A REAL-TIME STRATEGY GAME IN THE FORM OF CHARACTER

SWITCHING: SOUL SWAPPER

Thesis Presented

by

Jing Kang

to

The Department of Art + Design

Submitted in partial fulfillment of the requirements

for the degree of Master of Science in Game Science and Design

in the Graduate School of the College of Arts, Media and Design of

Northeastern University

May, 2018
# TABLE OF CONTENT

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>2</td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>2. BACKGROUND</td>
<td>3</td>
</tr>
<tr>
<td>3. APPROACH/METHODOLOGY</td>
<td>7</td>
</tr>
<tr>
<td>4. RESULTS</td>
<td>15</td>
</tr>
<tr>
<td>5. DISCUSSION</td>
<td>27</td>
</tr>
<tr>
<td>6. CONCLUSION</td>
<td>29</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>31</td>
</tr>
<tr>
<td>APPENDIX</td>
<td>33</td>
</tr>
</tbody>
</table>
ABSTRACT

Platformer, role-playing, and puzzle solving are some popular game elements. This paper introduced the project I created which combines character design and level design into one original platformer game. The paper also discusses how all the mechanics and assets were produced, and finally lays out the final results of the game.

1. INTRODUCTION

As a gamer, my passion for games has always been about the characters, the problem-solving experiences, and the art aesthetics. I enjoy playing as different characters and experiencing the variety of gameplays that different character designs provides. The game genre that I love the most is the side-scroller puzzle game. This kind of game will primarily present obstacles to players and continually require them to react with increasingly fine responses (Moore, 2012). Furthermore, I have always considered my role as an artist, and the aesthetic of a game is the most effective part to impress me. Whenever I get into a new game, the element that always determines whether I will keep playing this game is the art of the game. Thus, now as a game design student and given the chance to create a game, I am eager to put my interests in character design and level design into my Thesis Project. I want to combine character design and level design into one game and let these two elements work for each other and create an innovative gaming experience. I also want to treat this chance as a 3D art challenge for myself and create all the 3D assets in my game from scratch. In sum, I am planning to create a character based side scroller game where players have to strategically use different characters’ different abilities to overcome obstacles from the environment (Such as huge gaps, water, walls, and enemies) while also having to consider their different weaknesses.
2. BACKGROUND

2.1 Character Design

*Overwatch*, a team-based first person shooter released in 2016, is a very successful example of character design in the industry. Players of Overwatch not only love the game mechanics but also enjoy the different gameplay strategies and experiences between characters. The tricks that make Overwatch’s character design so successful are:

1. The characters in Overwatch are all very distinctive from each other. They are so different from each other that the gameplay becomes entirely different when a player switches heroes. Some heroes are small and swift, some are sneaky and deadly, and some are enormous and hard to defeat. The designers have figured out how to make each one of the heroes look and feel completely different while also making sure it's easy to glean information about them at a glance (Mulkerin, 2016). Overwatch keeps players’ interests by giving players so many gameplay choices. For instance, when a player who used to play healer suddenly decides to try playing Tank, Overwatch will become a whole new game to this player.

2. The characters are all very well balanced so that all of them have obvious strengths and weaknesses of their own. While playing with different characters, players can have different dynamic gameplay interactions with the map and with other players, but none of the heroes should feel overpowered, so that the primary determining factor for the success of the player is the skill level of that player (Hyun, H., & Kim, T., 2008). For example, Playing a healer will require players to pay more attention to teammates than other kinds of heroes, so the game experience for healers will mostly be saving teammates and trying to stay away from the enemies.

3. As mentioned above, a well-designed map also helps with good game experience for different characters. In Overwatch, the maps are designed for all hero categories. There are high grounds for snipers to crouch; there are narrow gateways for tanks to hold; there are small paths for attackers to flank.
The map helps every hero to apply their strength while also limiting their movement and mobility to some extent so that the game is well balanced.

