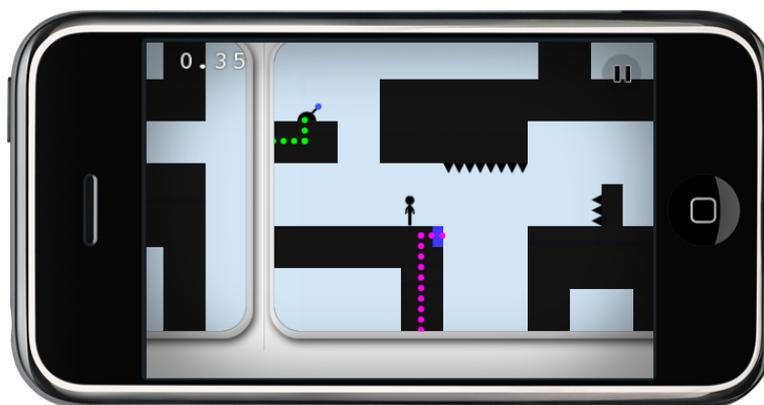
A new concept in Continuity 2 is the addition of levers, wires and gates. A closed gate prevents the character from moving through the passage where the gate is located. Gates are connected to a network of wire segments, which in turn can be connected to a lever. When a lever connects to a wire in such a manner and the character flips the lever, the state of the connected gate is toggled. If the gate is open, it is closed and vice versa. The wire forms connections across matching tiles and flipping a lever connected to multiple gates toggles all connected gates.



**Figure 6.2:** Levers connect to gates via wire segments. These connections span tiles. Green wire segments are connected through other segments to the lever and energized. Purple wire segments are not connected to any lever. When a gate is energized, it can be opened by toggling a lever.

#### 6.1.4 Spikes

In *Continuity*, the only situation where the character would die was when it fell out of a tile with no matching tile beneath. In *Continuity 2* the character can also die if it comes in contact with spikes. These spikes serve to limit the places to which the character can move.

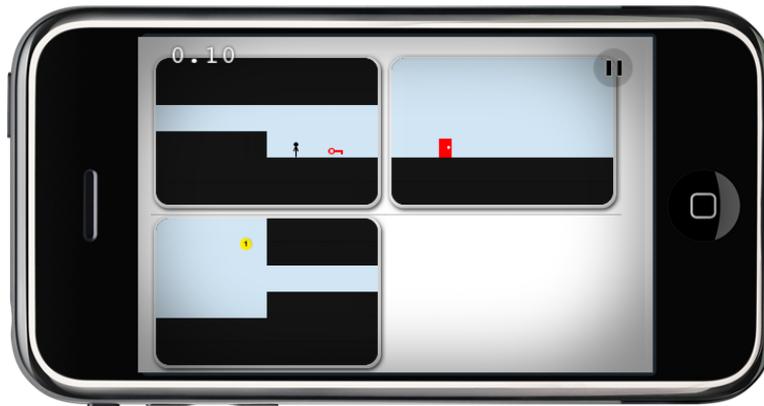


**Figure 6.3:** The character dies when it contacts spikes. These spikes are used to prevent the character from accessing certain areas.

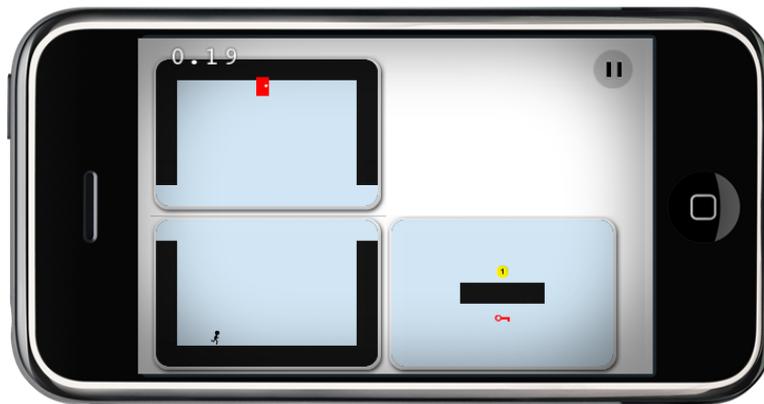
#### 6.1.5 Learning Through Playing

*Continuity* and *Continuity 2* lack tutorials or help text to instruct players how to play; figuring out how the game works is part of the puzzle. The control scheme for the character control is printed on the tile background in the same way as the keyboard controls were presented in *Continuity*.

