

# Gojin: The Five Gods

A Location Based Mobile Game for Tourism

A Major Qualifying Project Report

submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science and

Degree of Bachelor of Arts

By:

Natalie Bloniarz

Daniel Chao

Kailun Liu

Advised by

Professor Gillian Smith

## Abstract

Gojin is a mobile location-based game for English-speaking tourists visiting Kyoto, Japan. The goal of the game is to provide assistance in traveling between famous cultural landmarks and educate players about the location and culture. This is done while still maintaining the enjoyment of game content itself. In this paper, we discuss the design process we followed, from the inspirations for the designs and art, to the technical implementation of each section.

## Acknowledgements

We would first like to thank Noma-sensei, Lopez-sensei, Kohei-sensei, Ooi-sensei, the students we met at Ritsumeikan University, and the University as a whole for hosting our project. They gave us lab space to work, and were very kind in not only assisting our project when needed, but also socializing with us, giving us a great experience. Ritsumeikan University also provided us with safe and clean-living space at the BKC International House.

Next, we would like to thank our advisor Professor Gillian Smith for her support and feedback, overall enriching our MQP experience. Her advice was invaluable to our progression. We also appreciate her flying with us to Japan for three weeks to help us adjust and give more direct communication.

We would also like to thank Professor deWinter for her help during PQP, our Japanese culture and language preparation course. Her expertise in Japan as a whole gave us a great foundation to come into a new culture.

Finally, we would like to thank our playtesters. They played a tremendous role in the final refinements and data analysis of our project.

# Table of Contents

Abstract.....	i
Acknowledgements.....	ii
Table of Contents .....	iii
Table of Figures .....	vii
Table of Tables .....	ix
Table of Code .....	x
Chapter 1: Introduction .....	1
1.1: About the Project .....	1
1.2: Goal of the Project .....	1
1.3: About the Team .....	2
1.4: Paper Structure.....	2
Chapter 2: Background .....	4
2.1: Inspirations .....	4
2.2: Topics on Japanese Mythology .....	5
2.3: Culturally Relevant Locations in Kyoto.....	6
2.3.1: Fushimi Inari-Taisha .....	7
2.3.2: Nijō Castle.....	7
2.3.3: Yasaka Shrine / Gion District .....	8
2.3.4: Ryoanji .....	9
2.3.5: Ginkaku-ji .....	10
2.3.6: Myoshinji .....	11
Chapter 3: Design & Gameplay .....	13
3.1: Experience Description.....	13
3.1.1: Generalization of Audience.....	13
3.1.2: Our intended experience.....	14
3.2: Main Menu & Unlock System .....	14
3.3: Cutscenes .....	17
3.4: Map .....	18
3.4.1: How the Map Works with the Minigames.....	19
3.5: Minigames.....	19

3.5.1: Fushimi Inari Run .....	20
3.5.2: Nijo Castle Tower Defense .....	22
3.5.3: Yasaka Shrine / Gion District Rhythm Game .....	23
3.5.4: Ryoanji Line Trace .....	24
3.5.5: Ginkaku-ji Unblock the Moon .....	25
3.5.6: Myoshinji Temple Jigsaw.....	26
3.6: User Stories .....	27
3.6.1: Ken Tran .....	28
3.6.2: Steve .....	29
3.6.3: Dani .....	30
3.6.4: Carlie.....	31
3.6.5: HC Yung.....	32
3.6.6: HC Olde .....	32
3.6.7: Kebab Azdul .....	33
3.6.8: Yosuke .....	34
3.7: Experience Description.....	35
3.7.1: Generalization of Audience.....	35
3.7.2: Our intended experience.....	35
3.8: Paper Prototypes.....	36
3.8.1: Map.....	36
3.8.2: Fushimi Inari Catch the Mouse .....	37
3.8.3: Ryoanji Symbol Writing.....	38
3.8.4: Ball Maze.....	38
3.8.5: Ginkaku-ji Unblock the Moon .....	39
3.8.6: Myoshinji Temple Jigsaw.....	39
3.9: Iterative Design .....	40
Chapter 4: Technical Implementation.....	41
4.1: Main Menu & Unlock System .....	41
4.1.1: Persistent Data.....	43
4.2: Log System.....	44
4.3: Cutscenes .....	44

4.3.1: Dialogue.....	44
4.4: Map .....	45
4.4.1: Mapbox SDK .....	45
4.4.2: Camera Movement on Map.....	46
4.4.3: How Maps Interface with Minigames.....	48
4.5: Minigames.....	49
4.5.1: Fushimi Inari Run .....	49
4.5.2: Nijo Castle Tower Defense.....	51
4.5.3: Yasaka Shrine / Gion District Rhythm Game .....	53
4.5.4: Ryoanji Line Trace .....	54
4.5.5: Ginkaku-ji Unblock the Moon .....	55
4.5.6: Myoshinji Temple Jigsaw.....	56
Chapter 5: Art.....	57
5.1: Intro to Low Poly Art: Background and Inspiration .....	57
5.1.1: Definition of Low Poly Art Style .....	57
5.1.2: Inspirations from other games.....	58
5.2: Minigame Assets.....	60
5.2.1: Production Techniques.....	60
5.2.2: Fushimi Inari Run .....	63
5.2.3: Myoshinji Temple Jigsaw.....	63
5.2.4: Nijo Castle Tower Defense.....	64
5.2.5: Yasaka Shrine / Gion District Rhythm Game .....	65
5.2.6: Ginkaku-ji Unblock the Moon .....	66
5.3: Characters .....	67
5.3.1: Production of Characters .....	67
5.3.2: Overall Design and Inspiration .....	68
5.3.3: God Inari.....	69
5.3.4: God Susanoo .....	70
5.3.5: God Tenman Tenjin .....	71
5.3.6: Akashita.....	72
5.3.7: Yamata no Orochi .....	73

5.3.8: Sugawara no Michizane .....	73
5.4: Music .....	74
Chapter 6: Playtesting, Feedback.....	75
6.1: Background of Playtesters (Pre-Survey information).....	75
6.2: How Playtesters Felt (Post-Survey Information) .....	77
6.2.1: Minigame Ratings.....	78
Chapter 7: Postmortem .....	80
7.1: What was successful.....	80
7.1.1: Team Dynamics (and Division of Labor).....	80
7.1.2: Integration.....	80
7.1.3: Scoping.....	81
7.2: What could have been improved .....	81
7.2.1: Team Communication .....	81
7.2.2: Difficulties in Acquiring Playtesters .....	81
7.2.3: Game Playability.....	83
7.2.4: More options .....	86
7.3: Areas of future exploration .....	86
Chapter 8: References.....	88
Appendix A: Pre-Playtesting Survey .....	93
Appendix B: Post-Playtesting Survey.....	97
<b>Minigame Rating</b> .....	100
Appendix C: Playtesting Data .....	102
Pre-Playtest Survey .....	102
Post-Playtest Survey.....	108
Appendix D: Joint Ritsumeikan-WPI Puzzle Solving Robot .....	116
What We Learned .....	116
What We Would Do Differently.....	117

## Table of Figures

Figure 2.1: Fushimi Inari-Taisha Gates [1].....	7
Figure 2.2: Nijo Castle Guard Tower [10] .....	8
Figure 2.3: Yasaka Shrine [39].....	9
Figure 2.4: Ryoanji Zen Rock Garden [4] .....	10
Figure 2.5: Ginkaku-ji Pavillion [36] .....	11
Figure 2.6: Temples at Myoshinji [5] .....	12
Figure 3.1: Main Menu Screen.....	14
Figure 3.2: Unlock System Diagram .....	15
Figure 3.3: Tutorial Screen for Yasaka Shrine / Gion District Rhythm Game .....	16
Figure 3.4: The Loading Screen .....	17
Figure 3.5: One Cutscene for Ryoanji Line Trace .....	18
Figure 3.6: Map Screen with Player and Fushimi Inari-Taisha Prefab .....	19
Figure 3.7: Obstacles for Fushimi Inari Run .....	21
Figure 3.8: Screenshot of Fushimi Inari Run.....	21
Figure 3.9: The Three Towers.....	22
Figure 3.10: Screenshot of Nijo Castle Tower Defense.....	23
Figure 3.11: Screenshot of Yasaka Shine/ Gion District Rhythm Game .....	24
Figure 3.12: Screenshot of Ryoanji Line Trace .....	25
Figure 3.13: Screenshot of Ginkaku-ji Unblock the Moon.....	26
Figure 3.14: Screenshot of Myoshinji Jigsaw Puzzle .....	27
Figure 3.15: Map Paper Prototype .....	37
Figure 3.16: Display of Badges and Minigame .....	37
Figure 3.17: Sample Strokes for Ryoanji Line Trace.....	38
Figure 3.18: Cardboard Model of the Ball Maze .....	38
Figure 3.19: Sample Slide Puzzle.....	39
Figure 3.20: Sample Paper Jigsaw.....	39
Figure 4.1: Rendered 3d Buildings.....	46
Figure 4.2: Camera Movement Diagram .....	47
Figure 4.3: Pinch to Zoom Diagram .....	48
Figure 4.4: Swipe System .....	50
Figure 4.5: Left vs Right, Valid vs Invalid Stroke.....	55
Figure 5.1: Screenshot of Shelter [30].....	59
Figure 5.2: Screenshot of Epistory - Typing Chronicles [6] .....	59
Figure 5.3: Diagram of Tree Asset Creation .....	61
Figure 5.4: Half of the Spider Model in ZBrush.....	62
Figure 5.5: The Material Panel for the Shamisen.....	63
Figure 5.6: The Jigsaw Puzzle Scene in 3ds Max .....	64
Figure 5.7: Arrow Tower Levels from Left to Right: 1, 2, 3.....	65
Figure 5.8: Model of the Shamisen.....	66



Figure 5.9: The Lantern Model for the Note.....	66
Figure 5.10: The Cloud Models.....	67
Figure 5.11: Noh Masks [38] .....	69
Figure 5.12: The Model of the God Inari and His Mask .....	70
Figure 5.13: Susanoo and His Mask.....	71
Figure 5.14: Tenman Tenjin and His Mask .....	72
Figure 5.15: Akashita in a Traditional Image [46] and the Model of Akashita.....	73
Figure 5.16: Yamata no Orochi Model.....	73
Figure 5.17: Sugawara no Michizane Model and Mask Model .....	74
Figure 6.1: Types of Mobile Games .....	76
Figure 6.2: Time Playing Mobile Games .....	76
Figure 6.3: Playtesters Feelings on Gojin .....	77

## Table of Tables

Table 4.1: Minigame Statuses.....	42
Table 6.1: Playtester Ratings of Minigames.....	79

## Table of Code

Code 4.1: Example of Serialized Classes .....	43
Code 4.2: Generation of Objects in Fushimi Inari Run.....	51

# Chapter 1: Introduction

## 1.1: About the Project

Gojin is a location based mobile game for Android where players work to restore the power of five Gods through exploring and visiting real world locations in Kyoto. The players learn about each location through cutscenes and minigames based off of the history and mythology of their current location.

We decided developing on an Android mobile platform was the best option, because we wanted to create a location-based game to achieve an explorative real-world experience. The major advantage of developing for the Android operating system is that it allows applications to be directly installed to the phone, rather than having to go through the Apple App Store vetting process to allow for the app to be installed, making testing easier. The game was developed using the Unity Engine, a cross platform game engine developed by Unity Technologies [48].

## 1.2: Goal of the Project

The goal of this project was to understand if tourists would use a game to help explore the city of Kyoto. The intended experience we wanted to provide our audience is one of exploration, learning, immersion and enjoyment. When our audience arrives in Kyoto, they presumably want to explore or to learn about the culture. The player is guided to a point of interest (POI), then be prompted to participate in a minigame experience related to the location. Through these experiences the players will learn about the history of the POIs or what the location is known for, and they are incentivized to look around and explore the various aspects of each location using the 3d map. The

players can then freely move to another POI once they feel they are finished with their current location.

To accomplish the goal of our project we needed to research historical locations in Kyoto, Japanese mythology, and similar games to understand what makes them successful.

At the completion of our project we feel that our concept was viable, however our current game lacks certain features that are important for our desired experience. We feel this way because of how hard it was to acquire playtesters, and their feedback involving certain components of our game. We feel that if we were to expand on this project, we would work to integrate more story and gameplay needed to meet our initial goal.

### **1.3: About the Team**

The team consists of Natalie Bloniarz, Daniel Chao, and Kailun Liu. Natalie is a Computer Science and Interactive Media & Game Development double major. She has had previous experience with Unity, which led to the decision to use that engine. Daniel is a Computer Science major with light familiarity using Unity, and has also done a previous project in Japan. Kai is an Interactive Media & Game Development major specializing in art. He has skills in modeling, rigging, texturing and animating. Most of the art assets in the game are created by him.

### **1.4: Paper Structure**

In this paper we describe the process of creating Gojin and the various elements needed to create the fully fleshed out game, and challenges that arose during development, and possible future work to expand on the finished project. In Chapter 2, we explore the related background

information of the project including our inspirations, Japanese mythology, and locations relevant to our game. In Chapter 3, we discuss the intended experience of Gojin and the design of major elements such as the map system and the minigames. In Chapter 4, we explain the technical implementation of each system of the game. In Chapter 5, we discuss the art style and production techniques used to create the assets in the game. In Chapter 6, we analyze the data from our playtesters, and the conclusions we can draw from that information. In Chapter 7, we summarize our experience working on Gojin including what went well, what could have been improved, and what future work could be implemented. In the Appendices, we include the materials from our playtesting session including surveys and the remaining data, and the result of the joint puzzle solving robot project with Ritsumeikan students.

## Chapter 2: Background

In order to create the experience we were aiming for, we needed to first research the inspirations for our game. We researched location based mobile games, and other related games, to understand the field of gameplay we are entering. Additionally, because our game is centered around educating people about Japanese culture while exploring various locations, we needed to study both the culture of Japan and these locations first to understand how to properly portray them.

### 2.1: Inspirations

The two main inspirations for our game were *Pokémon Go* [33] and the *Warioware* [15] series. *Pokémon Go* is a location based mobile game developed by Niantic Labs where players can catch Pokémon in the real world through augmented reality. *Pokémon Go* also highlights real world locations such as historical landmarks and points of interest by labeling them as PokéStops, which players are incentivized to visit. We were inspired by the PokéStop system on the main map screen to select culturally relevant locations, and direct our player to those locations. We decided to use a 3d map as a tool to get players to directly explore their current location, and make the experience more interactive with the location. Our minigame system was inspired by the *Warioware* series; these games, created by Nintendo, are composed of micro games where the player has to complete a simple task in a short period of time. We were also inspired by analog games such as scavenger hunts; these games often lead people around a set area, have a system of clues to direct players to the next location, and often have specific goals player must accomplish.

## 2.2: Topics on Japanese Mythology

Due to the theming and story of our game, it was necessary to research Japanese mythology. Our story involves five gods, and their fight with various evils to restore their power, thus we had to investigate the gods of Japan, as well as yokai culture. The gods and evils described in this chapter are the ones we chose to use in our game.

The original source of much Japanese mythology is *Kojiki*, (Title Translation: Record of Ancient Matters) in which the tale of how both gods and yokai came into being. The story says that Izanagi, one of the two original deities, bathes in a bath to purify himself. The water droplets that fall from his body are what turns into all of the gods and yokai people know today [51]. Among these, we chose the ones that went with our locations.

For Fushimi Inari-Taisha, we chose Inari, referred to as the god of rice and prosperity. Inari is always associated with foxes in mythology in some way, either having foxes as messengers, or being a fox himself [2]. This relationship is theorized to be because of how foxes protect rice fields by eating harmful rodents [21]. In contrast to foxes, Akashita is a yokai that is known for wreaking havoc on farmers. Japanese rice farms are always kept extremely well irrigated. During times of scarce water, Akashita punishes farmers who steal water from other people's farms [29]. He is portrayed as a monster hiding within a thick cloud, with a long tongue and sharp claws [47].

Susanoo (also written Susa-no-wo) is the god honored at Yasaka Shrine in Gion. People go there to pray to him and his family for love and relationships. He is known for his mischief, as well



as his anger, as he was ultimately banished from heaven to the human world due to his poor behaviors [26].

His traditional rivaling evil that we use in our game is Yamata no Orochi, an eight-headed monster. When he arrived in the human realm, he learned of Yamata no Orochi terrorizing a small family by demanding one sacrifice every year [28]. Susanoo ended up killing Yamata no Orochi by luring him out with sake, and married the daughter of the family because of this [22].

For Ryoanji, we wanted to keep the theming of gods and evils, so we used a deity from a nearby shrine. Sugawara no Michizane (also known as Tenman Tenjin), is the god of education honored in Kitano Tenmangu Shrine. Despite being worshipped as a god, he is also feared as one of the strongest evils in Japan when he is displeased. He wreaked havoc on Japan after his death. Because of this dual nature, we selected Tenman Tenjin (his name after enshrinement) to be the god, and Sugawara no Michizane (his name when he was causing destruction) to be the evil [27].

### 2.3: Culturally Relevant Locations in Kyoto

For our project, we used historically and culturally relevant locations as key POIs, because these are the places tourists want to visit regardless of playing the game. Making the game progression similar to a standard travel path is important, as the game is supposed to help guide their travels, rather than impose a drastically different route.

The history of these places was integral to designing a culturally significant experience for our players. We also picked these locations because they have distinct enough features that could be

simplified into familiar minigames. Many of these locations also do not come with a lot of English support, such as signs, so our game could be particularly helpful for tourists.

### 2.3.1: Fushimi Inari-Taisha

Fushimi Inari-Taisha is the head (and largest) Inari shrine, dedicated to the Shinto god of rice Inari. The public image of Japanese shrines is often that of Inari shrines, specifically their bright orange torii gates. People come here to pray for bountiful harvests, business prosperity, safety, and many other things. One of the most famous features of the shrine is the long path up the mountain with thousands of Torii gates lining it [8]. The gates begin to increase in size as one climbs further up to the top of the shrine. When our group went there and traveled up the mountain path, we got a feel for how endless the torii gates felt, and saw certain obstacles that we put into our game.



*Figure 2.1: Fushimi Inari-Taisha Gates [1]*

### 2.3.2: Nijō Castle

Nijo Castle was the residence of the first shogun of the Edo Period, being fortified by two sets of large walls and moats to stop attacks. It is one of the best surviving examples feudal

architecture [17]. The double fortification layout was very pronounced in the route we walked when we went, therefore was the main inspiration for the location's respective minigame.



*Figure 2.2: Nijo Castle Guard Tower [10]*

### 2.3.3: Yasaka Shrine / Gion District

Gion is famous for being the largest geisha district. The entire area is very dedicated to the history and culture, having many elegant places to dine, teahouses, and traditional forms of entertainment. The district was built to accommodate the needs of visitors to the shrine.

There are many smaller shrines and temples within Gion, the biggest and most popular shrine being Yasaka Jinja. People come here to be blessed with love [18, 49] and to experience the Geisha culture. This is why we modeled our minigame on the shamisen, a traditional Geisha instrument.



Figure 2.3: Yasaka Shrine [39]

#### 2.3.4: Ryoanji

Ryoanji Temple was constructed as a Zen-Buddhist temple in the 1400's. It is most famous for their Zen rock garden, containing 15 larger rocks surrounded by raked gravel, and a large display of calligraphy writing. A lot of the features seen here follow the zen theme, giving people something to contemplate [37]. The zen theme and calligraphy served as the major source of inspiration for the minigame. Nearby Ryoanji is Myoshinji, the head temple for the same sect of Buddhism.



*Figure 2.4: Ryoanji Zen Rock Garden [4]*

### 2.3.5: Ginkaku-ji

The Silver Pavilion, Ginkaku-ji, is a Zen temple with a number of similarities to the famous Golden Pavilion, Kinkaku-ji. Both are Zen temples, built as retirement villas, and have similar architectures as Ginkaku-ji was modeled off of the Golden Pavilion. It holds a lot of cultural relevance, due to holding several works of art, several other temples, a moss garden, a dry sand garden, and a reflective lake.

The name “Silver Pavilion” was not originally associated with Ginkaku-ji. It came about after the construction, both to contrast the Golden Pavilion, and because moonlight reflecting off the dark exterior (previously covered in black lacquer) made the surface look like silver [16]. We took inspiration from the origin of the name to create our minigame.



*Figure 2.5: Ginkaku-ji Pavilion [36]*

### 2.3.6: Myoshinji

Myoshinji is a large complex of over 50 Buddhist temples, surrounded by a large wall. There are several more temples in the surrounding areas. Many of these temples and walkways are culturally significant by themselves. Most of the sub-temples are closed to public as monks still use them, however visitors can enter some of them, and view them from a public pathway [32]. When we went there, the sheer number of temples is what made the experience what it was, so we wanted to capture that notion in the minigame.



