Software Requirements Specification

for

Chrono Split

Version 1.1

Prepared by Team Chrono

Team Members: Darrin Perry, Omer Yurdabakan, Alex Hendrickson, and Levi George

October 6th, 2020

Table of Contents

Table of Contents ii

Revision History iii

1. Introduction 1

1. Purpose 1

2. Document Conventions 1

3. Intended Audience and Reading Suggestions 1

4. Project Scope 1

5. References 1

2. Overall Description 2

1. Product Perspective 2

2. Product Features 2

3. User Classes and Characteristics 2

4. Operating Environment 2

5. Design and Implementation Constraints 2

6. User Documentation 3

7. Assumptions and Dependencies 3

3. System Features 3

1. Character Selection 3

2. Character Ability Trees 3

3. Stats and Bonuses 4

4. Buffs and De-buffs 4

5. Multiple Levels 5

4. External Interface Requirements 5

1. User Interfaces 5

2. Hardware Interfaces 5

3. Software Interfaces 6

4. Communications Interfaces 6

5. Other Nonfunctional Requirements 6

1. Performance Requirements 6

2. Safety Requirements 6

3. Security Requirements 6

4. Software Quality Attributes 6

6. Other Requirements 7

Appendix: Glossary 8

Nav mesh- otherwise known as a navigational mesh used by AI controllers to traverse levels and receive stimuli. 8

Box tracing- In unreal 4, box tracing, otherwise known as a ray trace is used to determine items in a path and attributes of said items. 8

Speed Run- The competition of completing a game as fast as possible, usually timed and done by people who have optimized pathing of gameplay of certain games. 8

Skins- Aesthetic addons in games that change the character appearance, usually thematic but can also be a simple re-color of the base model. 8

Buffs/De-Buffs- Buffs are anything in the game that increases a player's stats, while the opposite is true of de-buffs. 8

Farm- Farming in a game is referred to the ability to fight low level creatures to increase gold/stats/etc. 8

PVP- A mode in a game where a human controlled character fights against another human controlled character. 8

Multi-player- game mode where multiple human controlled characters interact in a server environment, can be cooperative or competitive. 8

Hit-box – An area on the character/AI/game environment that can be damaged and will register stimuli. 8

Skill Floor – The lowest possible skill needed to play a game with success. 8

Skill Ceiling – The highest possible skill needed to play a game successfully. 8

Crowd Control – Any ability that hinders the movement of a player or AI I.e stuns, slows, etc. 8

Game Mechanics – This includes but not limited to character abilities, rules of the game, how the player interacts with the environment. 8

Appendix: Models 9

Appendix: Prototype 10

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Omer/Levi | 10/06/20 | Initial Draft | 1.0 |
| Darrin/Omer/Levi/Alex | 10/08/20 | Updated language and scope | 1.1 |

# Introduction

## Purpose

This SRS is for a video game which is intended to be played for recreational purposes. The user will play as one of at least three playable characters. The gameplay mechanics will depend on the character chosen. After selecting the user will engage in arcade-style 2D gameplay clearing several levels as quickly and skillfully as possible. The game will be a speed run style with a twenty-minute timer that the player will have to race against. The goal of this document is to establish the requirements needed to complete a functioning game, while keeping the integrity of the vision.

## Document Conventions

N/A

## Intended Audience and Reading Suggestions

This document should be read by the developers, customers, and Faculty advisor. The developers should read every section to ensure that there is an understanding for the project. The main sections for the customers and faculty advisor to review are section 1.4 Project Scope, 2.7 Assumptions, and section 3. Features.

## Project Scope

The primary purpose of this game is to provide entertainment to the players. While also providing a tactile challenge in terms of controls and a speed-run aspect that tests the user’s capability to trim time off their gameplay. The scope will be entirely within the game system. No external software will be included nor will this game use client-server architecture to link games for multiplayer.

## References

Character Assets

<https://www.gameart2d.com/freebies.html>

<https://craftpix.net/freebies/>

<https://itch.io/game-assets/free>

Templates

<https://pdfslide.net/documents/astropolis-srs-1.html>

Game Engine Documentation

<https://docs.unrealengine.com/en-US/index.html>

# Overall Description

## Product Perspective

Raid will be an arcade-style game where users will select a character and attempt to play through the game as quickly as possible to beat the timer and defeat the boss in the least amount of time.