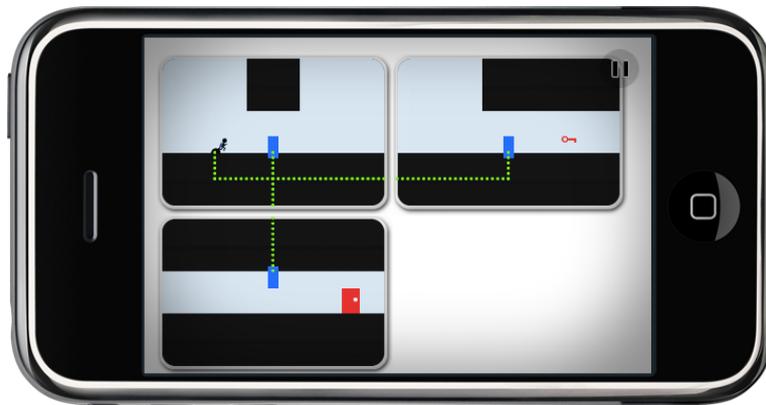
Continuity 2 instructs the player about the character movement controls over two levels, instead of only one as in Continuity. The first level teaches left and right character movement and the second level teaches jumping. In addition to these initial two, other levels have been designed to facilitate learning of the gameplay mechanics. In Continuity 2, there are levels that focus on introducing the player to mechanics such as tile-matching, the rotation controls and the levers, gates and wires system (Figures 6.4, 6.5 and 6.6).



**Figure 6.4:** For the character be able to move from one tile to another, the edges of the tiles have to match exactly. In this example, the character cannot move across the right border of the tile it is currently in.



**Figure 6.5:** The player can rotate the iPhone to change the game world's gravity into one of four directions.



**Figure 6.6:** A single lever can be used to simultaneously open or close multiple gates.

## 6.2 Out-of-Game Features

The 27 levels of Continuity 2 are divided into four worlds. When the player begins the game, only the first world is unlocked. The player can choose to play any level in an unlocked world, but cannot play levels in locked worlds. A world will unlock after the player has completed a given number of levels in the previous world. For example, to unlock World 2, the player needs to complete seven of the eight levels in World 1. It does not matter which seven levels the player completes. The player is not forced to ever visit the level select menu. When a player completes a level, he or she is presented with the options to visit the level select screen, replay the current level or move on to the next uncompleted level.

At any time while playing a level, the player can exit to the level selection screen by using the pause-menu. From the pause-menu the player is also able to restart the current level.

In the level selection menu, the player can browse all unlocked worlds. For each level, the game displays information about whether it has been completed, the current best speed run time and the time-trial achievement time.

The graphic design of all menus in Continuity 2 is temporary.



**Figure 6.7:** In Continuity 2, the levels are grouped into worlds. Each world is presented on a different screen. The player can choose to play any level in an unlocked world.

### 6.2.1 Achievement Time

A player can achieve an achievement time by collecting all the coins in a level and completing the level in less time than a preset value. A different value is specified for each level in the game. The game displays a timer in the upper-left corner at all times notifying the player how much time has elapsed while playing the given level. The only way to tell if the player has

beaten a given level's achievement time is to compare the completion time to the achievement time in the level select menu; no separate visualization is provided.

### **6.2.2 Saving Level Progress**

Unlike Continuity, which only saved the player's progress after levels had been completed or skipped, Continuity 2 saves players progress mid-level. Whenever the player quits a level, the state of the current level is saved. If a player quits the application while playing a level, the next time the player starts the application, the game will bypass all menus and load the state of the level from the last play session.

## 6.3 Controls

Continuity 2 uses the tacit control system. This in-game control scheme is entirely non-graphical with the exception of a pause button that pauses the game and displays the pause-menu.

As in Continuity, the in-game controls of Continuity 2 are divided into two modes. In the micro-mode, the player controls the character. In the macro-mode, the player slides the tiles. Double-tapping anywhere on the screen in either of the two control modes switches control to the other mode. Apart from double-tapping to switch modes, the player interacts with each mode in a different manner.

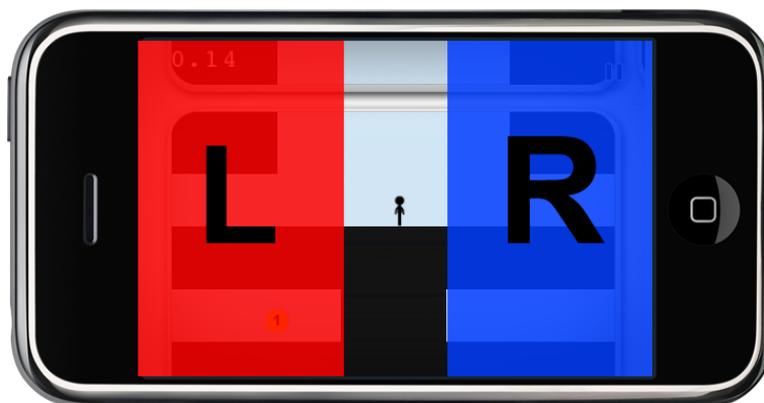
The player can modify the game's gravity by rotating the iPhone from either mode.