*Figure 2.6: Temples at Myoshinji [5]*

## Chapter 3: Design & Gameplay

The design of the game focuses heavily on encapsulated experiences within the minigames and on the map. To achieve the desired experience, we chose to focus on user stories for potential players, and base our design decisions on these potential players. We also created paper prototypes to test out initial minigame designs and other systems.

### 3.1: Experience Description

#### 3.1.1: Generalization of Audience

Our target audience, broadly, is English-speaking visitors in Kyoto. We targeted 18-40 year olds, because that age range makes up a large part of the United States gaming population [41], and a significant portion of the global gaming population [40]. Though we have a specific target age, people of other ages may play as well. All ethnicities are part of our target audience, as it does not affect the gameplay or cultural experience the application tries to deliver to them. We expect tourists to have low or moderate cultural knowledge, because if they are already familiar with Japan they have much less motivation to use our application. We expect that they have some experience with mobile games, but it is not required. Mobile gamers are the most likely to find our game, because they are the ones most likely to search for this genre in the app stores. People with physical disabilities are not included in our target audience, as our game requires a player to perform the physically-intensive task of exploring all over Kyoto. Members of our audience are most likely be traveling alone, due to the game being single-player, as well as requiring greater navigation aids if you are traveling by yourself.



### 3.1.2: Our intended experience

The intended experience we wanted to provide our audience is one of exploration, learning, immersion and enjoyment. When our audience arrives in Kyoto, they presumably want to explore, either to play or to learn about the culture. The player is be guided to a POI, then instructed what to do there to follow cultural etiquette. They also learn some history about the POI, or what it is known for. Players are incentivized to look around and explore the various aspects of each POI. They then can move to another POI once they feel they are finished with the one they are on.

### 3.2: Main Menu & Unlock System

The main menu serves as the hub to access all parts of the game. The player can access both the map and the minigames from the menu, and this creates a cohesive screen that the player can expect to return to. The other purpose of the menu is to allow for replayability of the game through the unlock system.

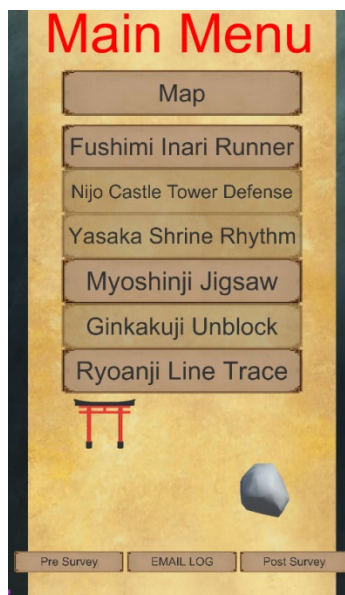


Figure 3.1: Main Menu Screen

The unlock system gives the player the opportunity to replay the minigames after they leave the location the minigame is tied to. When the game is opened for the first time all of the buttons aside from the map are disabled, and the badges are hidden. To unlock a game the player must visit the real-world location of the game and tap on the icon on the map. From this the player is shown a cutscene on the first visit to explain the story of the location. Once the player enters the initial cutscene, the minigame button is unlocked on the menu. If the player does not finish the initial cutscene, the menu button brings them back to the cutscene so that they can understand the full story. Once they finish the cutscene, the button switches to take the player exclusively to the minigame. When the player reaches a specific win condition set by each minigame for the first time, they unlock the badge for that game and are shown another cutscene. The badges on the menu serve as an indication to the player that they have beaten the minigame.

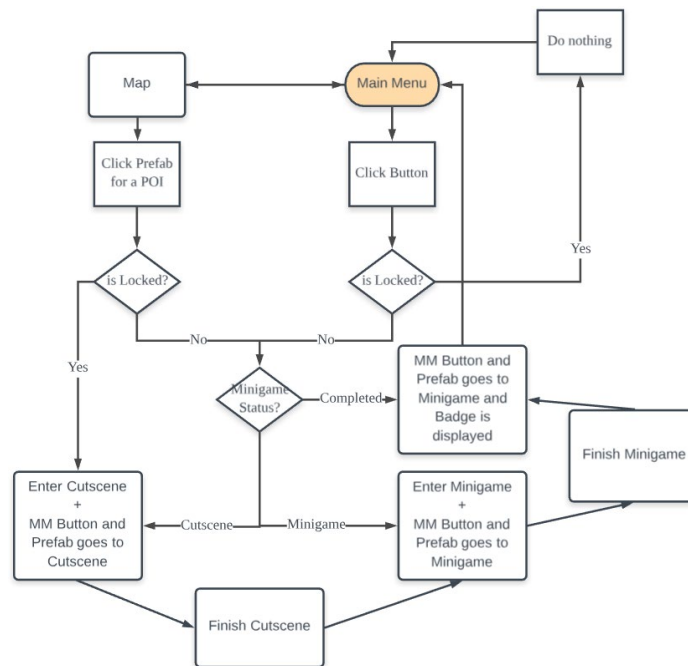
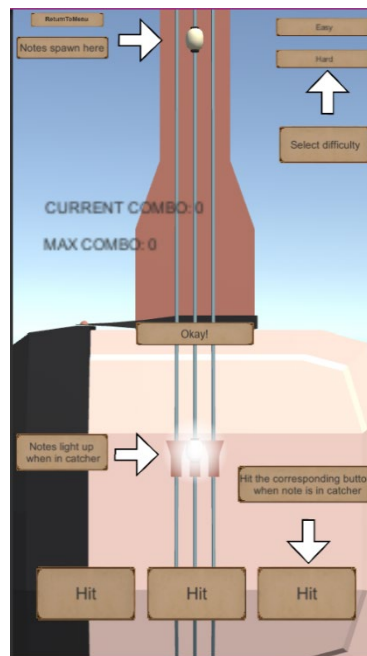


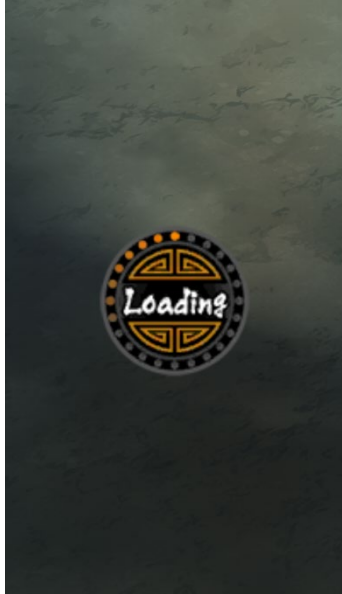
Figure 3.2: Unlock System Diagram

The unlock system also handles tutorials. The first time the player enters a minigame, they are presented with a tutorial as shown in Figure 3.3. The tutorials have an acknowledgement system in the form of a button to dismiss the tutorial once the player feels they understand the game. When the button is pressed, it will close the tutorial permanently; the player will not see it again even when re-entering the minigame.



*Figure 3.3: Tutorial Screen for Yasaka Shrine / Gion District Rhythm Game*

To allow for the mobile phone to process loading each minigame, we created a loading screen. The screen appears to let the player know that the game is being prepared in the background and not freezing. The dots circle around the loading circle to alert the player that the game is being prepared.



*Figure 3.4: The Loading Screen*

### 3.3: Cutscenes

Before and after certain minigames, the player sees a cutscene featuring a related god and evil character. The god character will narrate, describing some of the history of the location, story elements such as why they are talking to the player, and why the player is playing the minigames. When the player finishes the dialogue, they may proceed to the minigame.



Figure 3.5: One Cutscene for Ryoanji Line Trace

### 3.4: Map

The map that the player can see is a real-time 3d map, centered on the player's actual location. The map concept we used is similar to that of *Pokémon Go* [33], with the addition of a small guidance system. The player is able to see the roads, the roads' names, 3d models of buildings, and highlighted POIs. The POIs are displayed differently, based on what they are: either there is a small icon showing what kind of POI it is, or a custom 3d model in the case that the POI has a minigame.

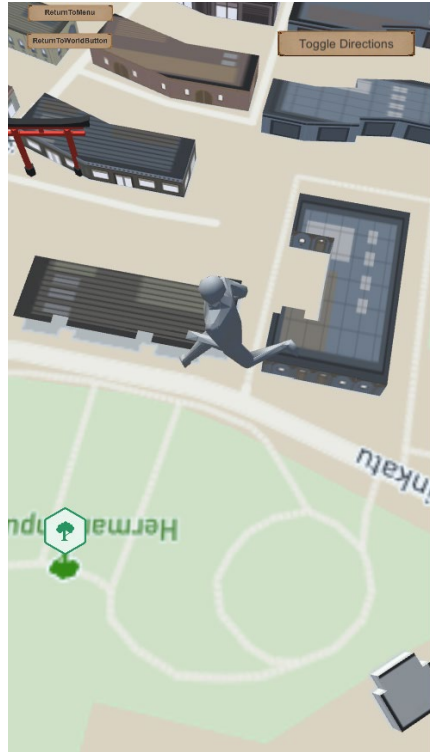


Figure 3.6: Map Screen with Player and Fushimi Inari-Taisha Prefab

### 3.4.1: How the Map Works with the Minigames

The map provides the player a way to access the various minigames, based on the player's location. Smaller POIs, such as other temples and shrines without minigames, serve to highlight a possible place to visit similar to a PokéStop in *Pokémon Go*. The locations we have deemed more culturally significant POIs are designated with 3d models on the map, to attract the attention of the player. These bigger POIs, when visited, open a non-player character (NPC) cutscene dialogue, that eventually prompts the player to play the corresponding minigame for that location.

### 3.5: Minigames

Specific locations that they player can explore in Gojin have a minigame associated with it. These minigames are based on the history or cultural significance of the location:

- Fushimi Inari-Taisha - An endless runner game where the player avoids obstacles and collects onigiri.
- Nijo Castle - A tower defense game where the player must defend from enemy samurai.
- Yasaka Shrine / Gion District - A rhythm game where the players tap along to lantern notes traveling along a shamisen.
- Ryoanji - A “follow the leader” style game to replicate strokes in kanji characters.
- Ginkaku-ji - A slide puzzle where players aim to shine moonlight on the temple.
- Myoshinji - A two layered jigsaw puzzle, where players must reassemble the map of Myoshinji.

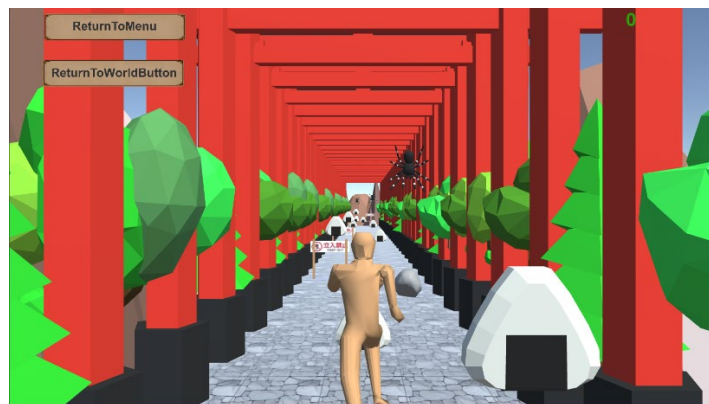
### 3.5.1: Fushimi Inari Run

Fushimi Inari Run is an endless runner minigame where the player must collect tokens and avoid obstacles. The player swipes up, down, left, and right, to move the character across the screen. The design of the game is inspired by the hike up to the top of the shrine, where hikers are enclosed in a tunnel of 10,000 gates (Figure 2.1). The player needs to collect onigiri to gain points. When the player reaches a certain score, the game will recognize the player has won. The collection system was inspired by the multiple waypoints along the trail where people can buy various goods. The design of the game is based off of the mobile game *Temple Run* [14], where players constantly run forward, collecting coins and dodging obstacles. There are three types of obstacles: a spider, a barrier, and a stone; these are based off of actual obstacles seen at the shrine by the team.



*Figure 3.7: Obstacles for Fushimi Inari Run*

The spider is based off of spiders seen making webs on the torii gates, the barrier is based off of one used to mark a closed trail, and the stone is based off of the uneven pathway steps up to the top of the shrine. The player must dodge these obstacles by either moving side to side, jumping over it, or sliding under it. Once the player achieves a score of 11 for the first time, they are taken to a cutscene and then are returned to the main menu. When the player dies, they are returned to the menu screen.

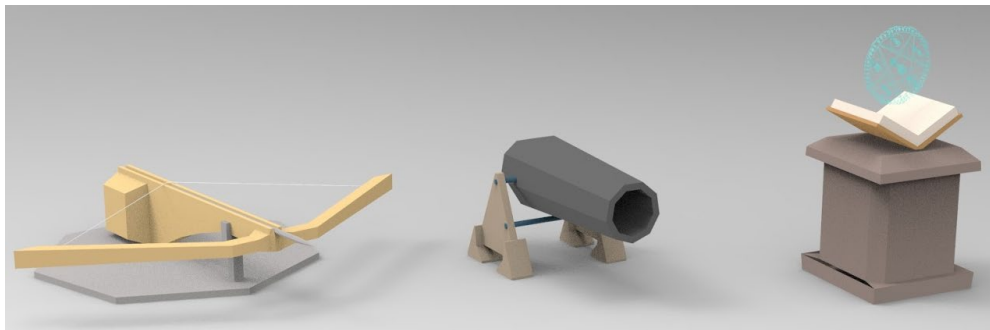


*Figure 3.8: Screenshot of Fushimi Inari Run*



### 3.5.2: Nijo Castle Tower Defense

The tower defense game of Nijo Castle is based off the popular tower defense game *Bloons Tower Defense 5* [34]. The player can see the highlighted path that the enemies will travel along. The player also has a budget to build towers, and each time they build a tower, money is subtracted. The three types of towers that can be built are: a tower that fires projectiles radially to a set distance, a tower that targets the first enemy that is within its range, and a tower slows down the enemies to half of their initial speed.



*Figure 3.9: The Three Towers*

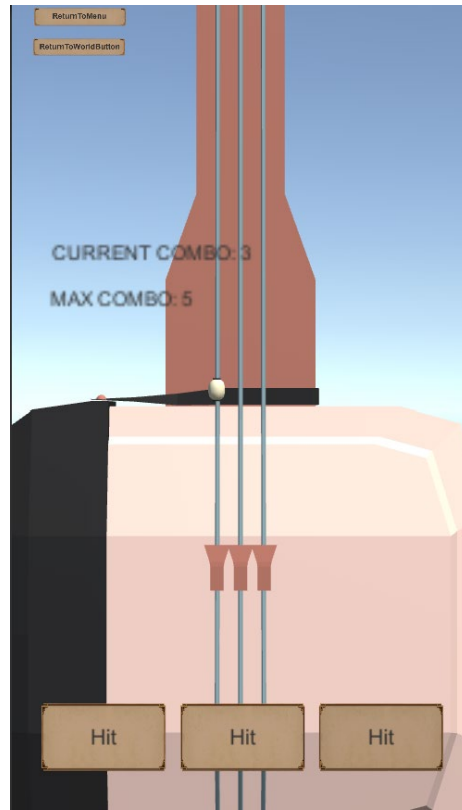
For each enemy the player destroys they get a set amount of money to buy new towers with for the next wave. The player also has a set health of 100, and for each enemy that reaches the end of the path, 10 health is subtracted. Once the player reaches a minimum score, they are taken to a cutscene, otherwise the game continues until the player loses all of their health and then they are returned to the main menu.



Figure 3.10: Screenshot of Nijo Castle Tower Defense

### 3.5.3: Yasaka Shrine / Gion District Rhythm Game

This rhythm game is based off of the mobile game *Tap Tap Revenge* [42], and the popularity of Japanese rhythm games. There are three tracks where notes can travel down towards the player. The design of the tracks is based off of the traditional Japanese instrument, the shamisen. The notes that travel along the strings of the instrument are shaped to resemble traditional paper lanterns. This decision was made because the Yasaka Shrine is known for the large number of lanterns that surround the stage in the shrine. The player then must hit the button of the corresponding track when the note reaches the catcher. When the note hits the catcher, the note illuminates to highlight to the player when to tap. If the player successfully taps the button when the note is in the catcher, the current combo of the player increases; if the player fails to tap the catcher at the correct timing, the current combo is reset to 0. The game also keeps track of a maximum combo that the player has achieved. If the player reaches a certain combo, the game will show that they have won.



*Figure 3.11: Screenshot of Yasaka Shinel Gion District Rhythm Game*

### 3.5.4: Ryoanji Line Trace

The inspiration for this game is the rock garden and calligraphy at Ryoanji. The stone garden at the temple has intricately drawn lines in the sand, and a collection of calligraphy in the hall surrounding the garden. The main mechanic of this game is replicating the stroke shown. The player is shown an image of a stroke to complete and then must replicate it by touching and dragging on their phone. We wanted to make this game feel peaceful, so there is no time limit on how long the player can take to draw or view the current stroke the player is up to. The visual design of the game is also simple to reduce distractions to the player, and is set to resemble the calligraphy scrolls in the halls around the rock garden. After the player completes the stroke, they are presented with the next stroke, and the completed stroke is drawn onto the scroll on the side of the screen. The final

characters the player ends up writing are 枯山水, which translates to a traditional dry landscape or Japanese rock garden.

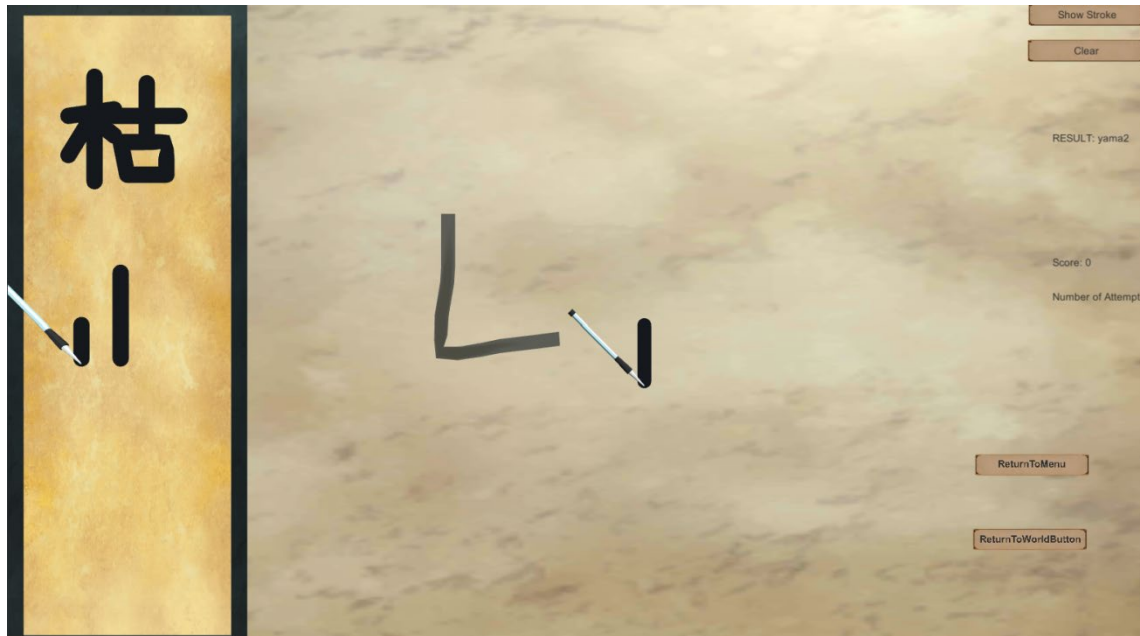


Figure 3.12: Screenshot of Ryoanji Line Trace

### 3.5.5: Ginkaku-ji Unblock the Moon

This game is modeled after *Rush Hour* [35], *Unblock Me* [23], and other similar slide puzzle games, where the player has to free the “main piece” by moving other pieces out of the way of the exit. The game is played on a 6x6 grid, where pieces can move either vertically or horizontally, depending on the type of piece in question. Pieces are 1x2, 1x3, 2x1, or 3x1 and move in the axis that they have a longer length for. This inspiration for this is the fact that Ginkaku-ji is commonly called the Silver Pavilion. However, the appearance of the temple is not actually that of silver, but rather of traditional Japanese architecture. The reason for its name is not only as an intentional contrast to the Golden Pavilion, but because the appearance when the moon reflects off of it is said

to have a silvery appearance. Due to this, the game has the “main piece” represent moonlight and the obstructing pieces represent clouds. When the clouds are cleared and the main piece reaches the exit, the light from the piece strikes Ginkaku-ji and illuminates the roof of the temple. There is no limit to the number of moves the player can take and there is no time limit. For added replayability, the levels are generated from a pool of positions that have solutions.