## Product Features

1. The game will consist of three playable characters, if the player defeats the boss in the time limit then they unlock a new character.
2. It will be equipped with a user login for player profiles that keeps track of success of the player. As their experience rises the game will become increasingly more difficult.
3. Players will be able to utilize skins through in game purchases.
4. An online leaderboard will display users and their best times by character in a ranking.
5. The game will be able to accommodate different skill levels by use of different starting characters without the use of a difficulty setting.

## User Classes and Characteristics

Users will most likely be between the ages of 13-35 who regularly play games. As the game is strategy based, but with high skill expression, it will be able to draw players from various genres of games. The game is easy enough for a beginning gamer, but with enough difficulty for a seasoned veteran to enjoy. The genre of the game lends itself to predominantly male users, but with the right character base, a female audience could be found.

## Operating Environment

The game will need a minimum of windows 7/or MacOS equivalent and above. A dedicated graphics card is preferred, but not necessary. A game controller is supported, but optional if a keyboard is available.

## Design and Implementation Constraints

The design constraints consist of optimized coding and graphics to save on memory and hard drive usage and GPU overheating.

The game is coded in C++ and is innately constrained by the language. This is partly remedied by blueprint implementation in the game engine.

This game is also constrained by the 2D paper environment. The environment does not lend itself to nav meshes, so a new way of receiving information for complex AI will need to be implemented to remedy this factor. One possible solution is to use box tracing.

## User Documentation

The player will be supplied with character backstory and ability lists. They will also be supplied with a general overview after character select. A tutorial level will exist to familiarize the player with the controls, but timer starts at the first portal jump to a new level.

## Assumptions and Dependencies

Player has a Windows Operating System enabled computer with preferably modern hardware.

Mouse and keyboard.

# System Features

## Character Selection

3.1.1 Description and Priority

The players should be capable of choosing a character that they wish to play through the game as. There should be at least three characters to play with the first version that is completed (this is high priority).

3.1.2 Stimulus/Response Sequences

When the player starts the game, it should display a character selection screen displaying each of the character. The characters should display their description and key abilities that are unlocked as the player unlocks them.

3.1.3 Functional Requirements

REQ-1: A The player cannot proceed without selecting a character

REQ-2: A Hovering over each character will display the corresponding key abilities

## Character Ability Trees

3.2.1 Description and Priority

Players can earn coins from killing monsters and beating levels. These coins can be used to purchase upgrades and abilities from the tree. (This is a high-priority feature).

3.2.2 Stimulus/Response Sequences

On pressing a pause button, the user will have the option to view their skill tree. The user can click on any of the buttons to activate the skill and attain it for use.

3.2.3 Functional Requirements

REQ-1: The skill tree will have a different number of paths depending on the character.

REQ-2: Each skill tree bonus will be based on stat improvements

REQ-3: Selecting a skill bonus will consume gold

REQ-4: Each skill can only be purchased once

REQ-5: In order to buy a bonus, the ones before it must have been selected.

REQ-6: The player can elect to keep choices by clicking a “commit changes” button.

REQ-7: Exiting the upgrade menu without clicking the button will leave changes unsaved, no gold will be used.

## Stats and Bonuses

3.3.1 Description and Priority

Each character will have base stats such as attack power, defense, speed, agility, etc. These will influence how much damage the character does on attack and other special events. This is a high priority feature.

3.3.2 Stimulus /Response Sequences

During an attack, these stats will be factored to determine if damage is applied and how much is to be applied.

3.3.3 Functional Requirements

REQ-1: Increases in the attack stat will cause attacks to inflict more damage.

REQ-2: Increases in the defense stat will cause incoming attacks to inflict less damage

REQ-3: Increases in agility will decrease the chance that a given attack will hit

REQ-4: Increases in speed will increase the movement speed of the character

REQ-5: Agility will be checked first, then translated to a percentage, depending on this percentage an attack may whiff or hit.

REQ-6: Total attack damage inflicted will be calculated first, then enemy defense will be factored in to determine actual damage.

REQ-7: Speed will always be applied changing only on de-buff.