### 6.3.1 Micro-Mode

In the micro-mode the player can move the character left or right and jump. The player can move the character left or right by touching the screen in areas to the left and right of the character, respectively. Touching these left or right areas moves the character when it is in the air as well as when it is on the ground. Movement is discontinued when the character is on the ground and the player stops touching the screen. The player can change the movement direction by dragging his or her finger from the left area to the right or vice versa.

These two areas are separated by a “dead space” in the middle of the display. Touching in this area does not cause the character to move. Jumping is triggered by swiping in any upwards direction anywhere on the screen. Depending on whether the swipe occurs in the left area, right area or center area, the character will jump to the left, right or straight up, respectively. These areas are visualized in Figure 6.8.

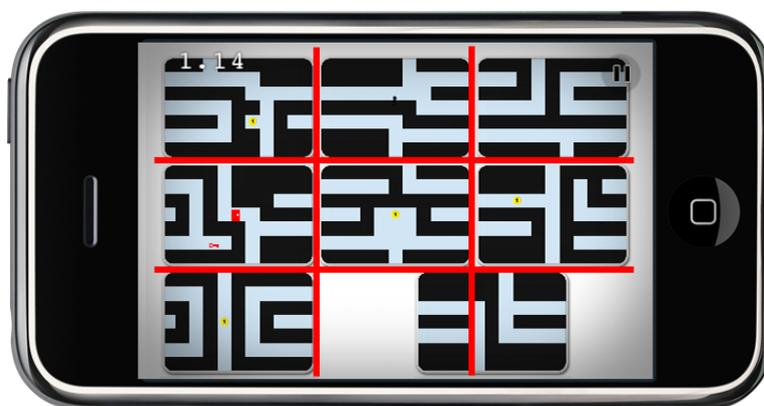
To prevent the character from moving when double-tapping to switch modes, a touch must last longer than 100 milliseconds before the character will begin to move.



**Figure 6.8:** A visualization of the tacit control scheme in the micro-mode. Touching in the left area will cause the character to move left. Touching in the right area causes the character to move right. Touches in the center have no effect.

### 6.3.2 Macro-Mode

A player can move a tile by touching and dragging it orthogonally. When a tile is released it keeps the velocity it had at the time of release. Friction is simulated, so the velocity decreases as the tile moves. When a tile is a eight pixels from a grid location it snaps in place. Only one tile may be out-of-place at once. If a tile is out of place, the player may not switch to the micro-mode.

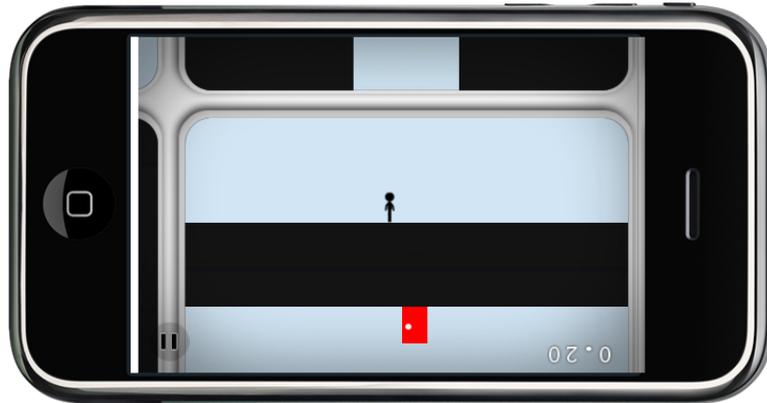


**Figure 6.9:** A visualization of the grid positions of the tiles in the macro-mode. The bottom-right tile is out-of-place while all the others are in place. The bottom right one is the only one that can be moved.

### 6.3.3 Rotation

Rotating the device causes the game world’s gravity vector to rotate in  $90^\circ$  increments. The gravity in the game attempts to mimic “down” in the real

world. The accelerometer is polled for the current orientation three times a second and so changing the gravity directly from up to down is possible if the player rotates the device  $180^\circ$  within a third of a second. If the device is held flat, so the screen is parallel to the ground, due the device has no way of knowing its orientation relative to the player. When held flat, the gravity vector does not change regardless of how it is rotated.



**Figure 6.10:** When the device is held upside down, the gravity of game world changes. This allows the character to stand on what used to be the ceiling.