*Figure 3.13: Screenshot of Ginkaku-ji Unblock the Moon*

### 3.5.6: Myoshinji Temple Jigsaw

The primary inspiration for this game was the fact that the main attraction of Myoshinji is the plethora of temples located in the area. The puzzle is based directly off of the map of the temple complex. There are about 30 smaller temples within the grounds of Myoshinji, and 10 more very close by. It is one of the biggest temples of its kind. The size and number of temples that not only

this branch has, but has in one place is part of what makes the place so magnificent, so we wanted to emphasize it in hopes that it would increase appreciation for what the structure is about.

To add a bit of complexity, we have a two-layer puzzle. One layer is the ground shape, and the other layer is for temple pieces. The temple pieces fit on top of the ground pieces, so the ground pieces have to be placed prior to the corresponding temple.



*Figure 3.14: Screenshot of Myoshinji Jigsaw Puzzle*

### 3.6: User Stories

To create these user stories, and to help define our target audience and target experience, we created a set of descriptors to identify our hypothetical players:

1. Name

2. Where they're from
3. Their experience with games
4. How long they're in Kyoto
5. Are they playing as soon as they get to Kyoto?
6. Gender
7. Age
8. Ethnicity
9. Experience with Japanese culture
10. Interested in tradition, anime, food. Type of interest
11. Experience with mobile games
12. Physical disabilities
13. First time in Japan?
14. Are they traveling alone?

These 14 descriptors formed the basis of the following user stories, and allowed us to more accurately imagine the experiences of our players.

### **3.6.1: Ken Tran**

Ken Tran is a 20 year-old who graduated college recently, and is visiting Japan for vacation because he wanted to experience the culture. He arrives in Kyoto in hopes of exploring the city on his own and immersing himself in the culture, but quickly realizes that getting around by himself is more difficult than he had anticipated.

He opens the game which gives a brief introduction and then provides a list of nearby POIs. Ken sees a big shrine nearby (Kitano Tenmangu) and decides that is as good place as any to explore. He goes there, somewhat hesitant about where to go or what to do after he enters the gate, he sees no signs in English anywhere. The game tells him to bow at the gate, and to wash his hands at the cleansing station, as a list of tasks, currently 0/2. Ken feels somewhat more comfortable after some basic interaction with the shrine. Ken explores the rest of the shrine, having enjoyed what he sees so far. He follows along with the game as well, able to play while he explores. Upon completion of the shrine in the game, Ken moves on to the next closest location, not knowing much about which place is more interesting than others, and enjoys the rest of his day traveling from location to location.

### **3.6.2: Steve**

Steve is an All-American 21 year-old male on his first trip to Japan, he was convinced to go by his girlfriend who is a heavily interested in anime and Japanese culture. They have a week long trip in Japan, 4 days in Tokyo and 3 days in Kyoto. After 4 days of being paraded around to arcades and he decides to take a day to himself and explore Kyoto. He looks for a guide in the app store and finds a game that says it helps players to explore and learn about Kyoto; he downloads the app. It first takes him to a small shrine he had wandered past on his way to get a drink, and explains that the god of this shrine is the god of education. He sees a group of high schoolers writing on blocks of wood and hanging them up in the shrine, he sees one in English of a student asking to get into the college he goes to in the US. After he leaves the shrine he decides to walk to the next location, taking a slight detour to go get some food from a nearby convenience store. When he leaves the convenience store, he decides to walk down side streets to have a nice relaxed paced afternoon stroll.



He eventually stumbles upon Kiyomizu-dera and sees lots of couples taking photos. He walks inside and the game tells him that this a famous place where couples go to see if their relationships will last, and plays a fun matchmaking game. He decides this place is somewhere that he wants to come back to and take his girlfriend. Much later when he finishes walking and exploring the side streets of Kyoto, Steve tells his girlfriend about Kiyomizu-dera that they have to go to in order to have their fortunes told about their relationships. Steve has a greater appreciation for some aspects of Japanese culture, but still can't really understand his girlfriend's obsession with anime boys.

### 3.6.3: Dani

Dani is an Asian 19 year-old female on her first trip to Japan. She is traveling with her American boyfriend Steve on a week long trip. The first four days were spent in Tokyo, and the last three days are in Kyoto. Upon arriving in Kyoto, her boyfriend decided to let Dani explore on her own, saying he just wants to explore, while Dani continues shopping. Dani agrees, however upon exploring, Dani discovers that Kyoto does not have nearly as much anime culture as she has been exposed to the last four days, after spending most of her time in Akihabara. She starts to wonder what to do while Steve is out. Steve is usually the primary navigator for cultural attractions, so she isn't as familiar with where to go. She looks in the app store and finds a guide application and realizes that even without much cultural information about Kyoto she can still explore.

She gets excited about exploring again and boots up the game, skipping somewhat fast through the introduction, to get to the guiding aspect. She looks at the icons of several POIs on the map, and decides Fushimi-Inari sounds relevant to her interests, since it reminds her of some of the games she has played. She is explained the relevance of the shrine, what it stands for, and prompts

her to try walking up the path of “one thousand torii gates.” She walks up part way, making frequent stops and playing the game to take a rest from walking. She is at the shrine for several hours, before getting tired and wanting to meet back up with Steve for dinner. She passes a few smaller locations on the way back to the hotel.

#### **3.6.4: Carlie**

Carlie is an elderly man in retirement, who is traveling with his family in Kyoto for a week. His family consists of his wife, son, son’s wife, and 12-year old grandson. His son mentions that he heard about a new guiding application/game for Kyoto and thought it would be a fun way to give his family something to do together, allowing both Carlie and his grandson to participate. Carlie agrees and opens up the app. The family decides together to go to the closest shrine, Heian Shrine, despite it being a smaller shrine. Since traveling together is slow, tedious, and at times strenuous for Carlie, going to closer locations is more important. The family explores the shrine while playing the minigame. Carlie enjoys the game, being a way to interact with his grandson; teaching his grandson about the culture of Japan. They stay there for about half an hour, much longer than an individual would stay at the same shrine.

The family, after traveling to another nearby location or two, decides they want to end the day with a famous tourist attraction, so using the list of places in the game as reference, they exit the game and go to Google Maps to navigate to the further-away Ginkaku-ji. When they arrive, Carlie and his grandson resume playing the app before wrapping up for the day.

### 3.6.5: HC Yung

HC Yung is a Singaporean gamer, live streamer, and youtuber, who frequently reviews new games to try out for his audience. One of his twitch followers suggested a new Location based travel application for him to try, and Yung thought it would create great content as his viewers usually love Japan. He booked his flight and flew over with the full intent to play and livestream all of his gameplay in real-time. The next morning after he arrives, he begins his stream and the application. He waits a minute for the chat to decide where he should go before starting to walk in that direction. He takes his time on the way, making sure to showcase all of the aspects of Japanese culture he sees, such as vending machines and matcha green tea ice cream.

He spreads his story-gameplay out for the entire week, covering all of the areas slowly so that he can make more content. He takes longer than an average tourist at each location, even traveling to the top of Fushimi Inari. Despite not wanting to climb so far up he decided it would make excellent YouTube content. He also tries to cover as many areas as possible, while having the highest score at the minigames for each area.

### 3.6.6: HC Olde

HC Olde is in Japan for two days on a business trip. His company flew him in for a short meeting, which concluded early in the afternoon. There is still an entire day before his return flight, and doesn't have any other plans for the day. He decides he wants to explore some of the local culture, so he has good stories to tell to his family about his trip. He doesn't have time to hire a tour

guide so he Googles in search of some tourist navigation application. He comes across a POI Location game and, being in a good mood, he decides to try it for fun.

He explores two smaller shrines on the route to Kinkakuji Temple, comfortable with the level of cultural information the game provides, while allowing him to kill time before his flight. After Kinkakuji, he decides dinner is a good time to conclude his journey here

### **3.6.7: Kebab Azdul**

Kebab Azdul is a female student from Australia who is studying abroad in Kyoto for two months. She is 22 years old and knows a little bit of Japanese culture from social media, but is not particularly knowledgeable about it. She occasionally plays Pokémon Go. Two weeks after she gets settled into her dorm she realizes she doesn't know enough about Kyoto, and wants to be culturally informed about where she is staying. She catches Pokémon wherever she goes, but she decides to search the app store for a more immersive experience. She downloads Gojin and waits until the weekends to play it. Saturday morning comes and she goes out early to start her traveling. She is vaguely familiar with the area very close to the school, but hasn't explored outside of the area.

She can tell from our map which POIs are bigger or smaller, and decides to stick with just the main tourist-areas for now, making the most of her weekend time. She heads directly to Fushimi Inari, keeping the app open to learn as much as she can as she looks around. Once she has explored Fushimi-Inari and feels comfortable with the POI, she decides not to walk up the mountain, and instead to go to the next big location of interest. She spends about an hour or two per location,

wandering and observing the culture, using the game to tell her about the location but not caring as much about completion of minigames or story.

She finds the app useful for getting around and plans to show it to other students studying abroad with her.

### 3.6.8: Yosuke

Yosuke is an American-born Japanese male, who is 21 years old. He is a student at college, but now that it's summer vacation, he is visiting his family in Japan. This is not his first visit, as he goes to Japan every few years for the same reason. He has been to a number of famous tourist attractions before as a child, some of which he can remember clearly, others he cannot.

He knows some basic history about a select few places that he has gone to, however because these places feel more comfortable and “normal” to him, he has not bothered to look deeper into the details of these places.

After a week of living in Kyoto with his grandmother, his friends in America mention that there is a game that can only be played in Kyoto, and the details thereof. Yosuke decides that it sounds interesting enough to try and play. Once he loads up the app, he sees the list of POIs. Being already familiar with the area, and not particularly motivated to travel far for this game, he decides to visit the closest shrine, simply down the street, without looking at the map

He starts getting interested in the story side of the game, as well as the badge /collection aspect of the game, as each badge to him feels like a digital good-luck charm. He visits the other shrines within 15 minutes from his grandmother's house before returning. Yosuke does not play

further in the next week, yet keeps the app on his phone to preserve the good luck charms, and in the case that he travels somewhere.

## 3.7: Experience Description

### 3.7.1: Generalization of Audience

Our target audience, broadly, will be English-speaking visitors in Kyoto. They will usually be 18-40 year olds, because that age range makes up a large part of the gaming population [X], but people of other ages may play as well. All ethnicities are part of our target audience, it does not affect the gameplay or cultural experience the application tries to deliver to them. We expect tourists to have low or moderate cultural knowledge, because if they are already familiar with Japan they have much less motivation to use our application. We expect that they have some experience with mobile games, though, but it is not required. Mobile gamers are the most likely to find our game, because they are the ones most likely to search for this genre in the app stores. People with physical disabilities are not included in our target audience, as our game requires a player to perform the physically-intensive task of exploring all over Kyoto. Our audience will most likely be traveling alone, due to the game being single-player, as well as requiring greater navigation aids if you are traveling by yourself.

### 3.7.2: Our intended experience

The intended experience we wanted to provide our audience is one of exploration, learning, immersion and enjoyment. When our audience arrives in Kyoto, they presumably want to explore, either to play or to learn about the culture. The player will be guided to a POI, then instructed what

to do there to follow cultural etiquette. They will also learn some history about the POI, or what it is known for. They are incentivized to look around and explore the various aspects of each POI. They then move to another POI once they feel they are finished with the one they are on.

### **3.8: Paper Prototypes**

We created a paper prototype for our game, hoping to somewhat closely resemble actual gameplay, in order to get the best data possible. The game consisted of maps for each floor of the Ritsumeikan building our lab was in, with specific locations highlighted. These locations correspond to the real-world points of interest, that open a minigame when “clicked.”

#### **3.8.1: Map**

As stated above, we have maps for each floor that have locations highlighted. The player switches maps depending on which floor they are on. They physically move between floors to mimic real gameplay. Once they walk to a location, they can choose (verbally) to select the POI marker, which will allow them to enter a minigame. The minigame depends on how many “!” is above the POI, 3 being the hardest.

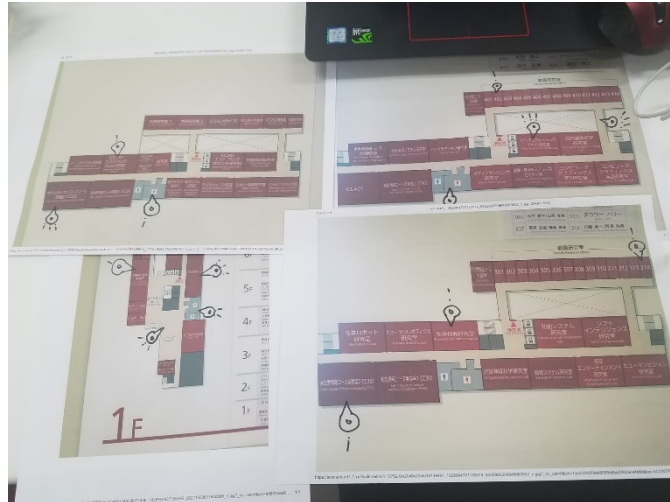


Figure 3.15: Map Paper Prototype

### 3.8.2: Fushimi Inari Catch the Mouse

The board is the 5 patches of grass. The game is supposed to be similar to Whack-A-Mole, so we place either a mouse card or a farmer card on a patch of grass, and the player will react. If the player cuts a mouse 3 times they win. If the player cuts a farmer they lose. Also shown is the badge for winning, which was given at every minigame for simplicity.

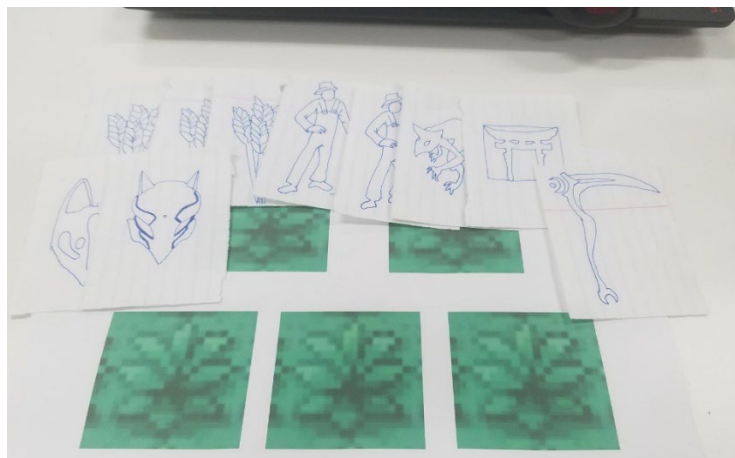


Figure 3.16: Display of Badges and Minigame



### 3.8.3: Ryoanji Symbol Writing

We gave the player a piece of paper and a writing utensil, and showed cards, one by one, for each stroke they are to copy.

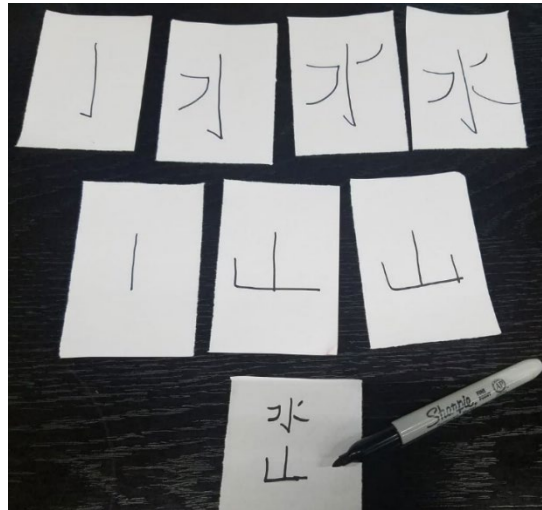


Figure 3.17: Sample Strokes for Ryoanji Line Trace

### 3.8.4: Ball Maze

This minigame is a ball maze, with indents on the cardboard, allowing the metal ball to roll from the start to finish, indicated by the board.



Figure 3.18: Cardboard Model of the Ball Maze

### 3.8.5: Ginkaku-ji Unblock the Moon

This game is a slide puzzle, where you create a straight line from the moon to the temple by moving pieces vertically or horizontally.



Figure 3.19: Sample Slide Puzzle

### 3.8.6: Myoshinji Temple Jigsaw

This is a simple jigsaw with unit-based pieces.



Figure 3.20: Sample Paper Jigsaw

### 3.9: Iterative Design

Over the course of the design process there were a number of changes made to the minigames. Initially, Nijo Castle Tower Defense was going to have multiple types of enemies, but due to time constraints was simplified down to have a single enemy. Similarly, the Yasaka Shrine/Gion District minigame was going to feature multiple tracks that are randomized upon load, but this was cut due to time constraints and the need to keep the app reasonably sized for easier download. Ryoanji Line Trace was originally intended to be tracing the strokes into a sand Zen garden, but due to difficulty in determining an efficient way to create the effect of tracing lines in the sand we changed it to be following calligraphy strokes on a scroll. Finally, Myoshinji Jigsaw Puzzle was originally going to be a one layered jigsaw puzzle, but to add some complexity and difficulty we implemented a two-layered puzzle system. The remaining two minigames had little to no changes made from the original design concepts.

The map component also went through a few iterations. For our first iteration of the map, it was more simplistic with minimalistic buildings, the player character, the tappable prefabs, and not much else. After that, we wanted to add more usability features and more details on the map to enhance user navigation. We started iterating by adding street name labels, as well as POI categorical icons such as a park icon over every park. We then added a small guidance system, a running animation for the character when the location changes, and changing the color scheme of the map to match the theme.

## Chapter 4: Technical Implementation

This chapter is divided into three major sections: the menu and unlock system, the implementation of the map system, and the implementation of the minigames. The menu and unlock system section talk about how we control and persist player progress through the game. The Map section details the Mapbox SDK [25], what we did with it, how we move the camera, and how the map loads the minigames. The minigame section covers the mechanics of how each minigame works in detail.

The game was initially designed on laptops for the aspect ratio 16:9, as that was the screen ratio of the phones we had available to test on. The two phones we used for testing were a Samsung Galaxy S7, and a Samsung Galaxy S7 Edge. Although Unity provides the Unity Remote app which can allow developers to test their apps on phones plugged into their computer, there are issues with screen resolution which led the team only testing full builds.

### 4.1: Main Menu & Unlock System

The unlock system stores a table of each game and their current status, as shown in Table 4.1. When the user begins the game, all of the minigames are set to locked. When the icon on the map is initially pressed the player is brought to a cutscene, and the Persistent UI will set the status of that game to “cutscene” and maintain that status. When the player then goes and enters the minigame from the cutscene, and finishes the minigame, the status is updated to “minigame” and “completed” respectively, using the same method.

This table of minigames and their statuses provides decision-making information for other parts of the game. When the player loads the minigame again from either the map or the menu, the game will load them to the correct stage in their progression: cutscene, minigame with secondary cutscene locked, or minigame with no secondary cutscene.

Furthermore, the main menu is heavily controlled by this table. If a game’s status is “locked,” the main menu locks the access button for that specific minigame. If a game’s status is “completed,” the main menu displays the badge for that location.

Status Name	Condition
Locked	The game has not been located on the map, and is unreachable from the main menu.
Cutscene	The game has been located on the map, but the player has not completed the initial cutscene. When loading from the map or main menu, the cutscene associated with the minigame loads first.
Minigame	The player has completed the initial cutscene and now is taken directly to the game from the main menu or map.
Completed	The player has achieved a winning condition and unlocks a final cutscene and a badge on the main menu. The game loads directly to the minigame, and a final cutscene will not be played again after the first viewing.

*Table 4.1: Minigame Statuses*

### 4.1.1: Persistent Data

To allow for more replayability and to account for the possibility of players closing the app, we implemented a save and load system to store the player's progress. The stored data holds information regarding each minigame including their status (Table 4.1). The persistent data also holds if the player has dismissed the tutorial screen for each game and conditionally loads the tutorial.

The player's data is stored into a file named 'gameInfo.dat' using a binary formatter, so that the player cannot edit their personal game data. The information is formatted using serialized classes, which allows the engine to store data and write to this file format. The status of the minigames are

```
[Serializable]
class PersistentData {
    SpecificData dataToStore;
}
[Serializable]
class SpecificData{
    string aName;
    public SpecificData(string aName){
        this.aName = aName;
    }
}
```

held in their own serialized class to allow for the storage of more complex objects.

Players can also completely erase all of their data and start the game anew by deleting this file.

The GameController class is where all of the persistent data is held, and therefore follows the singleton design pattern. This object informs all other classes about the status of the game, and handles the file input and output. Another part of the persistence system is the persistent UI is

injected into every scene of the game. This UI system serves to allow navigation to the main menu and the map from every game and cutscene.