### 2.2 Visuals and Mechanics

I envision my game to be a 2.5D platform game, and two excellent examples of this game category are *Ori and the Blind Forest* and *Limbo*. There are many things worth learning from these games:

#### 2.2.1 The visuals and 2.5D sidescroller technique

Both Ori and Limbo take place in a 3D environment, but the characters only move in a 2D style - players can only control the characters to move forward or backward. However, the characters and the cameras do not move in a pure 2D plane: they also have some movements in the z-axis - indicating depth - and these movements are designed by the level designer to let the character movements better fit the map and to let the camera movements help players see more of the map when necessary. This is the 2.5D game mechanic I aim to achieve.

To create the 2.5D sidescroller, the best way I find is to use the spline feature in *Unreal Engine*. More specifically, my approach is to let the character follow a pre-designed path, while the camera follows another pre-designed path. The camera is always at the nearest point on its own path to the character’s position and is always facing the character.

Besides the 2.5D mechanic, another thing worth learning from *Limbo* and *Ori* is that they both have great environmental art design, and they are consistent throughout the game. In Limbo, the game only has black and white color, but the great animations and gameplay give this game a dark feeling without being monotonous. This distinctive color choice is what makes this game characteristic and is how most of the
players remember this game. In the case of Ori and the Blind Forest, the art is exquisite and colorful, and it’s colorful for a reason. In this game, the consistent art style makes players feel like they are still in the same world when they pass through different game sections and the apparent color changes of the environment make players aware that they have entered a new game section. This spatial design helps players feel less lost when the map is enormous.

2.2.2 The checkpoint mechanic

For a platformer game, the checkpoint is critical in many ways. It divides the game into reasonable sections, and with smart checkpoint settings, it makes all the challenges in the game less frustrating. Bad checkpoints for this kind of design, ask players to replay large parts of the game due to their death or failure in some task, and this can often lead to anger and players giving up on the game (Kuchera, 2014). In Limbo, the game automatically sets checkpoints for players when they pass through a puzzle, while in Ori and the Blind Forest, players can set checkpoints whenever they want as long as they have the particular resources. They are both neat checkpoint mechanics that fit perfectly to the gameplay. Limbo has a pretty linear gameplay, and the automatic checkpoints divide the game in a very clear way so that players know when there will be a new puzzle. Then in Ori, because players are given multiple choices of where they can go, customized checkpoints enable players to remember their own decisions and enhance this free feeling.

For my game, I want it to give the player a relatively linear gameplay experience, so I think an automatic checkpoint like Limbo would fit best.
2.3 Level design and strategic play

When the game map coordinates perfectly with the gameplay and with different character features, it can significantly increase the playability of the game. As mentioned above, in Overwatch, many parts of their map design are character oriented. There are map spots that are designed specifically for a character or against a character. For instance, there is a bridge in the map Lijiang Tower: Garden, and it provides great opportunities for heroes who can push enemies around to make environmental kills, and meanwhile, this bridge is extremely dangerous for heroes who don't have flying abilities.

Since I plan to emphasize my different character’s various abilities and strategic play, I’ll also make my map character oriented. Also, spatial design for games have the advantage of not having to consider real-life physical rules (Christopher, 2014). For example, there can be platforms floating in mid-air that only the Ninja can reach by double jump; There can be blocks that only Sword can break by a melee attack.

Limbo’s puzzle designs are great examples of strategic play. By studying their puzzles, I summarized two kinds of strategies/puzzles that I want to implement into my game:

- Timing: This kind of obstacle requires players to do a sequence of actions in perfect timing - even a slightly late action will result in failure. This sort of obstacle is usually hard to solve, and according to Anderson (2017), if player finished it at an optimal time, usually there were either fear preceded by a rush play or greedy followed by a rush play.