## 6.4 Software

The application was developed using the iPhone SDK (Software Development Kit) version 3.1.2 and 3.1.3 made available by Apple. The iPhone SDK includes a number of software libraries, the integrated development environment Xcode, an iPhone simulator and a number of other development tools. The simulator is a tool that simulates the execution of an iPhone application and so it can be used to test an application without deploying to an actual device. The code was mainly written in the programming language Objective C with a small part in C++.

Version 0.9.0 beta 2 of an open-source game development framework named Cocos2D was used in addition to the iPhone SDK. Cocos2D is used as a rendering library. Additionally, the physics engine Box2D and the sound engine Cocos Denshion distributed with the Cocos2D framework were used for physics simulation and audio playback, respectively.

The level editor, which was developed to be able to create and edit levels using a graphical interface, was written in Actionscript 3.0 using Flex Builder. The level editor does not work on iPhone; it works on a PC using a mouse and keyboard.

## Chapter 7

# Discussion

This project was undertaken with the goal of fully designing and implementing a sequel to Continuity that felt tailor-made for the iPhone's interface, was familiar in gameplay mechanics and felt distinct from the original. Later, when we realized that we would not be able to complete the game to a satisfactory level of quality within the timeframe, we redefined the goal to concentrate on exploring the design space and producing prototypes. Continuity 2 could be released today in its current form, but we feel the game still requires more work.

The game features a custom-made character control scheme as well as other considerations for the iPhone's interface and context of use. The game likely feels familiar, as all of the game mechanics found in the original Continuity are in Continuity 2. The game uses either the same or similar artwork and sounds, and so is similar from visual and auditory perspectives as well. Our hope is that the new gameplay mechanics we've added and the different method of interaction with the game make it feel distinct from the original.

## 7.1 Planning and Work Process

During the development of Continuity 2, we re-prioritized further exploring the design space over completing the game within the allotted period. As a result of this re-prioritization, Continuity 2 still requires more art and sound assets, more levels and software refinement before we would consider it complete and ready to release.

The development of the original Continuity was mostly linear. Each step built upon the previous step, and we threw very little work away. This happened for two reasons. First, we had an unmovable deadline by which to complete the project. Second, most feedback from playtesting was positive and so our development time was spent making relatively small course-corrections rather than overhauling. This allowed us to finish the first game, to a relatively high quality, much more quickly than if we had made changes that would have required casting aside significant portions of the software or levels.

We planned for Continuity 2 without considering how fortunate the development process of the original was. In hindsight, we failed to schedule enough time to allow ourselves to develop designs and implementations that we would eventually abandon. We also overestimated the amount of work we would be able to reuse from the previous game, as many similar problems required new solutions due to other aspects of the design. Tile movement for example, was common to both games, but required a much more complicated solution in Continuity 2 due to the use of touch-based controls.

Of course, we could have stuck to a predetermined time plan. However, the quality of the game's design would likely have not been as high. If we had been forced to attempt to complete the game by the established deadline, we would have been inclined to try fewer concepts and stick with them regardless of whether they worked. It is likely that we would have entered the transformation and convergence stages of the design process much earlier if we had a fixed deadline. Instead, we were able to produce and maintain multiple designs simultaneously and choose the best one.

If we worked to meet a strict deadline, we might not have been able to invest the time to experiment with the rotation controls. We were never fully convinced that they served as a clear improvement to Continuity 2. Despite our misgivings, the rotation controls seem to be almost universally enjoyed by players. By prioritizing exploring various design solutions, we were able to develop and test the rotation controls along side alternatives. If we were stuck to a hard-and-fast time plan, we would likely have abandoned the rotation mechanics as we would have been unwilling to risk significant resources of implementing software and designing levels for a mechanic which might end up cut from the final game.

Further, without the ability to throw away work we could have been stuck with poor design decisions being cemented into the final product. For

example, at the time we implemented the touch-to-stop prototype, we felt very strongly that it was a great idea for iPhone. If we were stuck to a deadline, we would likely have only prototyped the touch-to-stop version of the game instead of developing two prototypes in parallel. After producing the touch-to-stop version, we would have been disinclined to throw away the work to explore other designs that could also turn out to be of poor quality. We might have been forced to simply try to make subtle changes to the concept in order to make it less bad.

## 7.2 Tailor-made for the iPhone

A lack of a strict schedule allowed us to explore a variety of control schemes for Continuity 2. The tacit control scheme that we developed is specifically developed for the iPhone's multi-touch display. Continuity 2 also accommodates interruption, which affords mobile play in a way the original did not.

The tacit control system is similar to the systems used in *Spider: The Secret of Bryce Manor* and *Jelly Car*. In all the games, the player touches towards where he or she wants the character to move. The primary difference is that Continuity 2 uses a different interaction for jumping. The tacit controls seem to work well in a low-paced platformer. Players have demonstrated an ability to proficiently control the character in Continuity 2 using the tacit interface, given some time to learn it.