## 4.2: Log System

To be able to understand more information about our players there was a need to create an additional log system. The standard practice for debugging android apps is to have them plugged into a computer to be able to read the log messages, but as a main feature of our game is to have people explore Kyoto, we needed another way to access this data. To do this we implemented a system that writes to a local text file stored on the device, and then creates a button to email us the log when the player finished testing the game. The log system attaches the time string to each message so that we could analyze how much time players spent on each game, and how many locations the player actually traveled to. The contents of the text file are then copied to the body of the email, and sent to us.

## 4.3: Cutscenes

Cutscenes were implemented before and after each minigame in order to maintain cultural relevance and give the user some logical progression through our game. They, each feature a god and evil character, inspired by the culture of the location.

### 4.3.1: Dialogue

Every cutscene has a dialogue box they need to go through to access the minigame. This dialogue starts when the user presses the “Start Conversation” button. When this button is clicked, it takes an already inputted list of sentences and stores them in a list. The first sentence will display the sentence

smoothly, letter by letter until it is complete, and waits for the user. If “Continue” is pressed, it will go to the next sentence in the list of sentences.

These sentences are stored in the toggleable log. When the user is finished with all the text, they will see a button that allows them to exit the cutscene to either the main menu or the minigame, depending.

## 4.4: Map

### 4.4.1: Mapbox SDK

Mapbox is a live location platform that provides real-world map data, as well as tools to help implement it into various applications. We felt it was the best choice because of its ease of use and the customizability it provides. Using this SDK, we can create a high-quality result, while saving a significant amount of time creating a similar service that Mapbox already provides. This time was used to create other aspects of our game. Mapbox provides high quality data, customizable aspects of the map, as well as an SDK to assist integrating it into our project.

Furthermore, pricing is not a concern as Mapbox is completely free to use up to 50,000 active monthly users. We do not expect to have anywhere near that many users for the scope of the MQP.

The Mapbox SDK comes with many built-in features that are perfect for our game. All of Mapbox’s map data is stored as GeoJSON which Mapbox can interpret and display in various ways. Some examples of what Mapbox can do are sorting by type of building, create basic 3d models of buildings, and place objects at specific locations.



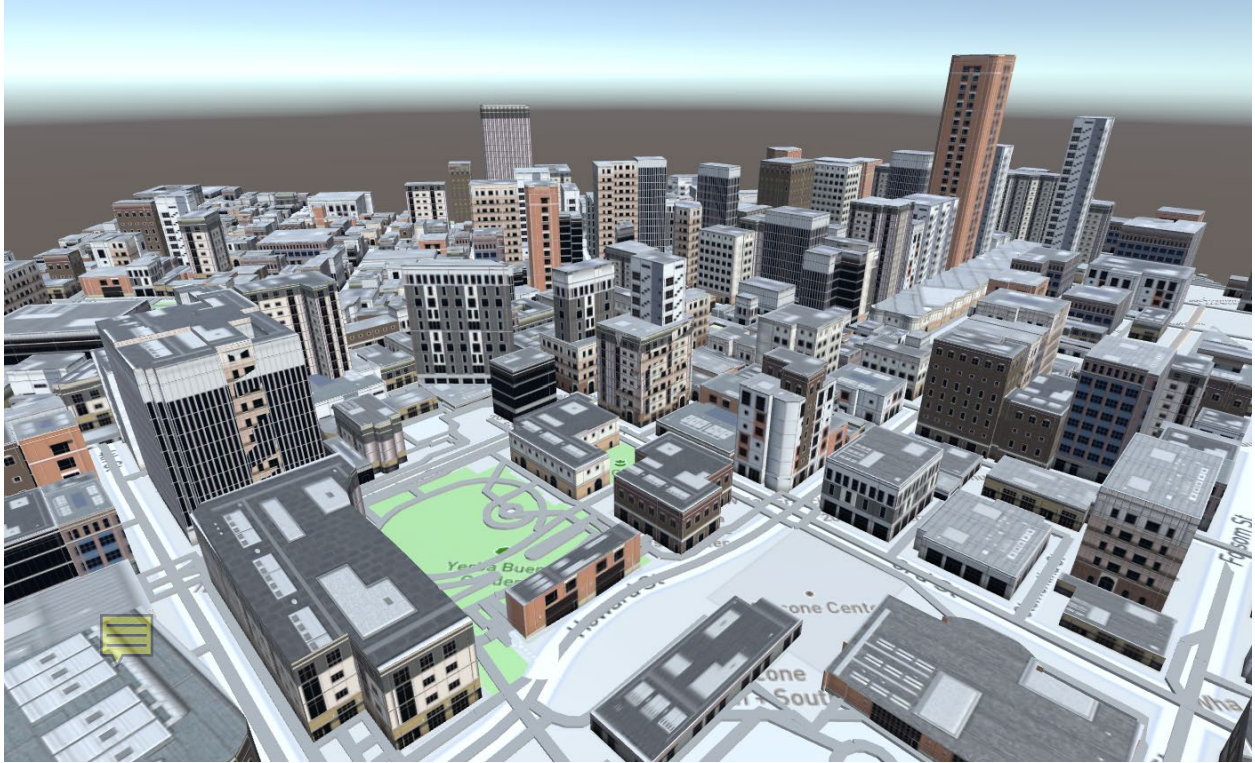


Figure 4.1: Rendered 3d Buildings

We used these features to spawn prefabs at the places that we have minigames for, and spawning small icons above each restaurant, shop, park, service, recreation center, and transportation to show which of these the POI's category is. These icons can also display the names of the POI, taken from the GeoJSON data sent by Mapbox, however we opted to hide all the names because the vast majority of the names were Japanese-only and could confuse a player.

#### 4.4.2: Camera Movement on Map

The camera system on the map is modeled off of the camera system in *Pokémon Go* [33]. The *Pokémon Go* system has both the ability to move around the player smoothly when the player drags his finger across the screen, as well as a two finger drag to scale the camera distance from everything else.

This camera orbit is done by parenting the camera to the player, and having a look at function to always face the player, so it will always follow the player at a set distance. We have an orbit script to rotate the camera around the player based on how they swipe on the phone. This script checks the change in the X and Y axis and the side of the screen they are on respective to each axis, then applies the correct rotation based on this. For example, if the player started directly below the player and drew a quarter circle counterclockwise around the player, they would be directly right of the player. The camera would recognize a leftward shift on the bottom half of the screen, and an upward shift on the right side of the screen, both of which tell the camera to move counterclockwise around the player as shown in Figure 4.2. This creates the effect of a camera just like *Pokémon Go*, where you can draw circles around the player to have the camera spin following the finger.

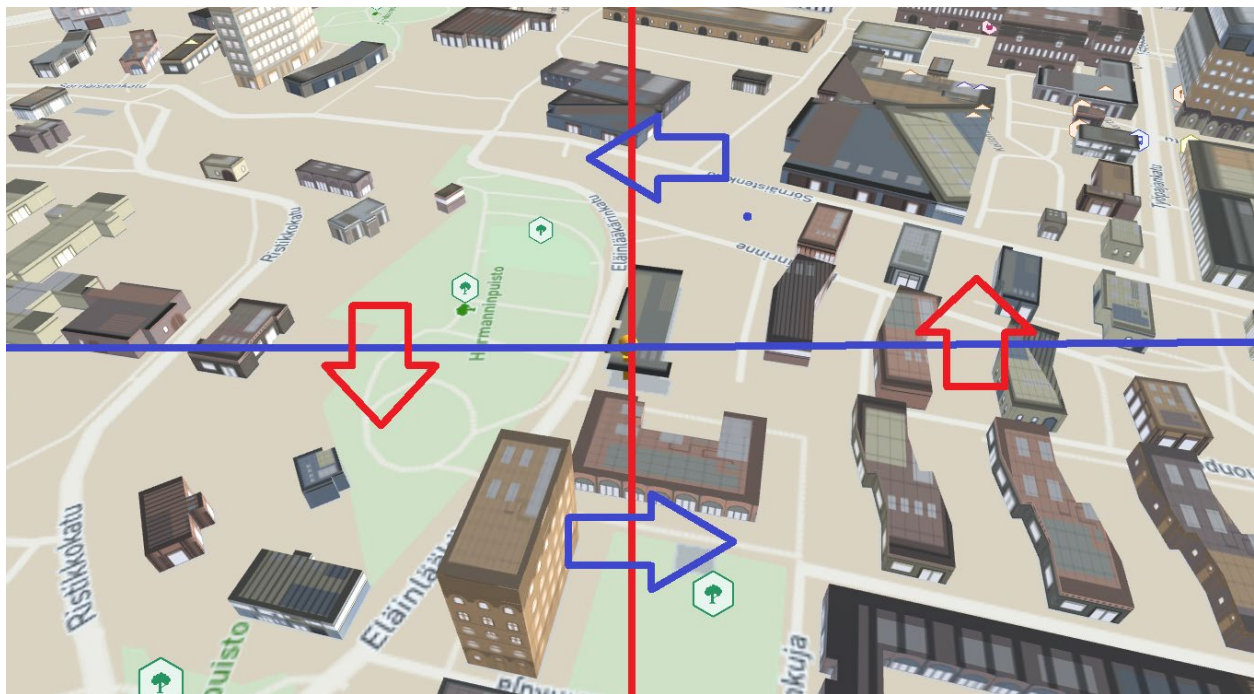
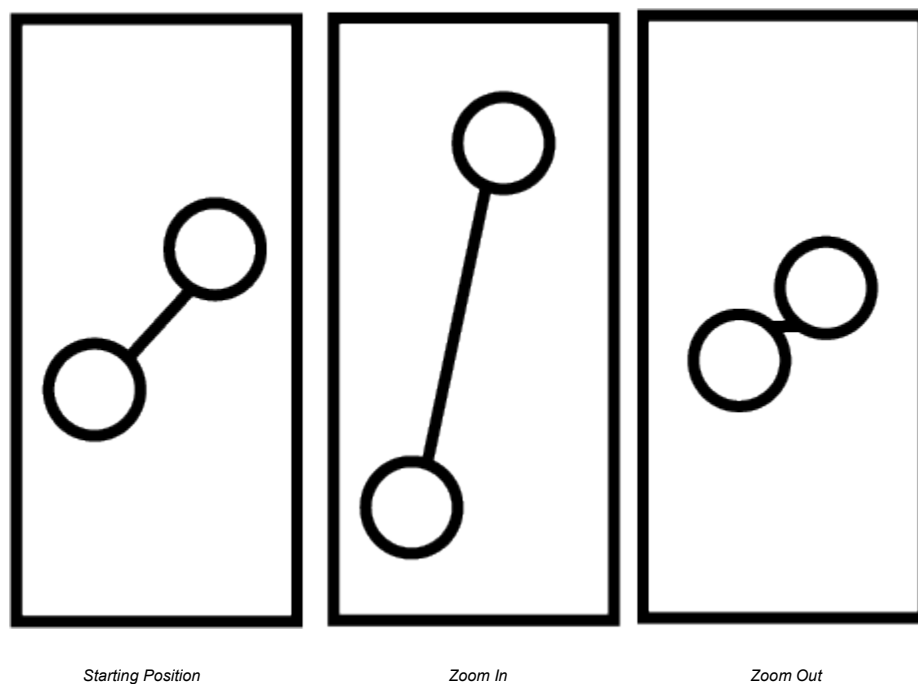


Figure 4.2: Camera Movement Diagram

The zoom script is used when there are two touches on the screen. It looks at where the two fingers started touching the screen and where they are at any point in time, finds the change in distance between the touches, and scales the camera distance to the player accordingly. If the player presses two points and spreads the two fingers away from each other, the camera distance from the player will decrease, and if the player moves his fingers closer together the distance from the player will increase, up to a set maximum distance.



*Figure 4.3: Pinch to Zoom Diagram*

These two scripts are able to work together, so you can zoom and rotate at the same time.

#### 4.4.3: How Maps Interface with Minigames

The map filters out and highlight POIs that the player may choose to visit on their trip. Depending on the exact location, they are emphasized using one of two methods. Either they will be highlighted by filter (for example, a bonus for visiting any “park”), or they will be highlighted by a

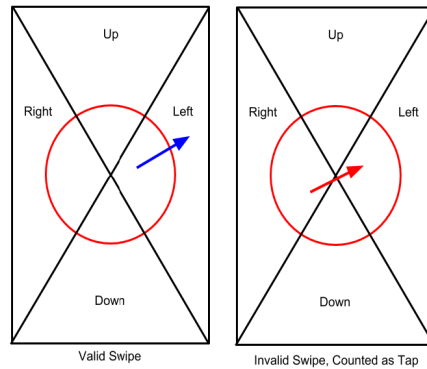
unique placement (for example, we created a specific object to represent Fushimi-Inari Taisha, that when pressed, opens up the story route for Fushimi-Inari Taisha along with the minigame. Both of these options are provided from the Mapbox SDK. Mapbox comes with scripts to either filter by a specific type of location, or place an object at a specified Latitude and Longitude. For our most important locations, we inserted the object manually using this method. When a game object is pressed, a script changes scenes to the minigame's scene.

## 4.5: Minigames

The minigames are what we spent most of our time developing, as they each have their own technical elements and design choices. In this section we detail the mechanics used in creating each of the minigames our game features.

### 4.5.1: Fushimi Inari Run

The main element of the game is the touch gesture system. Touch gestures are to move the player character across the lanes and to dodge obstacles that are above, in front of, and below the player. The system that checks to see if the player is swiping measures the pixels between system registered touches, and then determines the direction of the swipe using simple subtraction. The phone area is divided into four major sections (Figure 4.4) with a circle to denote a threshold that the player must drag across to have the input be considered a swipe. This threshold is to ensure that the player mean to swipe rather than just to tap. The swipe direction is also determined by difference in the x and y coordinates if the start and end locations of the drag.



*Figure 4.4: Swipe System*

After processing the player's input, the game decides what animation to play using a system of checks in the Unity Animator system. If the player swipes up, the game plays the jumping animation which actively moves the player character's collision box to prevent collisions with the obstacles. If the player swipes down, a slide animation plays with the same impact of moving the collision box with the model. The game also checks for the player's status: alive or dead, and uses that to decide to play the dying animation upon collision with an obstacle.

The obstacles and tokens spawn based on how far the player has progressed, making the game more difficult as time passes. Once the threshold of 10 platforms has been passed, the maximum number of obstacles and tokens are clamped to prevent an unwinnable game situation where all of the tracks have unavoidable obstacles. The following pseudocode breaks down how the spawning system chooses to create the objects.

```

private void GenerateObstacle():
    if player platform number > 10 than
        set max number of spawns to 11
    Generate random number from 1 to max number of spawns (inclusive);
    Calculate max and min distance to spawn objects between and set as valid range
    for generated number of spawns
        k = Generate random coordinate in valid range
        if k is not in list to spawn
            Add k to list to spawn
    for all objects in list of objects to spawn
        switch height component of coordinate
            case == 0
                Generate Stone at coordinate
            case == 1
                Generate Barrier at coordinate
            case == 2
                Generate Spider at coordinate

```

*Code 4.2: Generation of Objects in Fushimi Inari Run*

The type of obstacle that spawns are chosen based on a randomized height variable, and the horizontal track locations are chosen randomly from the three possible track options. The vertical distances are ingrained into the type of the obstacle itself to keep consistency.

#### 4.5.2: Nijo Castle Tower Defense

The enemy characters navigate the path through a system of waypoints. There is one master waypoint controller that holds the location information of all of the waypoints on the map. The enemy characters themselves navigate across the waypoints by storing the location of the next target

waypoint and move in a straight line towards that point. Upon reaching the final waypoint the enemy characters self-destruct.

The tower placement system uses buttons and touch input to place towers on the map. The player taps the button of the tower they wish to generate and then that tower appears on the map near the button. Then the game tracks the player's input to move the tower across the map/board, and using a system of triggers stops the player from placing the tower too close to or on the path, off of the play area, or overlapping over an existing tower. The player places the tower by double tapping on where they wish to place the tower provided it is a valid location. The validity of the potential location is shown by a semi-opaque circle around the tower that changes from green to red to indicate an invalid placement.

There are three types of towers the player can place: a radial shooter, a tower that targets the nearest enemy, and a tower that slows down the enemies. The first tower blindly fires projectiles radially around the tower at a set rate. The second tower tracks the closest enemy within its range and then fires at a standard rate at that enemy. It checks every frame to continually find the closest enemy to it. The third tower halves the speed of the single closest enemy within its range once every set period of time. The cost of each tower is 10 units, and the player starts with a budget of 100. Money is earned by killing enemies at a rate of 5 units per enemy. The cost of upgrading the tower is 10 units for the first upgrade and 20 units for the second upgrade. Each tower placed can be upgraded up to two times. This is achieved by the player tapping the tower, turning the radius blue, and then tapping the upgrade button. If the player exceeds the maximum amount of towers they can

afford to upgrade, the upgrade button vanishes until they deselect any number of towers to lower the total price into an affordable range.

The difficulty of the waves of enemies grows exponentially with the wave number. For the number of enemies per wave the function:  $NumberOfEnemies = \lfloor e^{(CurrentWaveNumber + 1)} \rfloor$  is used. The speed of the enemies' navigation increases with a similar exponential function:  $SpeedOfEnemies = \lfloor e^{1/(CurrentWaveNumber + 1)} \rfloor$ . The last variable enemy characteristic is the time between enemy spawns, which follows the same formula as the speed of the enemies.

#### 4.5.3: Yasaka Shrine / Gion District Rhythm Game

The game parses an XML file that holds all of the note data for a song, using the System.XML library. Each note has child nodes that tell the game where to spawn the note and at what time. The location data is stored as a single character: 'L', 'C', or 'R'. From this information the note is spawned at the spawner location at the top of each track. The time data is a float used to store when the note should be spawned relative to the beginning of the song.

After the song data has been parsed from the file, the game waits for the player to hit either the easy or hard start button. From that point the game plays the audio file and then begins to generate the notes. The notes are given an assigned velocity so that they move at the exact rate of the BPM of the song. When the notes collide with the catchers, a boolean is set to true to indicate that if the player touches the button now it will give them a point. The scoring system is triggered by the tapping of the button beneath the respective catcher. The combo tracking system stores the number of notes a player has successfully scored and then checks if the current note they scored on is directly



after the previous note; if not, or if the player hits the button beneath the catcher without a note too early or late, the combo is set to 0.

#### 4.5.4: Ryoanji Line Trace

The main technical element of this game is how the player's stroke input is processed. As mentioned in Chapter 2 the player is shown a character stroke by stroke, and then needs to mimic each stroke as they appear until they reach a set accuracy. To process the input, we decided to use the Gesture Recognizer scripts available to the Unity community created by Mads Møller [24], specifically the C# version of the code provided by Gabriel Cirera. The main reason we decided to use these scripts was because it provided an easy way to add our desired strokes and score the accuracy of the user. The player stroke data is stored in an array of 64 Vector2D points that can be scaled and rotated as needed to calculate the accuracy. The score is calculated in a two-step process. The first step is converting the pixel location coordinates on the screen to coordinates based on the world. The second step is comparing the converted points to the existing library of template gestures point by point to find the most likely match.

The player is first shown the stroke they must attempt using an animated brush with a trail behind it to show the stroke. After the player is shown the stroke, they must draw the stroke on the screen until they get an accuracy of at least 75% for the game to recognize the stroke. This accuracy was chosen because of the limits of the gesture recognizer scripts and the processing power of smartphones, as a higher stroke accuracy requirement would need many more comparisons would have to be made to determine the score. Every time the player places their finger to the screen and then removes it is counted as a stroke and checked for correctness. At any time, the player can

choose to view the sample stroke again by selecting the button “View Stroke”. The player’s input is drawn to the screen using the built in Unity component Trail Renderer/Line Renderer, which generates a trail along the path of the object’s motion. Once the stroke is completed the game redraws the stroke onto the scroll on the side of the screen, clears the screen of all remaining attempts and draws the next sample stroke.



*Figure 4.5: Left vs Right, Valid vs Invalid Stroke*

#### 4.5.5: Ginkaku-ji Unblock the Moon

The main technical element of this game is simply moving pieces vertically or horizontally. The technical element of moving a piece is initially very straightforward, as it is a simple script that moves a selected piece to where the mouse points. There is a trigger object at the exit, which will run the end game animation once the player has won. Only the “main piece” is allowed to go through the exit.