- Sequenced actions: This kind of puzzles requires players to figure out what they should do and in what order to help them overcome this obstacle.
3. APPROACH/METHODOLOGY

3.1 Materials

3.1.1 3D Modeling and animation Tool - Maya

As mentioned above, I envision my game to be a 2.5D game, which means all the art assets in the game are 3D models. To create 3D models, I have several options to choose from: Maya, Zbrush, and Blender. Maya is the 3D modeling software that I know the best. It’s been known to be a much more powerful application compare to others when it comes to animation. (Pluralsight, 2014) However, It is not very capable to created detailed sculpture. Zbrush on the other hand, is famous for its sculpting capabilities, but it does not support animations. Blender provides both modeling and animating and it is free to use, but it is a brand new software to me and I do not know any of the interfaces. Considering the time I have, it would be time-consuming to learn a new modeling software; the art style I want to pursue for my game is Low-poly 3D, and the reason is low-poly models does not require high configurations to run on PC, and it can create an environment that draws player’s attention. (Couture, 2016) Thus, in my case, sculpting is not necessary. In sum, Maya is the best option for me to create my 3D models and animations.

3.1.2 Game Building Tool - Unreal Engine 4

Unreal Engine 4 (UE4) and Unity are arguably two of the most popular game engines available to the public today. (Pluralsight, 2014) I want to choose a game engine base on two aspects: A. Coding is not my strength, so the game engine that doesn’t require too much high skill coding would be a preferred choice. B. I want to use a game engine that can produce better 3D environment. Thus, I decided to use UE4 for my project, because it has visual script Blueprint which is simpler for me. Also, UE4 has higher graphic capabilities.
3.2 System Design

The system of my game will include the following modules: UI; Character; Enemy; Spline; Animation; Player actions.
- UI: This module will provide information for players on the screen, including tutorials, player’s character health, sound feedbacks base on player’s actions.

- Characters: I will create three characters attached with three distinct sets of abilities, 3D models, and animations.

- Enemy: The enemies will be able to deal damage to the player, and they will also be able to be destroyed by the player.

- Spline System: I will use the splines provided by UE4 to realize my 2.5D game style. By letting the character and camera following predesigned spline paths, the gameplay will be 2.5D.

- Animation: I will create three different State Machine for three characters and therefore let the characters be able to play correct animations in different states.

- Player Actions: In this module, I will code to realize the following functions: Player can switch between three characters by pressing “Q” “W” “E”; Player will die to enemy’s attack; Player will be able to attack and deal damage to the enemy; Player will die to designed environments such as deep water.

3.3 Character Design

According to Sloan (2015), the Core-and-Shell model is a helpful tool for character design, where the core focuses on gameplay, and the shell focuses on representation. Consulted with this theory, I created three characters who all stand out from each other, and provide a well-rounded skill set for players. They are Sword, Ninja, and Cat.

3.3.1 Sword

This character is a swordswoman. She has a sword for her melee attacks, and she has an agile appearance: simple clothes with a high ponytail. I also want her moves to look determined, which I will try to create
through animations. Sword has the ability to attack, and she is the only character in the game that has attack ability, so this is her strength and the reason for players to switch to her. She has the weakness of low mobility. Concept sketch see image 3-2.

3.3.2 Ninja

The design of this character was ability oriented. I already know that I want a character who has the ability to double jump, and it was natural that this character should be a ninja. For the design of this character, I want him to be dressed all in dark with just a shape of a ninja. The height of this ninja will be
similar to Sword for the reason of gameplay needs, and his ability to double jump will provide him high vertical mobility, and it will be his strength. Concept sketch see image 3-3.

3-3 Ninja Concept Design

3.3.3 Cat

This character will be having a height half of Sword and Ninja, so that it can pass through certain obstacles that the other two won’t be able to pass. While designing this character, there was a hesitation between a tiger or a cat. Considering this character cannot attack and has a very small figure, I found that a cat character will make more sense. The cat will have a main color of white to contrast with Ninja. It has
the ability to dash to the direction it is facing, which gives it a high horizontal mobility as its strength. Concept sketch see image 3-4.

3-4 Cat Concept Design

3.4 Level Design

As mentioned in section 2.3 (Level design and strategic play), what I aim to create through my level design is an experience where players need to perfectly sequence their actions and with some harder level chunks, the actions also need to be at perfect timing in order to pass through. I will also design enough checkpoint positions in the game. Generally they will be before every difficult level chunk. However, bad checkpoints ask players to replay large parts of the game due to their death or failure in some task, and this can lead to frustration and anger. (Kuchera, 2014) Thus, the positions of the checkpoints in my game will go through multiple iterations after playtests. I want to divide my level into chunks, where each level chunk contains one main big challenge that can be separated into several sequenced actions. Image 3-5 and 3-6 below illustrated some of my level chunk design concept.
3-5 Level Concept Design 1
4. RESULTS

4.1 Game Mechanics

4.1.1 Character Switch

Image 4-1 below shows part of Character Switch module I wrote in Blueprint where player switches to the character Cat.