The relative strengths of the system are that it is visually minimal and allows for predictability and repeatability once learned. Virtual directional buttons or thumbsticks are relatively small and may prove difficult to press and offer little feedback. The tacit controls, on the other hand, allow the player to press anywhere within a relatively wide area to set the movement direction of the character.

Games like *Spider: The Secret of Bryce Manor* afford a high level of precision in the character's jumping; the character jumps along the vector of the user's swipe. In this model, each user gesture maps one-to-one to an in game character action. In Continuity 2, this breadth of character actions is not required. Continuity 2 maps a large number of different user interactions to a small set of character actions. Continuity 2 ignores the direction of the swipe and only considers the region of the screen in which it occurs. If the swipe occurs in the left, center or right region of the screen, the character will jump to the left, straight up or to the right, respectively.

Of course, the tacit controls also have their relative weaknesses. Tacit controls possess less discoverability than virtual buttons. Since the tacit controls accept a wide range of inputs, players may have some problems developing an accurate mental model of the inputs and outputs. For example, a player may try to swipe the background of the level from the right to the left, in order to move the character to the right. If the player begins this swipe on the right side of the screen, the character will move to the right. Since the player is presented with his or her desired effect, the player's behavior is reinforced. On the other hand, players tend to immediately understand how to control a game via virtual buttons as a button only has two states (pressed or not pressed) and one method of interaction (touching it). However, when these buttons are presented as a virtual gamepad they force the player to hold the device in a specific manner. The tacit control system allows the player to hold the device in a variety of ways and use either his or her thumbs or index finger.

The rotation mechanics of Continuity 2 are another coupling of the iPhone's form to gameplay. While a player could use keyboard keys or buttons on a gamepad to rotate the game world on a PC or console game, these methods of interactions would be abstracted away from their effect. In Continuity 2, the player interacts with the game in a more direct way. To move a tile, the player touches and drags a tile. To rotate the game world, the player actually rotates the device.

In addition to considering the interface, we considered the context of use of an iPhone when designing Continuity 2. The game accommodates the propensity to interruption of a mobile phone as a gaming device. Since the game saves exactly where the player left off in a level when the application quits, the player can play a single level over multiple play sessions. This is made easier by the fact that when restarting the game, the game bypasses all menus and reloads the saved state. Playtesters have indicated that they felt indifferent to positive about this feature. They have not indicated that they have trouble picking up where they left off.

### 7.3 Distinct but Familiar

We found the *stating objectives* method to be effective at clarifying what aspects of the original Continuity we wished to maintain and which we wished to alter. By establishing these goals in terms of broader concepts rather than specific gameplay mechanics, we gained a better understanding of how we wanted the player to feel while playing Continuity 2.

In terms of gameplay mechanics, Continuity 2 ended up quite similar to the original Continuity. The player tries to navigate a character through a level in the form of an  $n$ -puzzle using two separate modes of play. In one mode, the player controls the character. In the other mode, the player controls the tiles. Likely, Continuity 2 will appeal to players in the same way that Continuity did. However, we are uncertain as to what extent those that played Continuity will find Continuity 2 to be a distinct experience.

All the in-game gameplay mechanics from the original Continuity are present in Continuity 2. To these, Continuity 2 adds several new mechanics. Players have expressed that the coins and levers, wires and gates feel consistent with the design of the original Continuity. We feel especially confident that the levers, wires and gates strike a balance between feeling distinct from Continuity while still making allowing the general experience to be familiar. They also happen to help solve a deficiency in the original game's design by allowing the character to have a greater influence over the level by making levels less static. Additionally, they are a game design element that can be reused in a variety of different ways.

Of the gameplay mechanics that originate in Continuity 2, the rotation mechanics seem like the most significant. Players have expressed almost exclusively positive feelings about the rotation mechanics. However, if players perceive the rotation controls as the default way to move the character, we are still concerned that the players will make solving puzzles exclusively about changing gravity via dexterity while overlooking other solutions. The rotation mechanics may make Continuity 2 feel too distinct by completely changing players' perceptions of the nature of the puzzle solving from being based on tile sliding to being based on rotating the iPhone. A strong gameplay benefit of the rotation is that it alleviates the problem from the first Continuity of character movement being too punitive. The rotation mechanics allow the character to generally alter gravity to undo any movement that he or she perceives as being a mistake.

Another way the character movement has been made less punishing is via the coins. Whereas in the original Continuity there were very few checkpoints, Continuity 2 contains many. This means that if the character dies in a level, the player loses very little progress.