The issue lies in the fact that there are many edge cases that allow unexpected behavior which had to be fixed. The Unity colliders and rigidbodies have limitations, in that sometimes if a piece travels too quickly, it is able to bypass other objects entirely before collisions are checked (even at slow speeds and maximum timesteps). A piece will end up either inside another piece, past the piece,

or inside a wall before it notices a collision has occurred. To remedy this, we firmly restricted the piece movement to within the confines of the 6x6 grid, then made it so if a piece was in a collision (if it ended up inside of another piece) it would back up slowly until it is no longer in a collision. Sometimes this also did not work because the piece would jump over 50% of the way through, and be pushed out the other side by the same logic, so we restricted the movement speed to a maximum of <1 unit per frame.

#### 4.5.6: Myoshinji Temple Jigsaw

The core mechanic here is a jigsaw fitting mechanism. There are pieces and holes; each piece has one corresponding hole, either in another piece, or on the board. The ground pieces have holes on the board, and the temple pieces have holes on the ground pieces. The holes use triggers to tell if the correct corresponding piece is directly above it. If the piece is released above a trigger, the piece will be pressed onto the board, and unable to move. Similarly, if a piece is released anywhere else, even if it passed through the trigger box, it will remain active on the board. Once the last piece is locked onto the board, and all pieces are disabled, the user will win.

The pieces are assembled in multiple layers as well, in order to handle multi-slot pieces, such as a ground piece with two temples on top. There is a main piece container that handles collisions, and a collider to represent the hole, if applicable. The art is attached but separate, due to being centered at the origin.

## Chapter 5: Art

### 5.1: Intro to Low Poly Art: Background and Inspiration

Many reasons lead to the final decision of using low poly style for our game. First, we made a game that consisted of several minigames, so we needed large number of different art assets for different minigames. They also may not be reusable due to the different theming of each minigame, so having art assets that are easy to create is beneficial. For example, Fushimi Inari Run needed many elements such as torii, obstacles, and temples, while Nijo Castle Tower Defense needed an entirely different set of assets. Creating low poly objects was significantly easier than modeling detailed realistic objects. Low poly models were easier to model, texture, rig and animate. This style allowed us to create a higher volume of simpler models that were required to appropriately theme every minigame. Second, we can keep consistency throughout different games by implementing objects with the same art style. It was very important to hold a consistent art style so that the game felt like one cohesive experience. We were able to bind differently themed minigames together through this uniformed low poly style. We aimed to create a relaxing, yet serious atmosphere for travelers to learn about the places they visit. Thus, the style fits the purpose of our game.

#### 5.1.1: Definition of Low Poly Art Style

Low poly art style is a 3D art style where the number of polygons of 3D objects are lower than more realistic 3D objects [11]. The goal of this art style is to abstract the design of the object but still allow for easy recognition. Designer Kaitlyn Ellison described the low poly style as “... a

bunch of similarly colored triangles or rectangles combining to create an intricate geometric shape, which somehow manages to be abstract but still demonstrates a clear figure of some sort...” [20].

As stated by Josh O’Caoimh, an artist who has worked on assets for video games, low poly art has a few major advantages due to its simplicity. First, the models take up less space. This helps to save resources so the game can run on lower-end smartphones. Second, low poly objects are easier to model, texture, rig and animate. The production time of low poly models is significantly shorter than the time it takes to create more realistic models. Third, games using low poly art style often age slower than the games trying to achieve realism, since realistic models improve faster due to hardware and software improvements [19].

### 5.1.2: Inspirations from other games

The art of this game was set to be light and relaxing. Since we made a game that consists of different minigames, we felt the game should create a relaxing atmosphere for the players to better enjoy the game. Sections of the game are set in shrines and temples in Kyoto area so, the art of the game needed to reflect Japanese culture. A wide variety of games inspired the artistic design of Gojin, but the following three were the most impactful.

*Shelter* [31] is a game produced by Might and Delight in 2013. As described on the official website of Might and Delight, *Shelter* is a “tamagochi-like game that takes you back to the forest of Shelter every day... [31]”. In the game, the player is a mother badger who is trying to protect her cubs. The art style of the game is low poly and papercut, where the animals and environments in the

game look like handmade paper models. The designer of the games used this style to create a relaxing atmosphere for players in the game to better socialize with others.



Figure 5.1: Screenshot of Shelter [30]

Another source of inspiration was *Epistory - Typing Chronicles* [7]. The art style of the game is originally based on origami, a traditional Japanese paper folding art; the characters and environmental objects are all low poly. The environmental design of the trees and rocks served as a major source of inspiration.



Figure 5.2: Screenshot of Epistory - Typing Chronicles [6]

Gojin also draws inspiration from *Ōkami* [3], an adventure game about Japanese gods, produced by Capcom on 2006. The way it depicted Japanese gods is quite different from the other Japanese games at that time. The goddess of the sun, Amaterasu, was depicted as a white wolf. The art style of the game is low poly style mixed with a traditional ukiyoe style. Although the main reason of using low poly style is the limitation of the capacity of PS2, it still succeeds in conveying the relaxing and light atmosphere through the art style.

These games created a sense of admiration and acknowledgement of the charm of a low poly art style among the team.

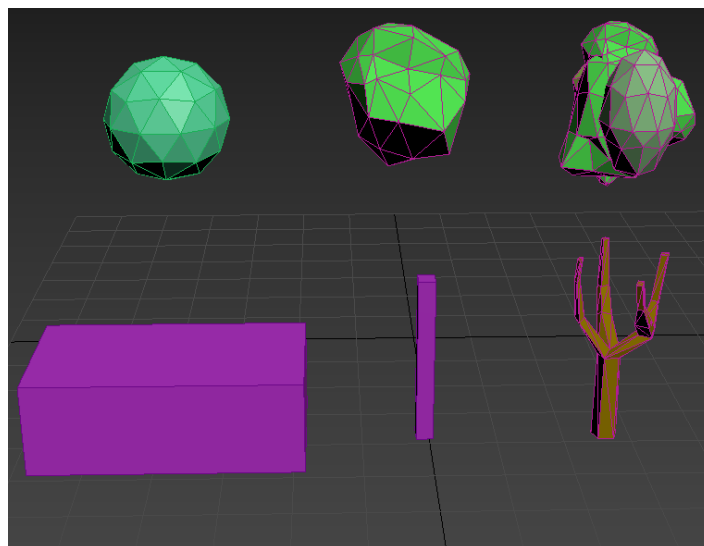
## 5.2: Minigame Assets

There were two major themes when designing environmental objects for the game: the emphasis on relaxing atmosphere, and an accurate reflection of the real-world locations. To achieve these two themes, each object created was designed referencing multiple pictures. Additionally, an effort was made to personally visit each location to allow for a more contextual understanding of the objects.

### 5.2.1: Production Techniques

The procedure for creating the models was divided into two major categories. For simple environmental objects, 3ds Max was used to create the general shape and work from extremely low poly to slightly higher poly. For complex shapes, ZBrush was used to model a high poly version of the model, and then the number of polygons was reduced using various techniques.

As for simple shapes like mountains, trees and buildings, it was easier to start with simple shapes and modify the vertices directly. For example, we have four different kinds of trees. The initial step was to scale a box to look like a basic tree trunk. Then the simple trunk was modified by: adding extra vertices to the surface using swift loop tool or cut tool, extruding certain surfaces to create branches, adding more vertices, and then modifying the vertices to make the trunk looks more organic. For the crown, the vertices of a simple geosphere were dragged to break the rounded surface; the sphere was then cloned multiple times, modified, and then assembled. The final step was just simply combining the two parts together.



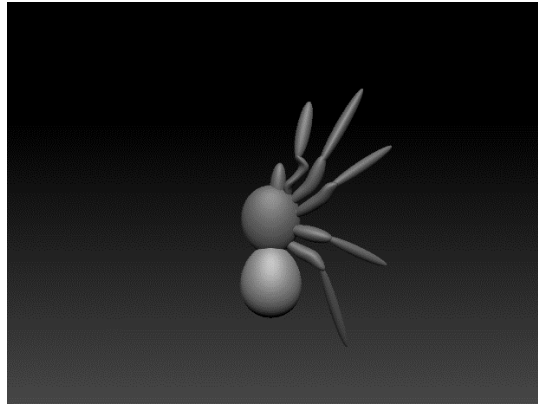
*Figure 5.3: Diagram of Tree Asset Creation*

*From top left to top right: Creation of crown. From bottom left to bottom right: Creation of trunk and branches*

For complex shapes such as the spider, ZBrush was used to model a high poly version of the object. The basic shape was generated using ZSpheres. This was used to have a better control over the proportions and overall shape of the spider. After the basic shape is created, the move brush was used to adjust the legs of the spider. After finishing the high poly model, decimation master and ZRemesher were used to decrease the polycount of the model until the contoured surfaces looks



polygonal. Some of the higher poly models were imported to MeshLab and underwent further polygon decimation. After the model was reduced to the desired low poly appearance, it was imported into 3ds Max for any detailed modification needed before it was ready to be in the game.



*Figure 5.4: Half of the Spider Model in ZBrush*

As one huge adventure of low poly art style, texturing the objects was not a huge challenge. However, to make the models accurate to the real-world original objects, it was important to assign an appropriate color scheme to the them. The strategy used to achieve the desired color scheme was to find reference photos of the object, put the picture into a color picking website, extract the color information of certain area, and input the color code into the ambient color slot of certain materials. Sometimes adjustments were needed for other parameters of the material such as specular level and glossiness, to make a metallic or a leather material. This strategy helped to better reproduce accurate colors and make the models more believable.

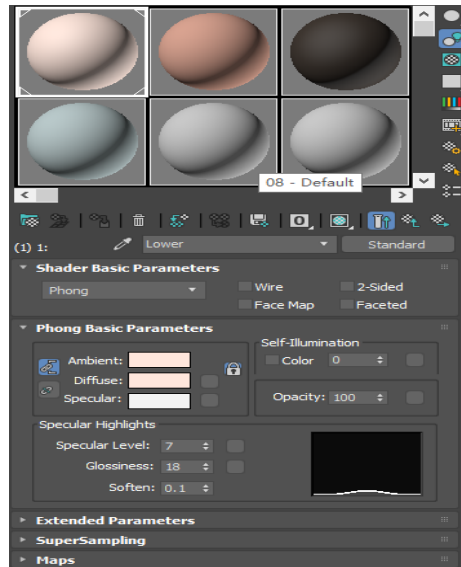


Figure 5.5: The Material Panel for the Shamisen

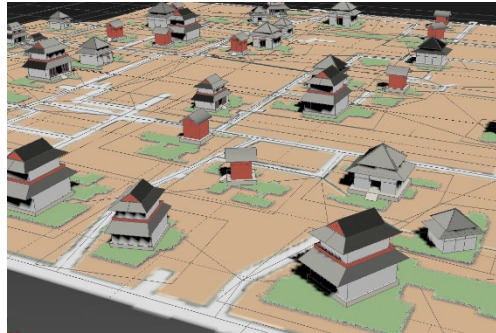
## 5.2.2: Fushimi Inari Run

In Fushimi Inari Run, we needed to create various obstacles for the player to avoid. When talking about dangerous objects hanging from above, one probably immediately thinks of spiders. We had seen a lot of spiders hanging in everywhere in Japan, so we decided to use giant spider as the obstacle that the player must dodge in the game. Also, when visiting Fushimi Inari, there had been spider webs clinging to some of the torii gates. The item the player aims to collect is an onigiri. This is based off of the shops that sell food along the path to the top of the shrine. To create a more detailed and immersive environment, basic models, such as trees, rocks, and mountains, were designed and added to the scene. The generic appearance of these models allowed them to be reused into other aspects of the game.

## 5.2.3: Myoshinji Temple Jigsaw

For other minigames like the Myoshinji jigsaw puzzle, the objects more closely resemble the real buildings in the temple. After extensive research, four basic buildings were modeled to represent

four of the most easily distinguished temples. Even though there are only four variations of temples, when we reuse these assets multiple times in the scene they do not look repetitive due to the differing placement angles.



*Figure 5.6: The Jigsaw Puzzle Scene in 3ds Max*

#### 5.2.4: Nijo Castle Tower Defense

Nijo Castle Tower Defense used imaginary elements more than the actual elements of the place, due to the mechanics of the towers. For the three different towers, the models were created to give a visual cue as to their behavior. The tower models consist of a crossbow, cannon, and magic book, and each tower has three different levels. To address the leveling mechanic, each tower has different models for each level. For example, the level one arrow tower is just a huge crossbow with a base. When it is leveled up to level two, the shape of the crossbow becomes more complicated and there is a basket of arrows next to the bow itself. And leveling the tower to the third level brings the crossbow into a pavilion. The pavilion is designed based on the structure and color scheme of the buildings in Nijo Castle.

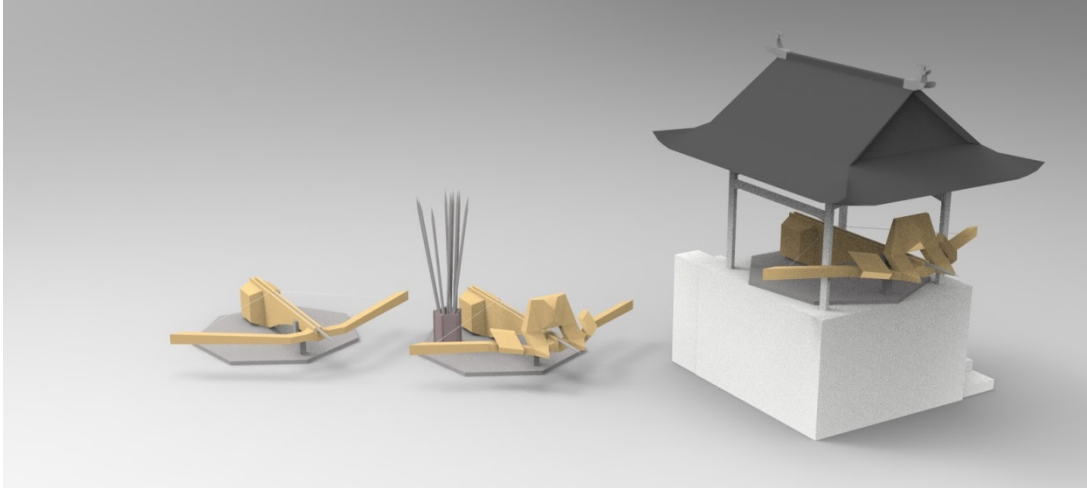
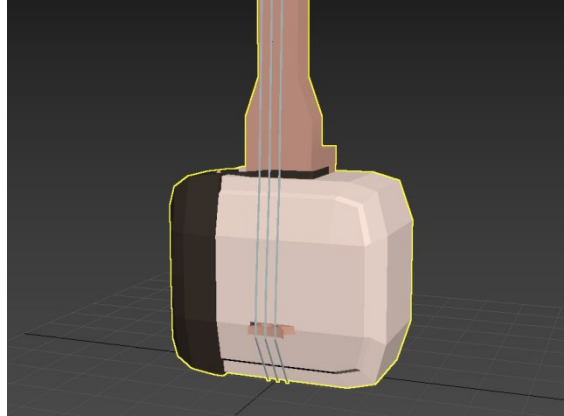


Figure 5.7: Arrow Tower Levels from Left to Right: 1, 2, 3

The enemy that the player aims to defend themselves from is a samurai. The samurai was modeled from the costume that people wear when performing Kendo, mixed with the cloth of the main character in *Gintama* [45].

### 5.2.5: Yasaka Shrine / Gion District Rhythm Game

A standard rhythm games needs a least 3 elements, the notes, the tracks for the notes, and the catcher. In order to maintain a consistent theme for these objects and incorporate Japanese elements in the game, we decided to set the rhythm game along a shamisen, a traditional Japanese instrument used by Geishas [9]. The body of the instrument was used as the background for the game, and the strings of shamisen were used as the tracks of the notes.



*Figure 5.8: Model of the Shamisen*

For the notes, no simple unified Japanese representation of music notes existed, so we took inspiration from the elements in Yasaka shrine and the surrounding Gion district. The note was modeled after the type of lantern that is widely used as decoration when Gion Matsuri is happening.

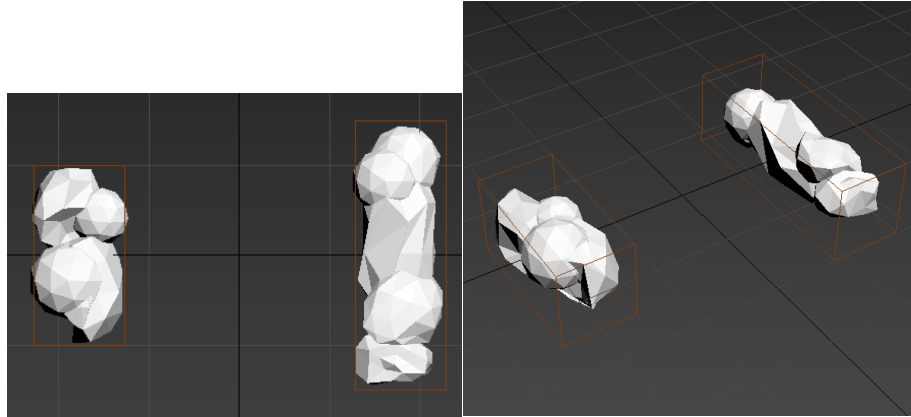


*Figure 5.9: The Lantern Model for the Note*

## 5.2.6: Ginkaku-ji Unblock the Moon

The Ginkaku-ji Unblock the Moon was themed around moonlight and night time. This theme was used to create an illusion of peace when the players are in this minigame. A peaceful night sky was needed as the backdrop, so two kinds of clouds were created for the puzzle pieces and a moon for the surrounding environment. Because it is a sliding puzzle, the puzzle pieces needed to be designed to specific size constraints of 1x2 and 1x3. To achieve this, two or three boxes of 1x1

standard size were merged together and then modified to resemble clouds. This process allowed for



*Figure 5.10: The Cloud Models*

the creation of models that are in the shape of clouds and in standard sizes.

## 5.3: Characters

### 5.3.1: Production of Characters

Since we have many humanoid characters, two basic models were made, one male and one female, for the characters. By doing so, the model could be reused for the creation of every character. For the humanoid characters, the first step was to start with the high poly version of the basic model and reduce the number of polygons using ZBrush, therefore removing levels of detail from the model. Then, the character is duplicated with face area masked, and the remaining polygons are deleted. Next, working specifically with the isolated face model, the move brush was used to sculpt the model into a mask. After that, the poly build brush and move brush were used to sculpt the mask. When the general shaping of the mask was completed, the dam standard brush was used to carve lines on the mask.

For the clothing and accessories of the characters, 3ds Max was preferred, because it works better for hard surface modeling. To make the clothing fit the characters, the character was exported from ZBrush and imported to 3ds Max without positioning or scaling it. The clothing was then modeled according to the size of the imported character. By doing so, ZBrush properly positioned the clothes on the character. Some of the accessories such as the necklace needed surface modeling. For these models, the PolyDraw tool in 3ds Max was used to create polygons on the surface directly. Giving them a shell modifier made them look like chains or clothes that are on the skin of the character.

After modeling the character itself and all the accessories, decimation master was used to decrease the number of the polygons of the model. When all the models have achieved the desired low poly appearance but still easily recognizable, they were transferred to 3ds Max for assembly and scaling for Unity.

### 5.3.2: Overall Design and Inspiration

There are two categories of characters in our game: the gods and the evils. Originally, we had planned for five gods and five evils for the game, but due to scoping reasons, we were only able to implement three of each.

The three gods are Inari, Susanoo, and Tenman Tenjin. The three evils are Akashita, Yamata no Orochi, and Sugawara no Michizane.

The gods are designed to be faceless humanoids with masks and clothing that identifies them. The idea of making the gods wear masks comes from the famous Japanese traditional music drama,

Noh (Figure 5.11). This performance originates from the 14th century. Masks are always involved in such performances [12]. The performers of Noh believe that rather than makeup, masks have certain inherent powers and have a spiritual quality [43]. Thus, giving the gods masks made their appearance more mysterious.



*Figure 5.11: Noh Masks [38]*

The evils are all traditional rivals of the gods we chose, with the exception of Akashita. They all look different so we had to carefully make each one individually look good.

### 5.3.3: God Inari

Inari is the god in Fushimi Inari Taisha. He is the god of rice and prosperity. In Shinto mythologies, foxes are considered as the guardian of Inari so our model of Inari wears a fox mask. The mask has red carved lines to add complexity and decoration. Inari also wears a fox tail to add additional ties to the fox guardians. For the clothing, Inari is wearing a Japanese kendo style long



skirt. His belt is a small torii, as Senbon Torii is the most famous landmark of Fushimi Inari Taisha.