![4-1 Switch Character to Cat](image)

When player switches to a character, this Blueprint module tells the rest of the game what character the player is currently on, so that they will be able to operate the skills of that particular character. Then the mesh model and animation class of this third person character switch to the correct set, and the previous set from the other character becomes invisible.

4.1.2 2.5D Gameplay

Image 4-2 ~ 4-3 below shows how the character and camera in the game are following pre-designed splines.
During every tick of seconds, when the character is able to move and receiving a “Move Right” command, these Blueprint modules keep putting the character and camera on the correct position along the spline while also managing to set their rotations so that the character and camera are always facing the right direction.

### 4.1.3 Character Abilities

Image 4-4 illustrates how Sword is able to attack.
4-4 Attack Module of Sword

When the player is near enough towards an enemy, in another word when they are colliding with the enemy’s collision box, and when the character Sword is active, the pressing of the left mouse button will apply enough damage to the enemy to destroy it.

Image 4-5 illustrates how Ninja is able to double jump.

4-5 Double Jump Module of Ninja
There is a sequel node especially for Ninja when player press “Space”. When player press “Space” for the second time when Ninja is still in the air, this character will be launched vertically with a speed, and that’s how the Ninja double jumps.

Image 4-6 illustrates how Cat is able to Dash.

Similar to Ninja’s double jump, when the character Cat is active and player press “Shift”, the character will be launched horizontally with a speed. In order to make the dash fast, the speed is set to be 15000 times of character’s normal speed. Then after the dash time is over, the Blueprint stops the character’s movement immediately, so that the character won’t fly off the map.

Image 4-7 illustrates how the game sets checkpoint.
When the player walks past - in another word, had a collision with - the checkpoint actor, this module remembers the character’s current location, and when the character died, the game will respawn the character to the most recent checkpoint location.

### 4.1.4 Level Design

Image 4-8 shows part of the tutorial in the game.

![Image 4-8](image_url)

- **Press “W” to sawp to a soul of ninja**
- **then you can double jump by pressing “Space” bar twice.**

**4-8 Tutorial**

This part is teaching the player how to switch between characters.

Image 4-9 shows the most complex level chunk in the game.
4-9 The most complex level chunk in the game

To pass through this level chunk, firstly player needs to switch to Ninja and double jump to point 1, then switch to Cat immediately before they fall down to pass through wall A. Then, the player needs to dash through point 2 since there will be bullets coming. When the player reaches point 3, he/she needs to switch to Sword to destroy the enemy. Then the player needs to switch to Ninja to double jump to point 4, then switch to Cat to avoid a few bullets from the other enemy if necessary. When the player feel the timing is right, he/she needs to switch to Sword and destroy the second enemy. Finally, the player needs to switch to Ninja and double jump to point 5 and instantly switch to Cat and dash through wall C.

4.2 Art Assets

4.2.1 3D Models and Environmental Arts

All the 3D models of this game were created in Maya, and have a consistent “low poly” art style. The characters models and environmental art assets are exhibited in image 4-10 ~ 4-13 below.
4-10 Character Sword
4-11 Character Ninja
4-12 Character Cat
4-13 Environmental Art Assets
4.2.2 Animations

To animate the characters better and give them their unique poses that can reflect their characteristics, I created different skeletons for each of them and built the animations with multiple references. Since animations are not able to be put into the paper, I will only show the skeletal mesh in this following section.

For Sword, I create the walk cycle in a way that makes her move looks determined and brave. For Ninja, while creating Ninja’s animation, I watched some Japanese cartoons to look for references, and finally decided on a running cycle where Ninja’s both arms are in the back. For Cat, since Cat is a quadruped animal, the bones are different from human, so the skeletons of Cat have different settings and controls with the other two characters. More specifically, there is one more joint on every leg of Cat compared to human, and the IK handles on those legs each link to not three, but four joints.