The out-of-game aspects of Continuity 2 are almost entirely different than in Continuity. These out-of-game changes are mostly directed at making the game feel more rewarding. We added a level selection menu that

shows which levels have been completed so the players can more easily view their progress. Further, Continuity 2 possesses achievement times which serve as an opt-in system for players to seek out extra challenges. Although we've not been able to adequately test these changes, our hope is that they add a greater sense of reward to players.

## Chapter 8

# Future work

While Continuity 2 could be released, there are a few things that can be done to greatly improve the game. For some of the remaining issues, solutions have been designed that need to be implemented. Other problems are more loosely defined and need to be addressed with further design work.

First of all, the menu system, as it appears in the game now, holds all the necessary information and provides all the necessary functionality but does not provide a smooth or pleasant user experience nor it is visually appealing. This is especially important for the level selection screen, which is meant to provide players a rewarding checklist of levels to complete.

Visuals for some of the game elements such as keys, wires, gates, levers and doors are also needed to provide visual consistency. Continuity 2 re-uses the two music tracks found in Continuity and adding new music is another relatively easy way to address the goal of the game being distinct from its predecessor.

Furthermore, adding the option to choose virtual buttons should be considered. It is trivial to implement and likely will not greatly affect other design decisions. Given that, it may make the game more appealing to players that are primarily familiar with playing games using gamepads.

Finally, level design is an area that needs more attention. Playtesting has shown that the two introductory levels designed to introduce the control scheme are unsatisfactory and so effort needs to be devoted to make these levels better introduce the game's controls. The game would also benefit from more levels in general. Further playtesting of the whole game is preferable so that one can investigate if the game has the appropriate difficulty curve. Once this playtesting is performed, levels that are too hard or easy can be removed or moved and new levels can be added.

Another matter that needs to be addressed before release is iPad compatibility. As the iPad was not released when this project started, it has not been targeted or considered in the design work. All of the iPhone applications available in the Apple App Store are however available for download

and use on both iPhone and iPad. The major concern with playing the game on the iPad is that rotation may be tedious due to the much larger form factor. Consequently, testing on the device is needed and iPad specific solutions should be explored.

## Chapter 9

# Conclusions

Continuity 2 possesses all of the same gameplay mechanics as the original Continuity. Because of the lack of change, the game should feel quite familiar to players of the original game. Although we attempted to modify the gameplay more radically through mechanics, such as touch-to-stop, these new mechanics worked less well than the original design.

We spent considerably more time in the divergence phase of development than we planned. This allowed us to invest more resources into taking risks. Our ability to tolerate discarding designs and implementations has allowed us to explore areas that we likely would not have otherwise explored. Further, we did not find ourselves stuck trying to make the most out of poor design decisions. Some of these risks, like touch-to-stop movement, did not work out. Others, such as the tacit and rotation control systems have proven fruitful.

The tacit control scheme works well for the low-intensity nature of Continuity 2. Unlike other systems that use touches and gestures, tacit controls promote repeatability of input and predictability of consequence over rich, precise actions. The lack of any visual component also suits the visual aesthetic of both the original game and sequel. However, tacit controls require more explicit instruction than virtual buttons. We feel confident that given more time, we will be able to properly explain the tacit controls within the game.

The rotation controls serve as a sizeable departure from the original game. They provide a way for controlling the character to be more potent and less punitive. While the rotation mechanics transfer some of the level creator's power over the player's path through the level to the player, it still seems possible to create challenging levels. Despite our misgivings, playtesters have indicated that they enjoy the rotation mechanics of Continuity 2 regardless of whether they previously played Continuity.

Some of the most predominantly distinct elements exist out-of-the game. The level selection system, achievement times and save system of Continuity

2 are entirely new. As we have been unable to thoroughly test these features, we are unsure of their contribution to the player's experience. Players may perceive these out-of-game additions as being crucial or largely irrelevant to their experiences with Continuity 2.

Although we do not know to what extent these out-of-game features affect the player's perception of Continuity 2, we are certain that the in-game aspects meet the goals we set out to accomplish. Specifically, the tacit controls, tile dragging and rotation controls provide the player with an experience that feels designed purposefully for a multi-touch device. As the majority of the gameplay mechanics remain unchanged from the original Continuity, the game provides enough familiar experiences that returning players understand the connection between the games. Lastly, the levers, wires and gates mechanic and rotation controls set the gameplay experience of Continuity apart from the original.