*Figure 5.12: The Model of the God Inari and His Mask*

#### 5.3.4: God Susanoo

Susanoo is the god enshrined in Yasaka Jinja. He is one of the oldest gods in Japan. He is a warrior but is also known for his anger. Many disasters are attributed to him, but it is said that he also helps Japanese people. The most famous event he is known for is probably when he killed Yamata no Orochi. Susanoo's mask was made by combining several Noh mask features that emphasized bravery and anger. This design is also partially inspired by Susanoo in *SMITE* [44]. He wears a straw hat and traditional Japanese armor.



*Figure 5.13: Susanoo and His Mask*

### 5.3.5: God Tenman Tenjin

Tenman Tenjin is depicted as a wise old man. He was a governor and politician of the Heian Period. His mask was designed to look like the face of an old man. The clothing is inspired by the uniform of politicians in the Heian Period. The long sleeves and slippers help identify this. The color scheme of Tenman Tenjin is black and white, following the color of the uniform. Black and white also give players an illusion of solemnness. The necklace is inspired by the mark on the lanterns in Kitano Tenmangu, the shrine of Tenman Tenjin in Kyoto.



*Figure 5.14: Tenman Tenjin and His Mask*

### 5.3.6: Akashita

Akashita is a beastal monster with a long red tongue and its body is covered by a dark cloud [29]. Inari does not have a traditional enemy, however Inari is the god of prosperity and harvest, so a yokai that appears on farms and in fields was appropriate. Akashita is said to have appeared in those locations, and appeared to be a good foil to Inari.



Figure 5.15: Akashita in a Traditional Image [46] and the Model of Akashita

### 5.3.7: Yamata no Orochi

Yamata no Orochi is the monster killed by Susanoo in Japanese mythology. He has eight snake heads and a lizard body.



Figure 5.16: Yamata no Orochi Model

### 5.3.8: Sugawara no Michizane

Sugawara no Michizane is the original name of Tenman Tenjin. He is one of the most admired gods in Japan, but also one of the most powerful evil spirits in Japan. To reflect the

connection between him and Tenman Tenjin, this evil model was modified from the model of Tenman Tenjin. One third of the mask was deleted to imply that Sugawara no Michizane is the evil version of Tenman Tenjin. The mask is the symbol of the god's divinity, so breaking the mask means the sanity of the god is contaminated. One of his sleeves was also changed to a glove. This asymmetrical design suggests the evilness of Sugawara no Michizane. The drums floating behind the character are based on traditional images of Sugawara.



*Figure 5.17: Sugawara no Michizane Model and Mask Model*

## 5.4: Music

All of the music for the game was chosen to fit the theme of each minigame. The music in the game is from <http://www.rengoku-teien.com/>. We specifically used music from this website because it is available for free for a multitude of uses and it was composed by a collection of Japanese musicians.

## Chapter 6: Playtesting, Feedback

Our playtester pool was rather small due to how difficult it was to find playtesters, so the data in this section may not be as accurate of a representation of the overall target audience as we would have liked. That said, we were still able to get valuable feedback about our game and the people playing it. All of the data we reference can be found in Appendix C, and the survey questions can be found in Appendices A and B. We were able to get two testers to play the game in its entirety, who were able to travel to the various POIs in Kyoto, as well as four additional playtesters to test all of the features of our game without actually traveling. We aimed to recruit people from the BKC International House through both posters and by asking them directly, but this yielded zero participants.

### 6.1: Background of Playtesters (Pre-Survey information)

Our playtesters were majority male 20-25 year old WPI students, traveling together with some knowledge of Japan. This prior knowledge of Japan certainly has limited our data on how players that have no knowledge of Japan would feel about the game. Our playtesters are well mixed regarding the kind of mobile game they prefer most, with RPGs being the most popular. We asked this to understand if these players had any experience with similar games. The amount of time they spend playing mobile games per day is split evenly between 15 minutes or less, and 60 minutes or more for the players that play mobile games.

### What kind of mobile games do you play most?

6 responses

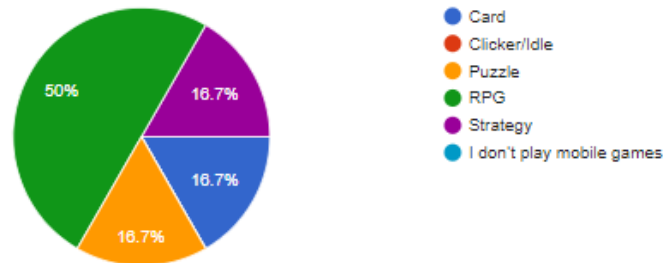


Figure 6.1: Types of Mobile Games

### About how much time do you spend playing mobile games a day?

6 responses

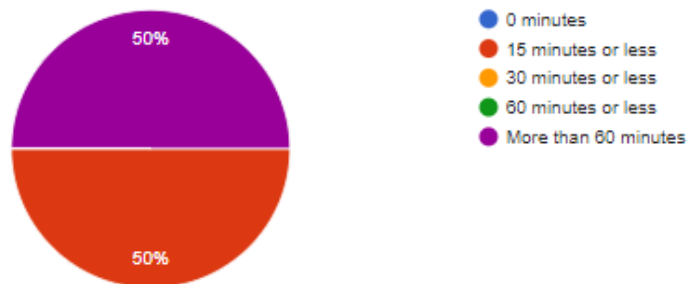


Figure 6.2: Time Playing Mobile Games

Most of our playtesters wanted to see shrines and temples in Kyoto, compared to more generic interests such as “sightseeing” and “malls and restaurants.” This information supports our decision of highlighting POIs such as these.

We also asked players: 1) if they have taken guided tours in the past, 2) if they plan on taking a guided tour, and 3) if they have taken guided tours in the past. Our playtesters responded that they are not interested in guided tours whatsoever, despite half of them having taken guided tours in the past. This data helped to affirm our decision to not fully direct players to specific locations and allow

them to explore on their own. They also said they would enjoy a game such as ours to navigate around, so our game concept still seemed viable.

## 6.2: How Playtesters Felt (Post-Survey Information)

We wanted to get an overall opinion of our game to see how playtesters felt about it as a whole. The overall opinion on our app differed greatly from tester to tester, however the two most prominent feelings our playtesters had were “confused” and “happy”. The players were given the options of: Excited, Scared, Annoyed, Frustrated, Confused, Happy, Surprised, Bored, and other. The playtesters explanation for the response of confusion was due to a lack of game instruction. We only had the location-specific characters to narrate, and didn’t have any sort of guide narrator to walk the player through the game’s system and goals. Because of this, when the player loads the game, they are taken directly to the main menu with the list of minigames fully locked, and no context as to what to do. Confusion was understandable in hindsight.

What feeling did you get when using the app? Check all that apply.

6 responses

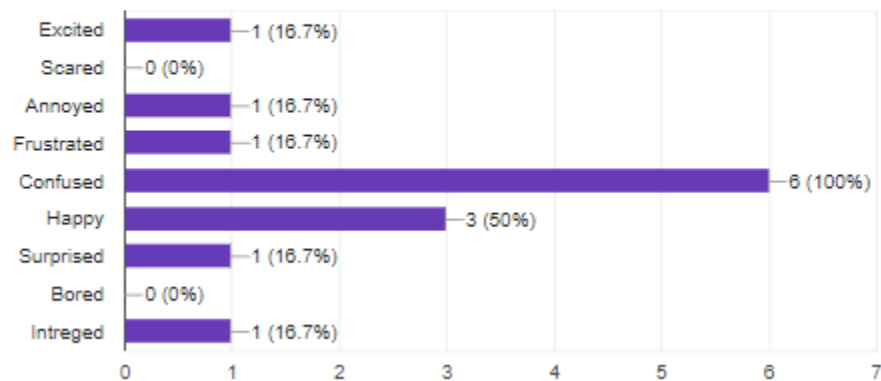


Figure 6.3: Playtesters Feelings on Gojin



Half of the playtesters said the game made them “happy,” and liked completing as much of the game as possible, overall feeling it was a good way to explore. There were opinions voiced that some minigames were harder than others, with the harder ones being preferred. This leads us to believe that some of the easier minigames should have been more difficult. This was a concern we were aware of, however it was simply a matter of trying to “balance” six different minigames, when each one requires careful consideration.

Other thoughts that were voiced included the application needing more to do, since the overall time playing the game is far less than the time actually traveling (per location: ~5-10 minutes playing our game, compared to 1-2 hours exploring). From the beginning of the project, we thought of giving interactivity to intermediary locations, however we were not able to implement this. This seems to be one of the biggest complaints our playtesters had, in some form.

Despite mixed opinions on various parts of the game, nearly all of the playtesters enjoyed the 3d Map and the Minigames. However, nobody used the built-in navigation system. We wanted to include a navigation system, however it was out of the scope of this project to create a navigation system to replace Google Maps.

### 6.2.1: Minigame Ratings

To better understand playtesters opinion of the minigames, and see how we can better adjust each one, we asked them to rate each game they played on a scale of 1-4 labeled Terrible, Okay, Good, and Very Good. The minigame our playtesters rated the highest was Ginkakuji Unblock Me. This was likely because it was the most balanced, having the levels being identical to some *Unblock*

Me [13, 23] levels.

Some minigames had mixed reviews, as shown in Table 6.1, such as Fushimi Inari Run and Myoshinji Temple Jigsaw, which seemed to be due to game difficulty issues. Fushimi Inari Run was visually appealing to our players, however some said it was too easy, with not enough obstacles. Myoshinji Temple Jigsaw, on the other hand, was too difficult for some people.

Minigame	1 star rating	2 star rating	3 star rating	4 star rating	Did not play
Fushimi Inari Run		2	3	1	
Nijo Castle Tower Defense	1	3	1	1	
Yasaka Shrine Geisha Rhythm		3	2		1
Ryoanji Line Trace		3	1		2
Ginkakuji Unblock the Moon			2	3	1
Myoshinji Jigsaw Puzzle	1	1	2		2

Table 6.1: Playtester Ratings of Minigames

A notable complaint we received was that there was not enough guidance to help the player play our minigame. We had the tutorial screen for each minigame, however perhaps an animated step-by-step system would have been more helpful to the player.

## Chapter 7: Postmortem

### 7.1: What was successful

#### 7.1.1: Team Dynamics (and Division of Labor)

Our team had a very successful team dynamic among us. Throughout the project we all maintained communications as a team without any issue arising. We were able to talk to each other in a friendly manner. Planning out our game and software layout also was excellent, as will be explained in the Integration section below. Work division was also rather agreeable. Natalie and Dan both handled technical elements of the game, with Natalie focusing more on the minigames and Dan focusing more on some features such as the location-based map, and the unlock/transition system, among other things. Kai, as the artist, produced art for the game. We felt like this was a strong and productive setup for our members.

#### 7.1.2: Integration

When it came time to integrate our parts, such as having the minigames interact with the world and map components, it was extremely easy and hassle-free to combine everything into one cohesive project. Every piece of the game so well encapsulated within itself by intentional design. The only thing we had to manage was adding the connections between each piece, which was very minimal lines of code aside from the unlock system. Furthermore, this organizational system was able to sustain itself for the duration of the project. We made sure not to work on the same piece at the same time, thus giving us no merge conflicts whatsoever.

### 7.1.3: Scoping

The way we scoped our project was not flawless, but it was arguably one of our successes. The daily pace of work was very comfortable, yet productive. We always got things done, however we did not feel rushed even at the end. We feel that we were able to achieve a relatively polished product in the time allotted to the project.

## 7.2: What could have been improved

### 7.2.1: Team Communication

Despite all three of our members feeling our team dynamics were excellent, there was still a lack of communication between our members which ultimately affected the end product. Throughout the development process, the art assets were sometimes being produced at a slower rate than we needed them, since one of members was limited to staying in the dormitory for most of the week. Our communication with him was rather limited simply because it was usually only available through messaging, rather than face to face interaction. This minimal interaction was why we had a critical art asset for the wrong location in our game, forcing us to improvise it at the last minute.

Similarly, we were not able to receive art assets fast enough to complete the game as much as we intended. We only had 3 locations' God and Evil art assets completed, out of the five we intended.

### 7.2.2: Difficulties in Acquiring Playtesters

Firstly, it is worth noting that it was exceptionally hard to find playtesters for our game, compared to other projects created by other groups. The issue is how narrow our target audience is,

combined with finding people within this narrow audience willing to dedicate time to playtest, with no real incentive other than enjoyment.

Furthermore, weather was also a concern. The very first day we attempted to start recruiting, for example, it was raining outside so there was absolutely no interest in doing an outdoor activity. The weather from Typhoon Jebi also had many culturally relevant locations still undergoing renovations, so this might be somewhat off-putting to a tourist.

The BKC International House seemed like a perfect place to recruit playtesters, because many of the students living there speak English, are here from overseas, and had recently moved into the dorm. When we approached a number of the students, however, they did not show much interest in playtesting, often because they were not planning to go to Kyoto very soon, or at least not for cultural visitations, and were instead busy with their own projects and plans. This kind of sentiment was very common among the potential playtesters we asked. We were also often ignored in our attempts to gather playtesters, leading to the use of WPI students only. The other people we tried to recruit were not willing to help playtest, either because of the time commitment it would require, or because they did not want to travel.

It was because of this that we were not able to reach the 20 playtesters we were aiming for, however we were able to get playtesting feedback for each of our separate components much easier, as that did not require the time and expenses of traveling to Kyoto.

### 7.2.3: Game Playability

The game lacks a proper potential audience, we've found. Part of the issue is that our initial target audience is rather limited; we only targeted English-speaking tourists who would want to use an application/game to guide their travels. Part of the feedback we received from our very limited location playtesting was that if a tourist is here to travel, they likely will have other resources to guide them like an experienced traveler, online guides, and Google Maps. Many travelers also prefer to simply explore on their own, without any such guides, especially if they are staying here longer than a week, perhaps. If our game had a lot more time to develop and make it fine-tuned enough so it could guide them culturally as a standalone application, adding more characters to add a personal connection, and actually implementing experiences for more locations, our application would likely be easier to play. It was an issue of not being able to develop locations as fast as we expected, as we only ended up with 6 minigames, whereas we were aiming for 10, and a full version of the game would have an experience for each location that was large enough (at least 25).

Furthermore, due to how long the minigames took to fully assemble, we were not able to implement many features that gives the game playability outside of these six set locations. *Pokémon Go* [33], for example, has Pokémon you can capture everywhere, and PokéStops to spin, whereas we have nothing of this sort. We also originally intended to include an AR portion into the game to supplement the minigames, but because of low prioritization, it was abandoned early on.

The game isn't as culturally relevant as we would have liked; we had many great ideas for things they could do at each location, but those got overshadowed by more technically important

things. Cultural relevance was later meant to be added in the cutscenes, however it was difficult even then, given we could not display too many art assets.

Due to a lack of iterative testing the minigames have a number of usability issues. The first was a lack of clear instructions on how to play the minigames, as discussed section 6.2.1. Players were unsure initially of how to play the minigames, but because the team was present during testing, the games were easily explained in person. If this project was to be continued, we would have added a step-by-step animated tutorial for each minigame and a button to access the tutorial at any point in the future. For Fushimi Inari Run, the loading time of the platforms and objects often slowed or froze the game. What we would have done to fix this is to simplify the number of 3D objects the game must generate, through reducing the number of environmental assets, and make the spawning algorithm move existing off-screen objects rather than delete and recreate them with each platform. For Ryoanji Line Trace, the stroke detection was finicky and hard to match due to it comparing against a single reference stroke. To fix this, we would have added more reference strokes for comparison to increase the chances of getting a match, as well as have more user testing to determine a fair minimum accuracy score. For the Yasaka Shrine/ Gion District Rhythm Game there is sometimes a mismatch between the beat of the songs and the notes due to an unclear tempo, and if we had the opportunity to revise this we would have written our own music; this would allow for a set tempo to easily determine when to spawn the notes and fix the audio-visual mismatch. In the case of Nijo Castle Tower Defense, the major issues are the width of the enemy path and the initial range of the level 1 towers. To resolve these issues we would shrink the size of the enemy path tiles, then allowing for the towers to get closer to the enemies and mitigating the need for increasing the

initial range of the towers to an extent, and also with player testing increase the size of the towers' range. For Myoshinji Jigsaw Puzzle, we initially designed the puzzle on a full-sized computer screen, so we had the ability to model the entire puzzle based closely on the real map of Myoshinji. The actual proportion and shape of the temple pieces and ground pieces is maintained. This ended up being a problem for users, as there ended up being too many temple pieces, where each one is too small. We would instead not follow the real-world map as accurately, and instead make a simplified, smaller puzzle, with the temple pieces being significantly bigger. This puzzle also had a coloring issue, in that the pieces did not have borders, which sometimes made some pieces harder to see. This would simply be resolved by giving every piece a solid border. Ginkaku-ji Unblock the Moon had the issue that the Moonlight block was sometimes hard to see, which confused some users. This was in-part intentional to make it look like actual light, rather than a block, but this in hindsight understandably added usability issues. We would still try to make the Moonlight block resemble light, but make it easier to understand what the block is.

Another major issue that influenced playability is the variety of phones that run Android. Because there is no set resolution for these phones, it was difficult to ensure that the app would work properly on every phone. Some of the text ended up too small on certain devices. In future iterations we would have liked to allow the app to resize the display based on the resolution of the current device to resolve the issue of illegible text.



#### 7.2.4: More options

Our options for minigames felt rather limited; all of our minigames took heavy inspiration from other well-known games. Having many more, smaller, faster minigames, and more original game concepts would give more body to the gameplay.

### 7.3: Areas of future exploration

If we were to continue to develop this game, there are many things we would continue to improve, notably ones mentioned in our postmortem. Firstly, we would simply have more of what we have. We have 6 minigames, 3 cutscenes, out of dozens of possible locations. In a hypothetical completed version, all of these relevant locations would have their own cutscene and minigame, with better animations than simply two art animations moving up and down, talking. They would also have a lot deeper cultural relevance, such as a task list for each location. Bowing at the shrine, scavenger hunts, or many other tasks that enforce cultural learning can be added. This includes tasks the player would do between major POIs.

We would also like a bigger playtesting group. We had a relatively good amount of time at the end of our travel to track down playtesters, however due to the increased difficulty of finding playtesters for a location-based game, compared to a VR or desktop game, we still were not able to find as many playtesters as we would have liked. Increased time would give us increased ability to playtest.

We could also expand our audience to Japanese tourists in Kyoto, or expand our range to more than just Kyoto. This style of gameplay can be theoretically relevant to many locations.