## Buffs and De-buffs

3.4.1 Description and Priority

The buffs and de-buffs are temporary stat bonuses that may be related to attacks performed by either the player or enemies. They can increase attack, decrease attack, among other things.

3.4.2 Stimulus/Response Sequences

Upon the use of certain abilities or basic attacks by specific enemies a de-buff or buff may be applied, which will temporarily decrease a given stat for the player character.

3.4.3 Functional Requirements

REQ-1: Attacks by certain enemies need to cause temporary stat decreases

REQ-2: Certain abilities should cause attacks to de-buff enemies

REQ-3: Certain abilities should cause the player character to be buffed

## Multiple Levels

3.1.1 Description and Priority

The game will consist of at least four levels connected by a hub world. Each level will thematically match each of the first three characters. The player needs to complete only one, the boss level, in order to beat the game. The rest can be played by the user to earn gold for stat bonuses to make subsequent levels (and the boss) easier to play through. One dragon enemy character will spawn randomly somewhere in one of the levels to facilitate player interaction with all levels since the dragon will be worth a large amount of gold. This is a high-level priority.

3.1.2 Stimulus/Response Sequences

Each level will be hard coded into the game. Upon spawn the player will be in the hub world and can walk into portals that take the player to each level.

3.1.3 Functional Requirements

REQ-1: The portal will teleport a player directly to the start of the respective level.

REQ-2: The player may not return until the level is complete

REQ-3: At the end of the level the player will be teleported back to the hub world where the original portal will remain open should the player decide to farm that specific level.

# External Interface Requirements

## User Interfaces

Each character in the game will have their own custom aspects for the UI, but each of them will display health, coins collected, time left, and major ability cooldowns. A pause screen will allow the user to manage their skill tree based from coins collected, but the pause function will be only accessible for a limited amount of times. A portal level will be available for them to access the same skill tree, but the timer will not be stopped.

## Hardware Interfaces

The two major interfaces will be a keyboard and mouse, but a controller will also be supported. Low end computers will be able to play the game on a low-resolution setting.

## Software Interfaces

The game will have no interfaces with other software, aside from a possible future database interface to store high scores.

## Communications Interfaces

This is currently a single player game, but some information will be stored on a server to populate leaderboards. A possible PVP or multi-player aspect could be added in the future, but currently not in the scope of this project.

# Other Nonfunctional Requirements

## Performance Requirements

The game must be able to run at 50-60 frames per second on Windows platform OS.  At this frame rate, the game will remain constant and playable. It will not be technically demanding and able to run on lower end computers. A settings menu will be implemented for the user to lower resolution settings to optimize their computer setup for optimal gameplay.

## Safety Requirements

The main safety concern is seizure from flashing lights and sound which will be kept to a minimum. As the game is set for 20-minute increments, then the risk of eye strain is limited.

## Security Requirements

Security will not be a concern for this project since there is no sensitive data being stored. No multi-player is available, so the risk of security attack is low.

## Software Quality Attributes

The usability of the game is of the upmost importance, also hit-box accuracy and time tracking will be a large focus. The game rewards precision of attacks and button controls. Collision needs to be extremely accurate because of various game mechanics and feel of the game.

**5. Intuitive Gameplay/Interesting Game Mechanics**

The game needs to have a low skill floor, but a high skill ceiling. The game needs to be intuitive enough that a beginner can understand, yet complex enough that a veteran will enjoy playing it. Controls need to be placed at a reasonable spot to avoid issues with inability to express skill in the game. The crowd control needs to be visible and understandable to the player. The mechanics themselves need to be interesting and fun to use.

# Other Requirements

None

Appendix: Glossary

Nav mesh- otherwise known as a navigational mesh used by AI controllers to traverse levels and receive stimuli.

Box tracing- In unreal 4, box tracing, otherwise known as a ray trace is used to determine items in a path and attributes of said items.

Speed Run- The competition of completing a game as fast as possible, usually timed and done by people who have optimized pathing of gameplay of certain games.

Skins- Aesthetic addons in games that change the character appearance, usually thematic but can also be a simple re-color of the base model.

Buffs/De-Buffs- Buffs are anything in the game that increases a player's stats, while the opposite is true of de-buffs.

Farm- Farming in a game is referred to the ability to fight low level creatures to increase gold/stats/etc.