# Bibliography

- [1] The 12th annual independent games festival - about. <http://www.igf.com/01about.html> Retrieved 18-Jan-2010.
- [2] Analysis: History of cell-phone gaming. [http://www.businessweek.com/innovate/content/jan2006/id20060122\\_077129.htm](http://www.businessweek.com/innovate/content/jan2006/id20060122_077129.htm) Retrieved 03-May-2010.
- [3] Apple - ipad - view the technical specifications for ipad. <http://www.apple.com/ipad/specs/> Retrieved 20-May-2010.
- [4] Apple - iphone - compare. <http://www.apple.com/iphone/compare-iphones/> Retrieved 20-May-2010.
- [5] Apple - iphone 3gs - technical specifications. <http://www.apple.com/iphone/specs.html> Retrieved 20-May-2010.
- [6] Apple - ipod touch - view technical specifications for ipod touch. <http://www.apple.com/ipodtouch/specs.html> Retrieved 20-May-2010.
- [7] Apple - press info - apple announces the new iphone 3gs—the fastest, most powerful iphone yet. <http://www.apple.com/pr/library/2009/06/08iphone.html> Retrieved 20-May-2010.
- [8] Apple - press info - apple launches ipad. <http://www.apple.com/pr/library/2010/01/27ipad.html> Retrieved 20-May-2010.
- [9] Apple - press info - iphone 3g on sale tomorrow. <http://www.apple.com/pr/library/2008/07/10iphone.html> Retrieved 20-May-2010.
- [10] Apple - press info - iphone premieres this friday night at apple retail stores. <http://www.apple.com/pr/library/2007/06/28iphone.html> Retrieved 20-May-2010.
- [11] Evolution of mobile gaming. [http://www.forum.nokia.com/info/sw.nokia.com/id/c52ab94e-e29d-498a-a36a-e80296e4184a/Evolution\\_Of\\_Mobile\\_Gaming\\_1\\_0\\_en.pdf.html](http://www.forum.nokia.com/info/sw.nokia.com/id/c52ab94e-e29d-498a-a36a-e80296e4184a/Evolution_Of_Mobile_Gaming_1_0_en.pdf.html) Retrieved 02-May-2010.

- [12] First generation 2g iphone vs. second generation 3g iphone : Tech digest. [http://www.techdigest.tv/2008/06/first\\_generatio.html](http://www.techdigest.tv/2008/06/first_generatio.html) Retrieved 20-May-2010.
- [13] Fly on the wall. <http://www.ibm.com/developerworks/java/library/us-fly.html> Retrieved 20-Jan-2010.
- [14] igda - 2005 mobile games white paper. [http://archives.igda.org/online/IGDA\\_Mobile\\_Whitepaper\\_2005.pdf](http://archives.igda.org/online/IGDA_Mobile_Whitepaper_2005.pdf) Retrieved 03-May-2010.
- [15] iphone - wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki/IPhone> Retrieved 20-May-2010.
- [16] ipod touch - wikipedia, the free encyclopedia. [http://en.wikipedia.org/wiki/IPod\\_Touch](http://en.wikipedia.org/wiki/IPod_Touch) Retrieved 20-May-2010.
- [17] Nokia - snake game - mobile revolution - story of nokia - company - about nokia. <http://www.nokia.com/about-nokia/company/story-of-nokia/mobile-revolution/snake-game> Retrieved 09-Feb-2010.
- [18] Sebastiano Bagnara and Gillian Crampton Smith. *Theories and practice in interaction design*. Lawrence Erlbaum, 2006.
- [19] Marion Buchenau and Jane Fulton Suri. Experience prototyping. In *DIS '00: Proceedings of the 3rd conference on Designing interactive systems*, pages 424-433, New York, NY, USA, 2000. ACM.
- [20] Joseph Cartman and Richard Ting. *Strategic mobile design : creating engaging experiences*. New Riders, 2009.
- [21] Mihály Csikszentmihályi. *Flow: The Psychology of Optimal Experience*. Harper Perennial, 1991.
- [22] Alan Dix. *Human-computer interaction*. Prentice Hall, 2004.
- [23] Tracy Fullerton. *Game design workshop : a playcentric approach to creating innovative games*. Elsevier Morgan Kaufmann, 2008.
- [24] Mario van. Hamersveld and Cees de Bont. *Market research handbook*. John Wiley and Sons, 1998.
- [25] Lars Erik Holmquist, Kristina Hook, Oskar Juhlin, and Per Persson. Challenges and opportunities for the design and evaluation of mobile applications. <http://www.tii.se/mobility/Files/MobileChallenges020914.pdf>, 2002.
- [26] John Chris Jones. *Design methods*. Wiley, 1992.