In the game engine, I created a State Machine so that the correct animation is played at the right time according to player’s actions. See image 4-15 ~ 4-16 below.
4-15 Animation State Machine for Sword

4-16 Player States Checking Module
4.2.3 Sound Credit

All the sound effects in this game project were downloaded from the website Freesound, and are eligible for non-commercial use.

5. DISCUSSION

The most valuable knowledge I gained from making this project is the whole workflow of making a game. From designing to coding to animating, I got the chance to get my hands on every step in game-making; I had a deeper look into the while game-making process. The reason I say this is that although I have made a lot of games in the past two years as class projects, I have always worked in groups, and my role has always been an artist or designer. I have only experienced the game making process from one point of view at a time, and there were problems in the games that I did not need to worry about or I did not even know they exist because they were not my responsibilities. However, while making this project, I need to consider all the perspectives when I’m designing a feature or making decisions: Will this affect my system structure? Will I need to change my character for this? Do I need to re-make the animations? Furthermore, whenever there is a problem, I need to fix it myself and it needs to be fixed as soon as possible. Luckily, I successfully overcame all the problems and bugs on the way, and by solving them, I now have a better problem-solving capability.

Another thing I learned from this project that will benefit me in the future is how to make animations for games. I took an animation class in Fall 2017, and I learn so many valuable animating techniques that I have always wanted to learn, and I applied almost everything I learned from that class into this game. However, that class was not game-related, so it took me quite a lot effort to learn how to create suitable
3D models and animations for UE4. That process was interesting and I enjoyed learning it, and now I’m more confident when it comes to create game assets. It also helped me improve my self-learning ability.

One thing I’m very proud of myself in the making this game is that I tried my best to give personalities to my characters through animations. I created three different skeletons for my characters and each of them has their distinctive animations that fit their characteristics. One thing I could do (and many other games did) is to create one skeleton and animation set and use the same one for all the characters, but it would go against my character design intention. One main purpose of my project is to design different characters, so I wish to do everything I can to make my characters stand out, so I think the time I spent for making all the animations are all well spent.

My game also has many limitations, and below are the features that I think could make the game better but could not be built into the game due to the lack of time:

- Range attack for players: At the very beginning stage of my design, I was going to create a fourth character with range attack ability. I think with another attack option, players can have even more strategies to choose from.

- More enemy types: I also designed other types of enemies. Such as the ones that can move in a specific pattern; the ones that can kill the player with one hit. I think if I can implement these enemies in the game, the difficulty of the game can be highly increased.

- Longer levels: As the design concepts are shown in section 3.4 (Level Design), there are many level chunks that I designed could not be developed into the game. I wish to continue building my levels and playtest the rest of the designs in the future.

- Jump animations: Right now, all three jump animations in my game are basically the character bending their knees in the air. In many great games, character jump animations have three
different states: Start jumping; while jumping; and end jumping, and there are three animations for these states, so that the character jumping will be more realistic and natural. The setting for this kind of animation is more complicated than what I have now, and I hope I can create something like this in my game in the future.

- Death feedback: For now, when the character dies, there will be a 1-second delay before the character respawns and a sound of game fail will be played as the death feedback. What I could do to make it better is to give each character a death animation. I could not make them into the game due to the limit of time.

6. CONCLUSION

In the past semester, I successfully completed the Soul Swapper game design and construction.

In the art part, I designed three characters for the game that players can swap between: Sword, Ninja, and Cat. I gave them different abilities so that they each have strength and weakness and make sure the gameplay is well balanced. I also created all the art assets. For 3D models, I made three meshes for the characters and five tree models, two stone models, one grass model, one wall model, one house model, and one bush model for the environmental arts. For character animations, I created three skeletons for the meshes and three to four animation sequence for each of the character, that is ten animation sequences in total.