## Chapter 8: References

- [1] Andy king50. WikiMedia Commons, Kyoto: Wikimedia. Retrieved from <https://commons.wikimedia.org/wiki/File:FushimiInariTorii.jpg>
- [2] Becky Yoose. INARI = Shinto Rice Kami. Retrieved September 30, 2018 from <http://www.onmarkproductions.com/html/fox-inari-university-of-wiscon.html>
- [3] Clover Studio. 2006. Okami. PlayStation 2. Capcom.
- [4] Cquest. 2007. Ryoan Ji, Kyoto zen garden. Retrieved from <https://commons.wikimedia.org/wiki/File:Kyoto-Ryoan-Ji MG 4512.jpg>
- [5] Fg2. 2009. Myoshinji. Retrieved from <https://commons.wikimedia.org/wiki/File:Myoshinji-M9707.jpg>
- [6] Fishing Cactus. 2018. Epistory. (April 2018). Retrieved September 30, 2018 from <https://en.wikipedia.org/wiki/Epistory>
- [7] Fishing Cactus. 2016. Epistory - Typing Chronicles. PC, Mac. Plug In Digital
- [8] Fushimi Inari Taisha. Fushimi Inari Taisha. Retrieved September 30, 2018 from <http://inari.jp/en/>
- [9] Geisha of Japan. 2018. Music – Jikata. Retrieved September 30, 2018 from <http://geishaofjapan.com/art/music-jikata/>
- [10] Gordon Cheung. Retrieved October 1, 2018 from [https://commons.wikimedia.org/wiki/File:Nijo\\_Castle\\_171206\\_\(24263278197\).jpg](https://commons.wikimedia.org/wiki/File:Nijo_Castle_171206_(24263278197).jpg)

- [11] Guest Blogger. 2018. An Intro to Low-Poly and Flat Design. (April 2018). Retrieved September 30, 2018 from <https://developer.amazon.com/fr/blogs/appstore/post/aaedd3b8-5e3f-4b4b-a567-c3257139cbcf/an-intro-to-low-poly-and-flat-design>
- [12] History of Masks. 2018. Noh Mask - Types of Noh Masks and Noh. Retrieved September 30, 2018 from <http://www.historyofmasks.net/famous-masks/noh-mask/>
- [13] iGamesHelper123. 2013. Unblock Me 1000 levels in 40 minutes. (January 2013). Retrieved September 30, 2018 from <https://www.youtube.com/watch?v=CQL3BrqfuT0>
- [14] Imangi Studios. 2011. Temple Run. iOS. Imangi Studios.
- [15] Intelligent Systems, Nintendo SPD. 2007. Warioware: Smooth Moves. Wii. Nintendo.
- [16] Japan Guide. 2018. Ginkakuji (Silver Pavilion) Temple in Kyoto. Retrieved September 30, 2018 from <https://www.japan-guide.com/e/e3907.html>
- [17] Japan Guide. Nijo Castle (Nijojo) Castle in Kyoto. Retrieved September 30, 2018 from <https://www.japan-guide.com/e/e3918.html>
- [18] Japan Guide. Yasaka Shrine Shrine in Kyoto. Retrieved October 1, 2018 from <https://www.japan-guide.com/e/e3903.html>
- [19] Josh O. Caoimh. Advantages and Disadvantages of the 'Low Poly' style in Game Design. Retrieved October 1, 2018 from <http://www.joshocaoimh.com/ramble/advantages-and-disadvantages-of-the-low-poly-style-in-game-design>
- [20] Kaitlyn Ellison. 2016. The secrets to successful geometric low-poly art. (January 2016). Retrieved September 30, 2018 from <https://en.99designs.jp/blog/tips/the-secrets-to-successful-geometric-low-poly-art/>

- [21] Karen Ann Smyers. 1999. The fox and the jewel: shared and private meanings in contemporary Japanese Inari worship, Honolulu: University of Hawaii Press.
- [22] Kevin Shau. 2018. Early Japanese Shinto III: Susanoo and Yamata no Orochi. (February 2018). Retrieved September 30, 2018 from <https://medium.com/@kevinshau/early-japanese-shinto-iii-susanoo-and-yamata-no-orochi-aac15a03065d>
- [23] Kiragames. 2010. Unblock Me. Android, iOS. Kiragames
- [24] Mads Møller. Gesture Recognizer. Retrieved September 30, 2018 from [http://wiki.unity3d.com/index.php/Gesture\\_Recognizer](http://wiki.unity3d.com/index.php/Gesture_Recognizer)
- [25] Mapbox. Mapbox. Retrieved September 30, 2018 from <https://www.mapbox.com/>
- [26] Mark Cartwright. 2012. Susanoo. (December 2012). Retrieved September 30, 2018 from <https://www.ancient.eu/Susanoo/>
- [27] Mark Cartwright. 2017. Tenjin. (May 2017). Retrieved September 30, 2018 from <https://www.ancient.eu/Tenjin/> 
- [28] Matsunaga Naomichi. 2005. Yamatanoorochi. (May 2005). Retrieved September 30, 2018 from <http://eos.kokugakuin.ac.jp/modules/xwords/entry.php?entryID=176>
- [29] Matthew Meyer. Aka shita. Retrieved September 30, 2018 from <http://yokai.com/akashita/>
- [30] Might and Delight. 2013. Shelter. Retrieved from [https://commons.wikimedia.org/wiki/File:Shelter\\_-\\_Screenshot\\_02.jpg](https://commons.wikimedia.org/wiki/File:Shelter_-_Screenshot_02.jpg)
- [31] Might and Delight. 2013. Shelter. Windows. Might and Delight. Retrieved from <http://mightanddelight.com/shelter/>

- [32] Myoshinji Temple. 2008. Myoshinji Temple. Retrieved September 30, 2018 from <http://www.myoshinji.or.jp/english/>
- [33] Niantic. 2016. Pokémon Go. Android, iOS. Niantic.
- [34] Ninja Kiwi. 2011. Bloons Tower Defense 5. Web Browser (Flash). Ninja Kiwi
- [35] Nob Yoshigahara. 1996. Rush Hour. Think Fun.
- [36] Ph0kin. 2014. Kyoto Silver Pavilion Pond. Retrieved from [https://commons.wikimedia.org/wiki/File:Kyoto\\_Silver\\_Pavilion\\_Pond.jpg](https://commons.wikimedia.org/wiki/File:Kyoto_Silver_Pavilion_Pond.jpg)
- [37] Ryoanji Temple. 大雲山 龍安寺 | Ryoanji. Retrieved September 30, 2018 from <http://www.ryoanji.jp/smph/eng>
- [38] sailko. 2018. Noh. (September 2018). Retrieved September 30, 2018 from <https://en.wikipedia.org/wiki/Noh>
- [39] Schulhofpassage. 2016. Yasaka Shrine. Retrieved from [https://commons.wikimedia.org/wiki/File:Yasaka\\_Shrine\\_\(24646034510\).jpg](https://commons.wikimedia.org/wiki/File:Yasaka_Shrine_(24646034510).jpg)
- [40] Statista. 2018. Distribution of video gamers worldwide in 2017, by age group and gender | Statistic. Retrieved October 1, 2018 from <https://www.statista.com/statistics/722259/world-gamers-by-age-and-gender/>
- [41] Statista. 2018. U.S. average age of video gamers 2018 | Statistic. Retrieved October 1, 2018 from <https://www.statista.com/statistics/189582/age-of-us-video-game-players-since-2010/>
- [42] Tap Tap Revenge. 2008. Tapulous. Android, iOS. Tapulous.
- [43] The-Noh.com. Introducing the world of Noh: Noh Masks. history and Types. Retrieved September 30, 2018 from <http://www.the-noh.com/en/world/mask.html>

- [44] Titan Forge Games. 2014. Smite. Windows. Hi-Rez Studios
- [45] Tokyo TV. 銀魂 テレビ東京アニメ公式. Retrieved September 30, 2018 from <http://www.tv-tokyo.co.jp/anime/gintama/>
- [46] Toriyama Sekien. 2018. Akashita. (September 2018). Retrieved September 30, 2018 from <https://en.wikipedia.org/wiki/Akashita>
- [47] Toriyama Sekien. akashita. Retrieved September 30, 2018 from <https://pantheon.org/articles/a/akashita.html>
- [48] Unity. 2018. Unity. Retrieved October 1, 2018 from <https://unity3d.com/>
- [49] Yasaka Shrine. History of Yasaka Shrine | Yasaka-Shrine. Retrieved October 1, 2018 from <http://www.yasaka-jinja.or.jp/en/>
- [50] Yokai.com. 2018. Aka shita - Yokai.com. Retrieved September 30, 2018 from <http://yokai.com/akashita/>
- [51] Zack Davisson. 2013. A Brief History of Yokai. (February 2013). Retrieved September 30, 2018 from <https://hyakumonogatari.com/2013/02/05/a-brief-history-of-yokai/>

## Appendix A: Pre-Playtesting Survey

\* Required

1. What is your gender? \* *Mark only one oval.*
  - Female
  - Male
  - Prefer not to say
  - Other: \_\_\_\_\_
2. What is your age? \*
  - \_\_\_\_\_
3. What nationality are you? \*
  - \_\_\_\_\_
4. How long is your visit to Japan? (Number of days) \*
  - \_\_\_\_\_
5. Is this your first visit to Japan? \* *Mark only one oval.*
  - Yes
  - No
6. Is this your first visit to Kyoto? \* *Mark only one oval.*
  - Yes
  - No
7. How long do you plan to spend in Kyoto? (Number of days) \*
  - \_\_\_\_\_



8. Are you traveling alone? \* *Mark only one oval.*
- Yes *Skip to question 9.*
  - No *Skip to question 10.*
9. Please Indicate the number of people you are traveling with: \*
- \_\_\_\_\_
10. How familiar are you with the culture of Japan? \* *Mark only one oval.*
- Totally unfamiliar
  - Slightly
  - Moderately
  - Very knowledgeable
11. What kind of mobile games do you play most? \* *Mark only one oval.*
- Card *Skip to question 12.*
  - Clicker/Idle *Skip to question 12.*
  - Puzzle *Skip to question 12.*
  - RPG *Skip to question 12.*
  - Strategy *Skip to question 12.*
  - I don't play mobile games *Skip to question 13.*
12. About how much time do you spend playing mobile games a day? \* *Mark only one oval.*
- 0 minutes
  - 15 minutes or less
  - 30 minutes or less

- 60 minutes or less
- More than 60 minutes

13. What do you plan to do in Kyoto? \* *Check all that apply.*

- See shrines
- Visit temples
- Visit Kyoto Tower
- Other: \_\_\_\_\_

14. Would you consider taking a guided tour? \* *Mark only one oval.*

- Yes
- No

15. Do you plan to take a guided tour? \* *Mark only one oval.*

- Yes
- No

16. Have you taken guided tours in the past? \* *Mark only one oval.*

- Yes
- No

17. How likely would you be to use a mobile app to help navigate around? \* *Mark only one oval.*

- Not at all.
- Somewhat likely.
- Likely.
- Very likely.

18. Would you like a game that helps you explore Kyoto? \* *Mark only one oval.*

- Yes
- No
- Maybe

19. Please enter your email \*

## Appendix B: Post-Playtesting Survey

\* Required

1. Please enter the same email from the first survey. \*
  - \_\_\_\_\_
2. How familiar are you with the culture of Japan? \* *Mark only one oval.*
  - Totally unfamiliar
  - Slightly
  - Moderately
  - Very Knowledgeable
3. What do you plan to do in Kyoto? Check all that apply. \* *Check all that apply.*
  - See shrines
  - Visit temples
  - Visit Kyoto tower
  - Other: \_\_\_\_\_
4. What feeling did you get when using the app? \* *Check all that apply.*
  - Excited
  - Scared
  - Annoyed
  - Frustrated
  - Confused
  - Happy

- Surprised
- Bored
- Other: \_\_\_\_\_

5. What made you feel this way? \*

- \_\_\_\_\_

6. Did you feel a sense of exploration from the app? \* *Mark only one oval.*

- Yes
- No
- Maybe

7. Did you enjoy using the app? \* *Mark only one oval.*

- Yes
- No
- Maybe

8. What features did you like in the app? Check all that apply. \* *Check all that apply.*

- Minigames
- Augmented reality map
- Direction system
- Other: \_\_\_\_\_

9. Why? \*

- \_\_\_\_\_

10. What features did you not like in the app? \* *Check all that apply.*

- Minigames
- Augmented reality map
- Direction system
- Other: \_\_\_\_\_

11. Why? \*

- \_\_\_\_\_

12. Do you feel like the app was a good way to explore Kyoto? \* *Mark only one oval.*

- Yes
- No
- Maybe

13. Why? \*

- \_\_\_\_\_

14. How likely would you be to use a mobile app to help navigate around? \* *Mark only one oval.*

- Not at all.
- Somewhat likely.
- Likely.
- Very likely.

15. Would you like a game that helps you explore Kyoto? \* *Mark only one oval.*

- Yes
- No
- Maybe

## Minigame Rating

16. Fushimi Inari Run\* *Mark only one oval.*
- Terrible
  - Okay
  - Good
  - Very good
  - Did not play
17. Nijo Castle Tower Defense \* *Mark only one oval.*
- Terrible
  - Okay
  - Good
  - Very good
  - Did not play
18. Yasaka Shrine / Gion District Rhythm Game \* *Mark only one oval.*
- Terrible
  - Okay
  - Good
  - Very good
  - Did not play
19. Ryoanji Line Trace \* *Mark only one oval.*
- Terrible

- Okay
- Good
- Very good
- Did not play

20. Ginkaku-ji Unblock the Moon \* *Mark only one oval.*

- Terrible
- Okay
- Good
- Very good
- Did not play

21. Mioshinji Jigsaw Puzzle \* *Mark only one oval.*

- Terrible
- Okay
- Good
- Very good
- Did not play

22. Please enter any additional comments or suggestions here \*



# Appendix C: Playtesting Data

## Pre-Playtest Survey

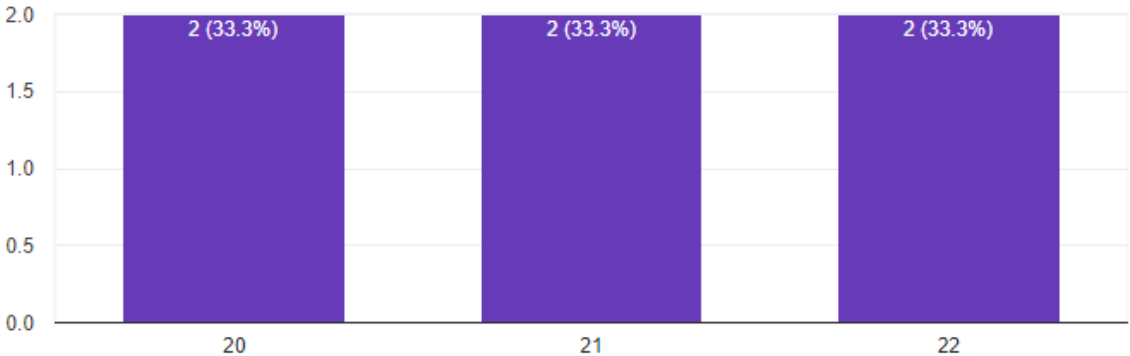
What is your gender?

6 responses



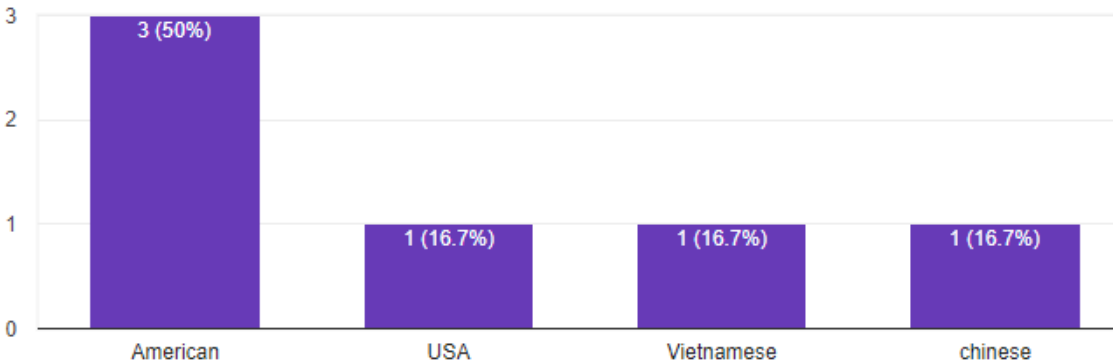
What is your age

6 responses



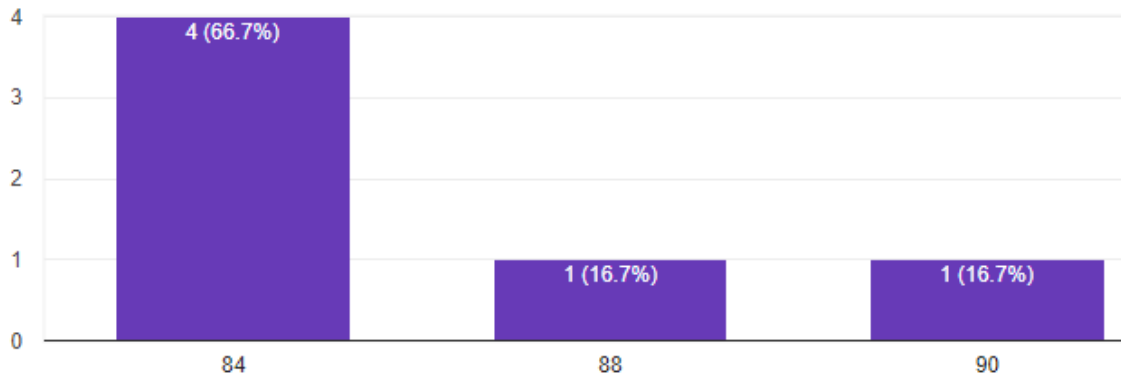
What nationality are you?

6 responses



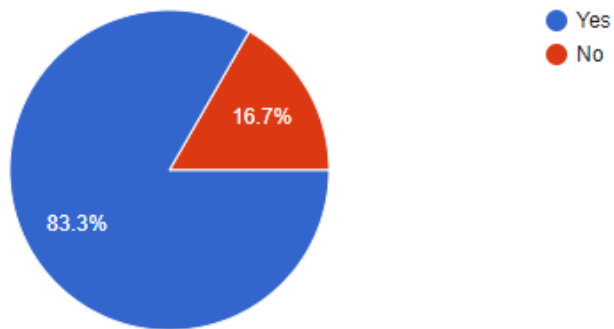
## How long is your visit to Japan? (Number of days)

6 responses



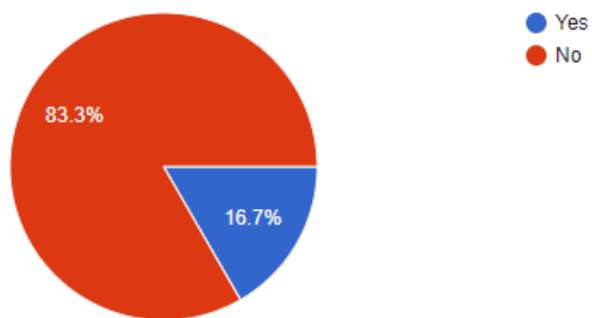
## Is this your first visit to Japan?

6 responses



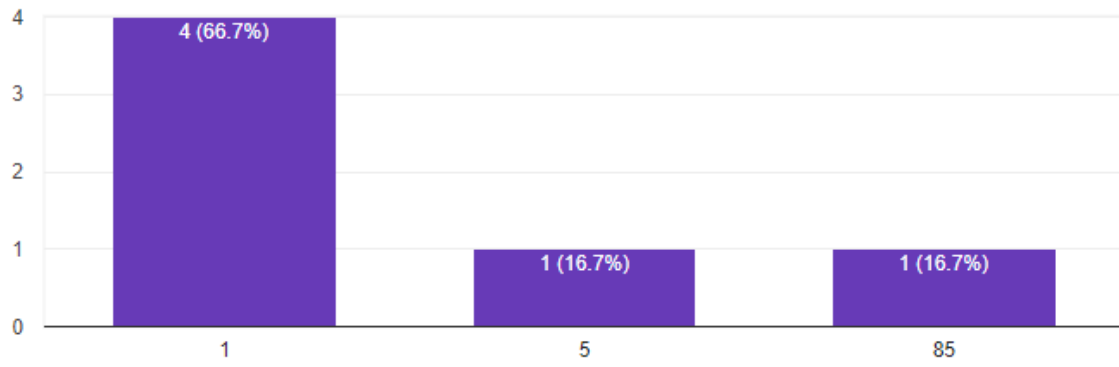
## Is this your first visit to Kyoto?

6 responses



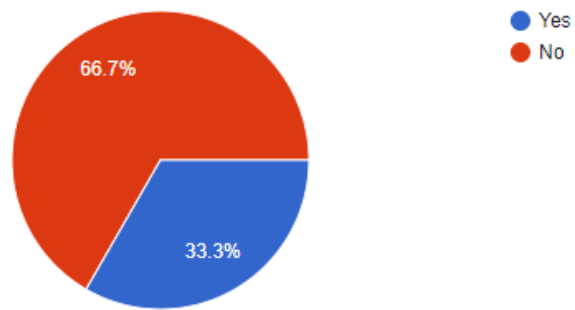
### How long do you plan to spend in Kyoto? (Number of days)

6 responses



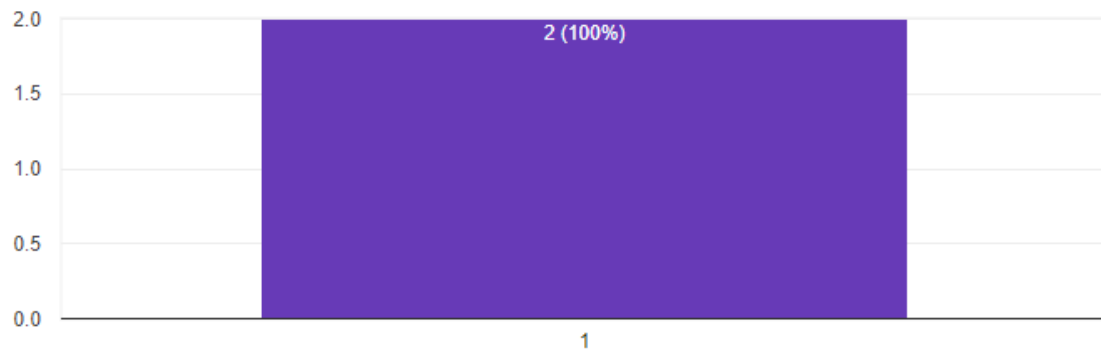
### Are you traveling alone?