- [27] Jesper Juul. *A Casual Revolution: Reinventing Video Games and Their Players*. The MIT Press, 2009.
- [28] Elina M. I. Koivisto. Mobile games 2010. In *CyberGames '06: Proceedings of the 2006 international conference on Game research and development*, pages 1–2, Murdoch University, Australia, Australia, 2006. Murdoch University.
- [29] Annakaisa Kultima. Casual game design values. In *MindTrek '09: Proceedings of the 13th International MindTrek Conference: Everyday Life in the Ubiquitous Era*, pages 58–65, New York, NY, USA, 2009. ACM.
- [30] Prasad V. Prabhakar, Martin G. Helander, and Thomas K. Landauer. *Handbook of human-computer interaction*. Elsevier, 1997.
- [31] Jakob Nielsen and Thomas K. Landauer. A mathematical model of the finding of usability problems. In *CHI '93: Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, pages 206–213, New York, NY, USA, 1993. ACM.
- [32] Donald A. Norman. *The design of everyday things*. MIT Press, 1998.
- [33] Alex Osborn. *Your creative power*. Myers Press, 2007.
- [34] Jim Rudd, Ken Stern, and Scott Isensee. Low vs. high-fidelity prototyping debate. *interactions*, 3(1):76–85, 1996.
- [35] Jesse Schell. *The art of game design : a book of lenses*. Elsevier Morgan Kaufmann, 2008.
- [36] Helen Sharp, Yvonne Rogers, and Jenny Preece. *Interaction design : beyond human-computer interaction*. Wiley, 2007.
- [37] Jason O. B. Soh and Bernard C. Y. Tan. Mobile gaming. *Commun. ACM*, 51(3):35–39, 2008.

## Appendix A

# Glossary of Terms

- Achievement time—The time under which the player tries to collect all the coins and complete a level. Each level has a preset achievement time.
- Adobe Flash Player—Software created by Adobe that is capable of SWF files in either a web browser plugin or using a stand-alone player.
- Character, the—The avatar that the player controls in Continuity and Continuity 2. The character is a genderless stick figure.
- Coins—Coins are optional, in-level bonus items.
- Continuity—A single-player puzzle platformer game for personal computers.
- Continuity 2—The sequel to Continuity, made for iPhone and the subject of this report.
- Continuity game, a—Any game that is part of a series of Continuity games.
- Door—Each level contains a single red door to which the character must bring all the level's keys in order to complete that level.
- Energized—the state of a wire or gate when connected by a series of wire segments to a lever. Energized wire segments are green.
- Game World—The environment through which the character can move that is represented upon the tiles.
- Gate—The blue game object that can be opened via levers when connected via wires.
- In-game—What occurs when the player is playing a level.

- iPhone—A mobile phone and computing device created by Apple, featuring a 3.5 inch multi-touch display. In this paper, this term is used to refer to both the iPhone and iPod Touch.
- iPod Touch—A computing device created by Apple similar to iPhone, but which does not function as a mobile phone.
- Key—An item type of which all must be collected to complete a level.
- Lever—A lever acts as a switch and toggles the state of gates to which it is connected via wires.
- Macro-mode—The gameplay mode from which the player manipulates the tiles.
- Macro-perspective—The zoomed-out view that is displayed when control is in in the macro-mode.
- Micro-mode—The gameplay mode from which the player controls the character.
- Micro-perspective—The zoomed-in view that is displayed when the control is in the micro-mode.
- Multi-touch—A display technology that can recognize and locate multiple fingers touching the display simultaneously.
- One-mode—A game design concept in which the player can simultaneously control the character—through the use of buttons—and the tiles—via touch—from a fixed perspective. This game concept replaces both micro-mode and macro-mode.
- Out-of-game—Any part of the application that is not experienced while playing a single level. These are parts of the software application such as the main menu, level selection screens, loading screens and level debriefing screens.
- $n$ -puzzle game—A sliding puzzle game in which a number of tiles exist in a grid with exactly one empty space. In the traditional game, the objective is modify a random configuration to a given configuration by sliding tiles into the empty space.
- Pause-menu—The menu presented when the player has paused the game while playing a level.
- Platformer—a video game genre characterized by jumping to and from suspended platforms or over obstacles (jumping puzzles)

- Smooth play—A goal for Continuity 2 in which the player spends longer periods of time controlling each mode of play rather than brief, stunted, sessions
- Tacit controls—An interface for controlling a platformer character that uses no visible user interface elements.
- Tiles—Each piece of the  $n$ -puzzle.
- Touch-to-Stop—A one-mode game design concept. The player directs the character's movement through the use of the buttons and the character continues to move in the indicated direction until told otherwise. When the player touches the screen, the character's world pauses.
- Two-mode—Gameplay mechanics that feature both a micro- and macro-mode.
- Wires—The colored connectors that are used to connect levers to gates.
- World—Each level belongs to a world. Worlds are either locked or unlocked. Players cannot play levels in locked worlds.