In the technical part, I realized 2.5D gameplay using Spline system from Blueprint, and all the other mechanic functions were also built from Blueprint nodes. Including real-time character switch, checkpoint system, character attack, enemy attack, damage apply, character death/respawn, and environmental death for the character.
In the level design part, I built one gameplay tutorial and three level chunks into the game with a total of 5 - 10 minutes gameplay time. All the level chunks require players to switch between characters and two of them provide more than one problem-solving approach. The three level chunks have difficulties from easy to hard, and they require players to have better and better performance to pass through.

I conducted five playtest sessions and a total of seven players played this game, and below are the information I gain from the playtesters:

- The level chunks worked well as I intended. All the playtesters find the beginning part easy to pass, and those are the parts where I designed to be simple and introductive. The parts where I intended to give players some trouble and spend some time before they get through, turns out to be successful at holding players’ paces.

- As I mentioned before, I wanted players to have the whole scene in mind and the needed action sequences in mind before they start moving in one level chunk, because only then player’s actions can be smooth and at a better timing. During the playtest, I observed that three players were mumbling the actions such as “Double jump, then switch, then shift...” while playing.

- One player reported that the control schemes of the game have some problems. Such as the Dash ability for cat is not intuitive enough for new players to learn. I noticed that some player had trouble learning Dash while some other players feel that the control for Dash was very natural for them. Due to coding problems, I could not give the Dash ability another control scheme for now, but the problem is definitely worth thinking in the future.

Although my project still has many flaws and limitations as I discussed in section 5, I feel that the outcome fulfilled my initial intention of combine character design with level design.
ACKNOWLEDGEMENTS

I would like to thank Professor Curry and Professor Barney for their advice throughout this project, as well as my friends Filiz Durgun, Suraj Guti, Shubham Gupta, and Vaishnaviben Shah for their playtesting. I would also like to thank Zhaokang Xu for his assistance in fixing coding bugs.

REFERENCES


APPENDIX

Soul Swapper Design Document

1 Game Story/Background

The story of the game happens in an ancient Chinese village. The girl with magical powers - the swordsman character in the game - was burdened with the task of restoring the lost magic stone of the village. So she is now on the way towards the adventures. With her magic powers, she has the help from a soul of the Ninja and a soul of the Cat to get her through difficult situations.

2 Game Objectives

1. Players need to reach the end of the game.

2. Players try to overcome obstacles by swapping between characters and using their special abilities.

3. The levels in this game will be clearly divided into small level chunks. In order to pass through a level chunk, players have to have the whole scene in mind and come up with a series of actions and execute them in perfect timing.

3 Characters

Character Sword

Ability: Melee attack. One attack destroys one enemy.

Character Ninja

Ability: Double jump. Can reach higher ground than other two characters.

Character Cat
Ability: Horizontal dash & much smaller figure. Can go through areas where the other two characters can’t.

4 Mechanics
Controls
A: Move left
D: Move right
Q: Switch character to Sword
W: Switch character to Ninja
E: Switch character to Cat
Shift: Horizontal dash (Only active when the player is using Cat)
Space: Jump (Press twice double jump when the player is using Ninja)
Left mouse button: Melee attack (Only active when the player is using Sword)

5 Enemy and Damage
Enemies have attack range and fixed attack frequency; player needs to observe first, then try to avoid the bullets from the enemies and get close enough to destroy them.

Player has 3 Hearts, each bullet from the enemy will take one Heart. When the player runs out of Hearts, they will respawn to the most recent checkpoint.

Death caused by environments (For example, fall into the river) will cost all of the Hearts and player will respawn immediately.
6 UI

Main Menu:

Appears when player press Esc.

Contains all instructions; an Exit button; and a resume button.

Hearts:

On the top left of the screen, a heart shape image and a number indicate players current Heart number.

Tutorial:

At the beginning of the game, as player proceeds in the level, instructions will show up and help player get familiar with the controls.

Sound feedbacks:

When the player attacks with Sword, a sound of sword swooshing in the air will play and let the player know they have attacked.

When the player wants to switch character in places where only Cat can fit in, a sound of error will play to let the player know is action is disabled at this place.

When the player dies, a sound of game ending will play to let the player know the character will be respawning.