6 responses



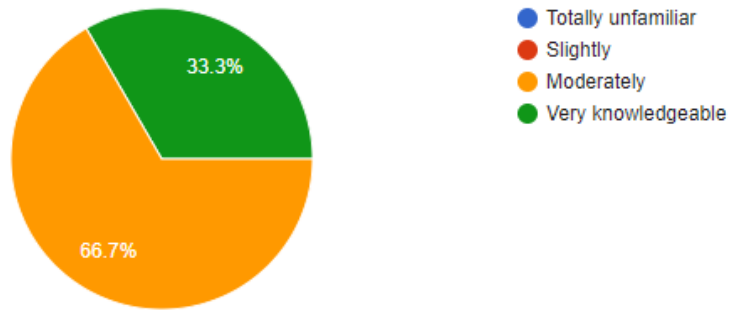
### Please Indicate the number of people you are traveling with:

2 responses



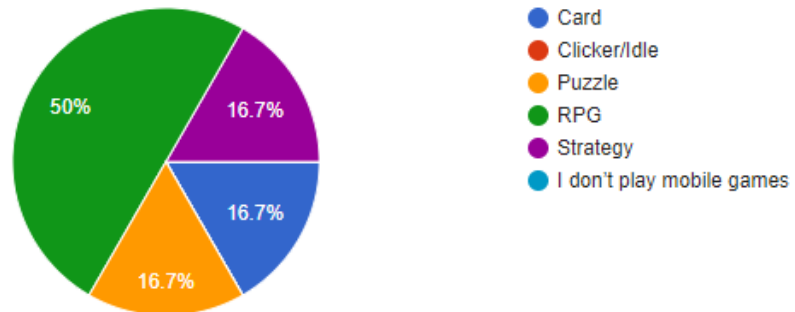
### How familiar are you with the culture of Japan?

6 responses



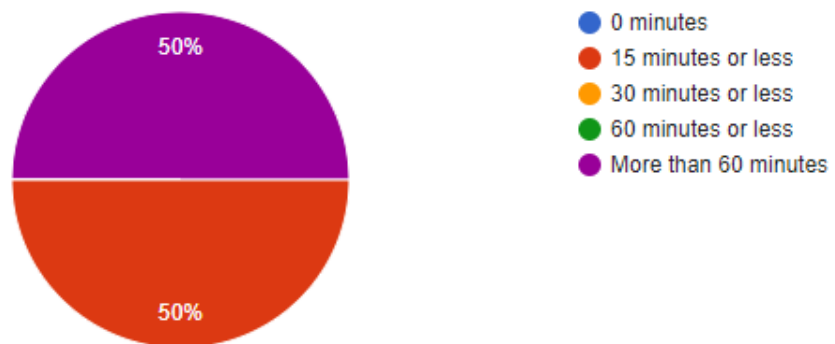
### What kind of mobile games do you play most?

6 responses



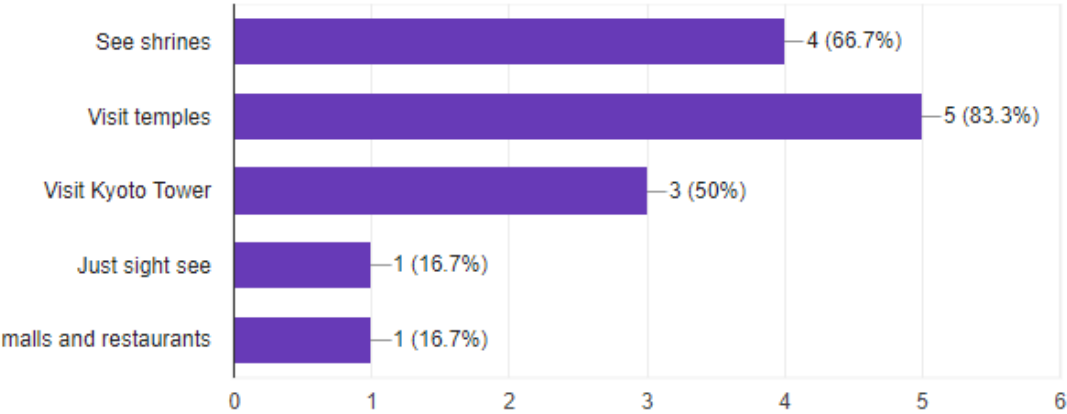
### About how much time do you spend playing mobile games a day?

6 responses



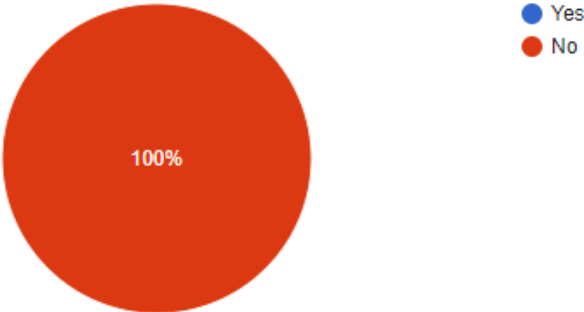
# What do you plan to do in Kyoto?

6 responses



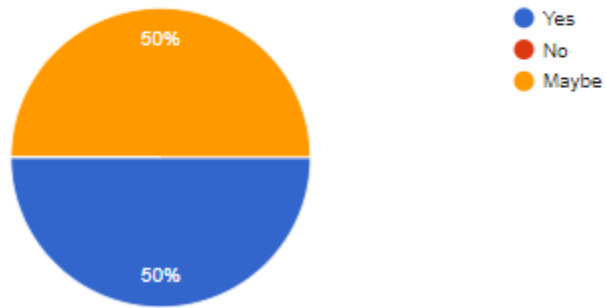
# Would you consider taking a guided tour?

6 responses



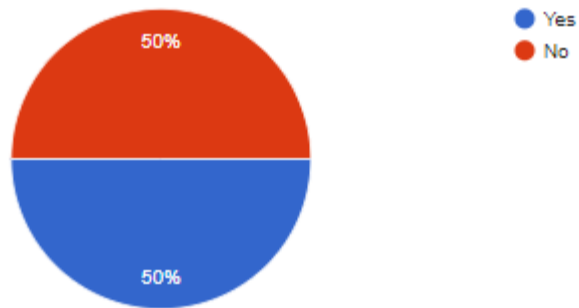
### Would you like a game that helps you explore Kyoto?

6 responses



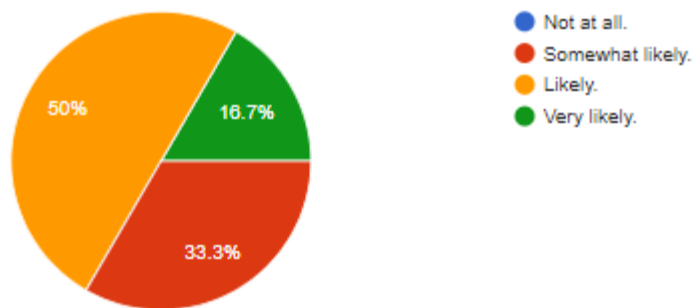
### Have you taken guided tours in the past?

6 responses



### How likely would you be to use a mobile app to help navigate around?

6 responses

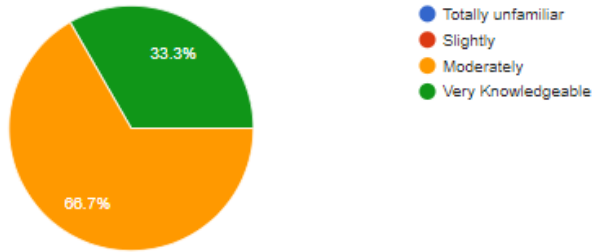


# Post-Playtest Survey

What f How familiar are you with the culture of Japan?

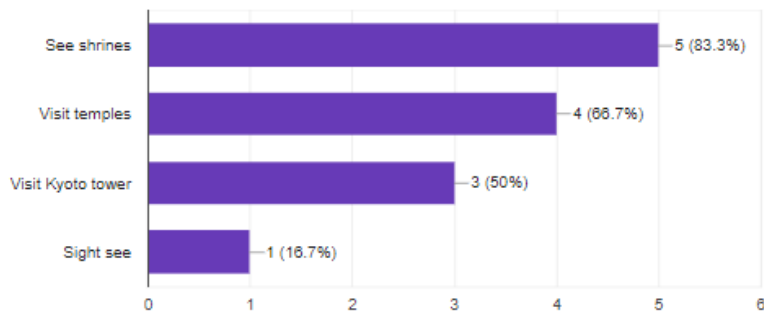
6 respons

6 responses



What do you plan to do in Kyoto? Check all that apply.

6 responses



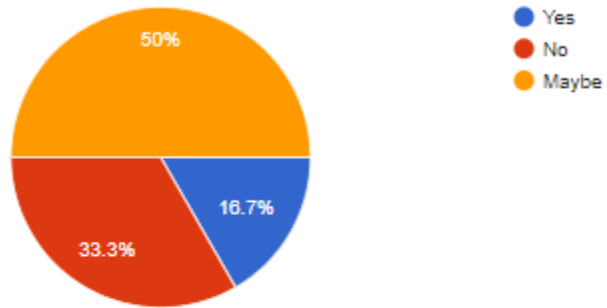
What r

6 respons

- I won't
- It was i
- Fucking
- the inst
- many tl
- puzzle

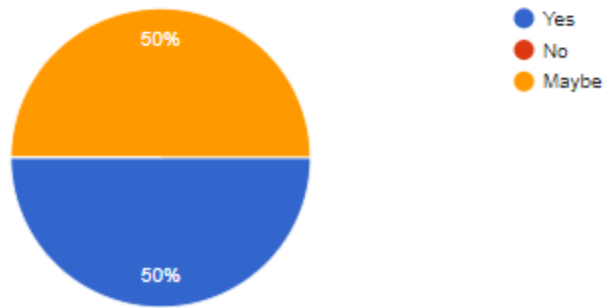
### Did you feel a sense of exploration from the app?

6 responses



### Did you enjoy using the app?

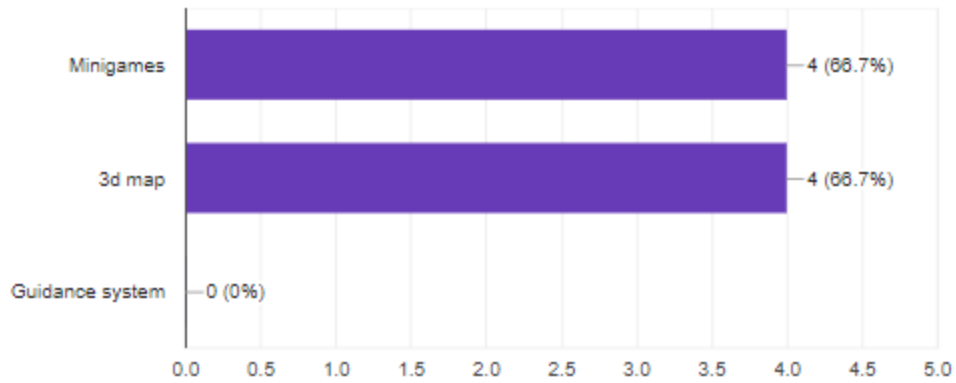
6 responses





## What features did you like in the app? Check all that apply.

6 responses



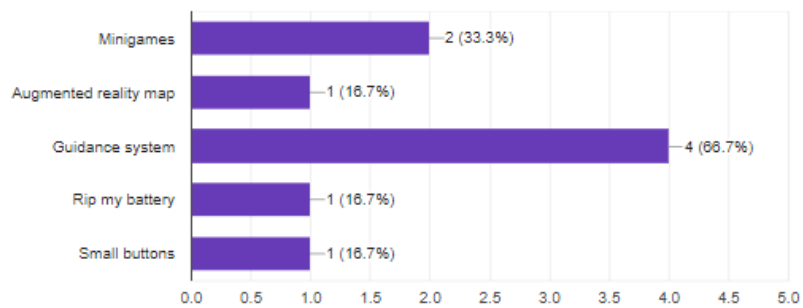
## Why?

6 responses

- I beat them, the map looked pretty nice.
- Map was pretty cool. Mini games could have been fun if less buggy
- It looked sick and cooler than Pokémon Go
- it looked good
- they are just fun
- minigames are why I would use the app

## What features did you not like in the app? Check all that apply.

6 responses



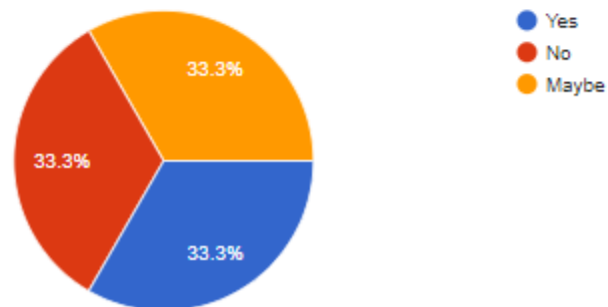
## Why?

6 responses

My battery is preciousssssss. Didn't use the guidance system at all
My hands are large and my phone made them even smaller
Jigsaw
it was not interactive enough
when changing direction and zooming in the map I always accidentally enter a game, and 3d objects that triggers the game is not too different from other buildings. and I didn't even know to touch on objects to enter game until I randomly try to click on things
I don't know what it is

## Do you feel like the app was a good way to explore Kyoto?

6 responses



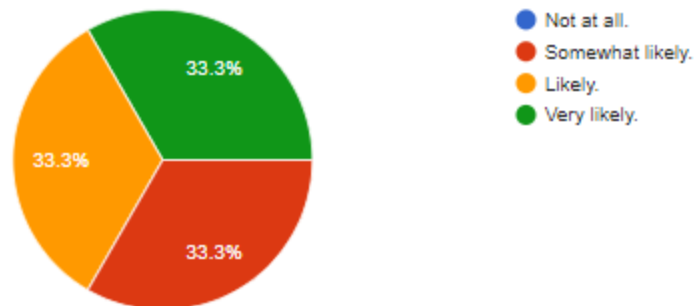
## Why?

6 responses

It only brings you to shrines, needs more than just sight seeing
Get you to want to go to the places listed in the game
Get to see very many temples
it was play outside of kyoto
the navigation and map is not very helpful or easy to use
doesn't have much guidance just games

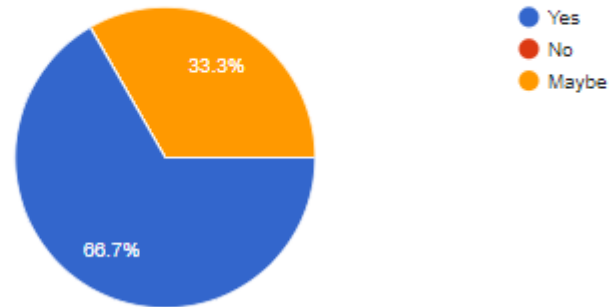
## How likely would you be to use a mobile app to help navigate around?

6 responses



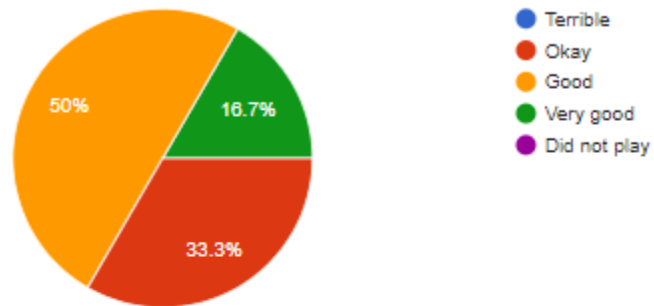
## Would you like a game that helps you explore Kyoto?

6 responses



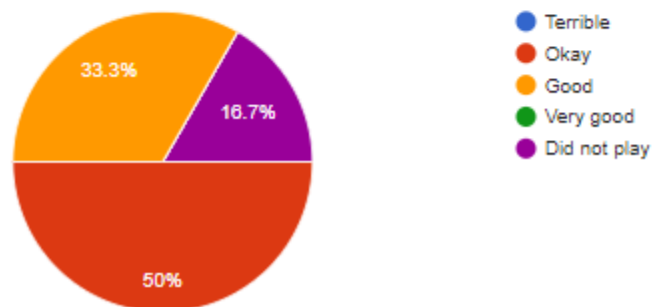
## Fushimi Inari Runner

6 responses



## Yasaka Rhythm

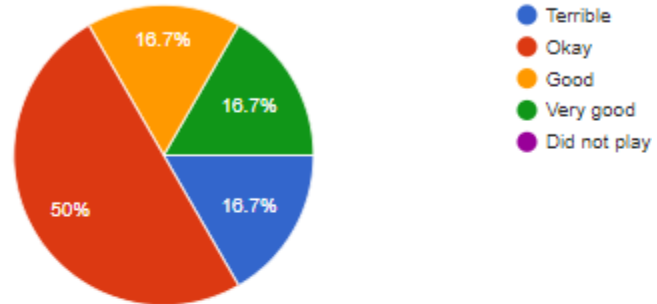
6 responses



---

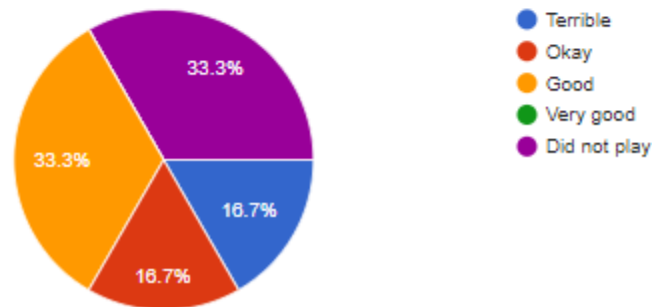
## Nijo Tower Defense

6 responses



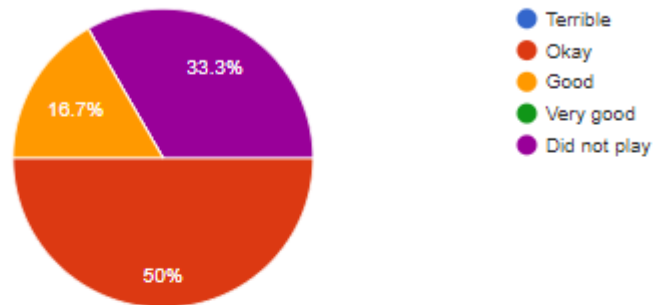
## Myoshinji Jigsaw

6 responses



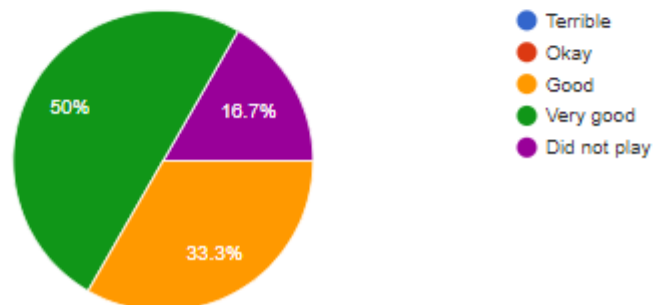
## Ryoanji Line Trace

6 responses



## Ginkakuji Unblock the Moon

6 responses



Please enter any additional comments or suggestions.

6 responses

- Less bugs :( more places that aren't just sight seeing
- Make temple run a little harder with more obstacles
- Fuck Jigsaw
- some minigames were confusing to play because there's not enough info to guide the players
- despite giving terrible rating for tower defense, it is fine if it doesn't have the bugs. also niari runners 3d animation is awesome, but but the tree model is really disturbing
- n/a

## Appendix D: Joint Ritsumeikan-WPI Puzzle Solving Robot

As a requirement for the Japan Project site, every MQP team that is hosted collaborates with the students from their respective universities to create some sort of project. For our project this year, our goal was to turn an X-Y plotter into an autonomous jigsaw puzzle solving robot. The main components of the project were to use image recognition to correctly identify each puzzle piece and to modify the existing X-Y plotter to be able to pick up the pieces and move them to their correct location. This project, however, was unable to be completed in our allotted time frame.

### What We Learned

We encountered a number of difficulties in our collaborative project, despite having the most progress out of the groups here. Because the project was more difficult than the previous projects, subdivision of the labor was necessary. We decided to work more with the hardware aspect of the project, while the Ritsumeikan students were working on the software, namely the image recognition. Because of this subdivision, we were not actually able to actively collaborate as closely with them as we might have needed. This led to a silence in communication between the team, and the inevitable failure of the project. The language barrier between us and the Ritsumeikan students also played heavily in this. The students who worked with us had trouble understanding us when we spoke to them, and often seemed afraid to directly converse with us in English. From this experience we learned that we need to work harder at overcoming difficulties to create a successful project, rather than pushing the problems aside to work on other projects.

## What We Would Do Differently

As we said, our puzzle solving robot was the closest to completion among the groups here. If we actually had some more dedicated time to all work together rather than prioritizing Gojin, we likely could have completed it.

If we could start over, we would better facilitate communication between us and the Ritsumeikan students so that our meetings would be more frequent, and we would be able to integrate our work together. We would also ensure that the project would receive an equal amount of attention as the MQP project.