PVP- A mode in a game where a human controlled character fights against another human controlled character.

Multi-player- game mode where multiple human controlled characters interact in a server environment, can be cooperative or competitive.

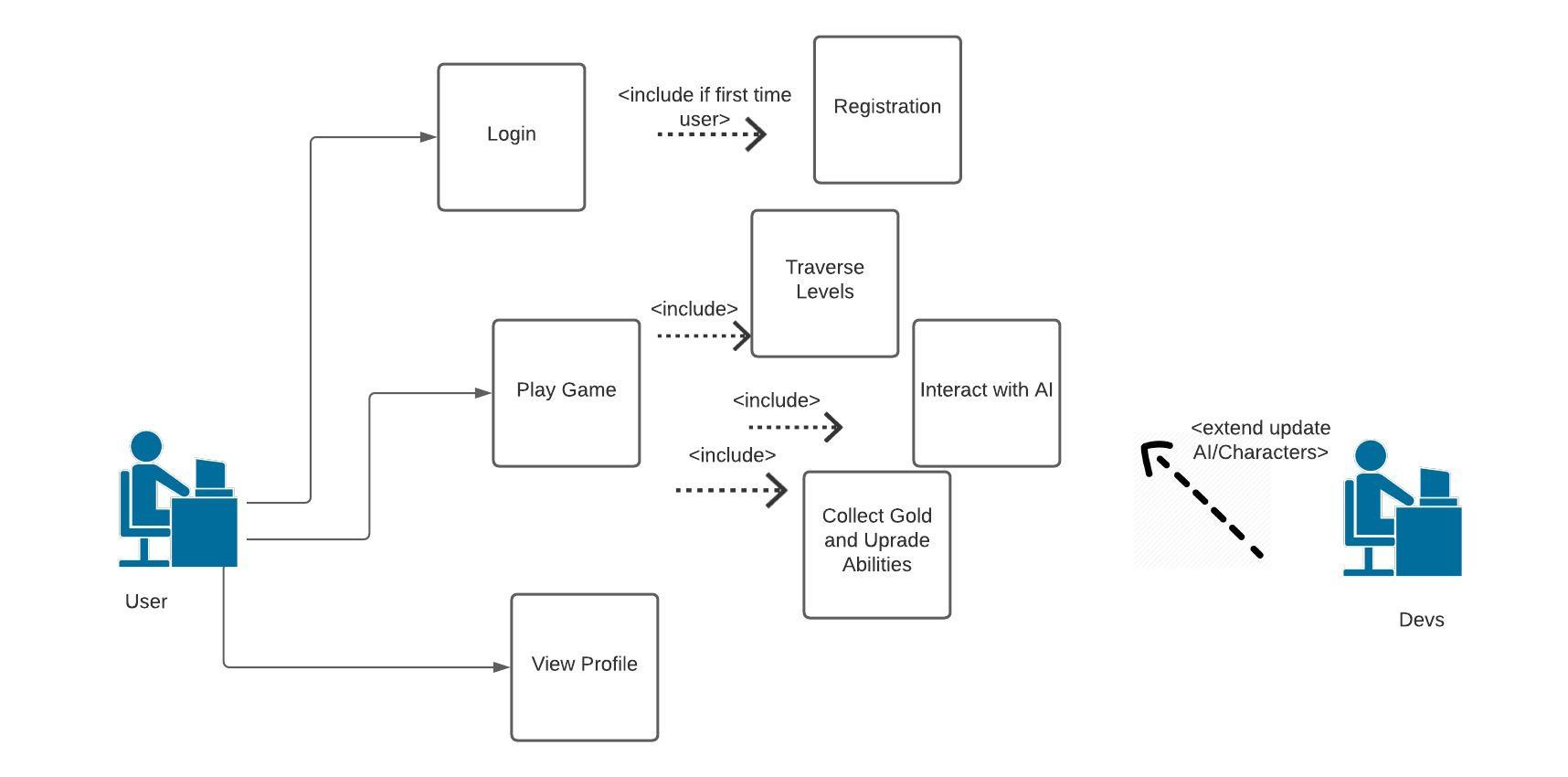
Hit-box – An area on the character/AI/game environment that can be damaged and will register stimuli.

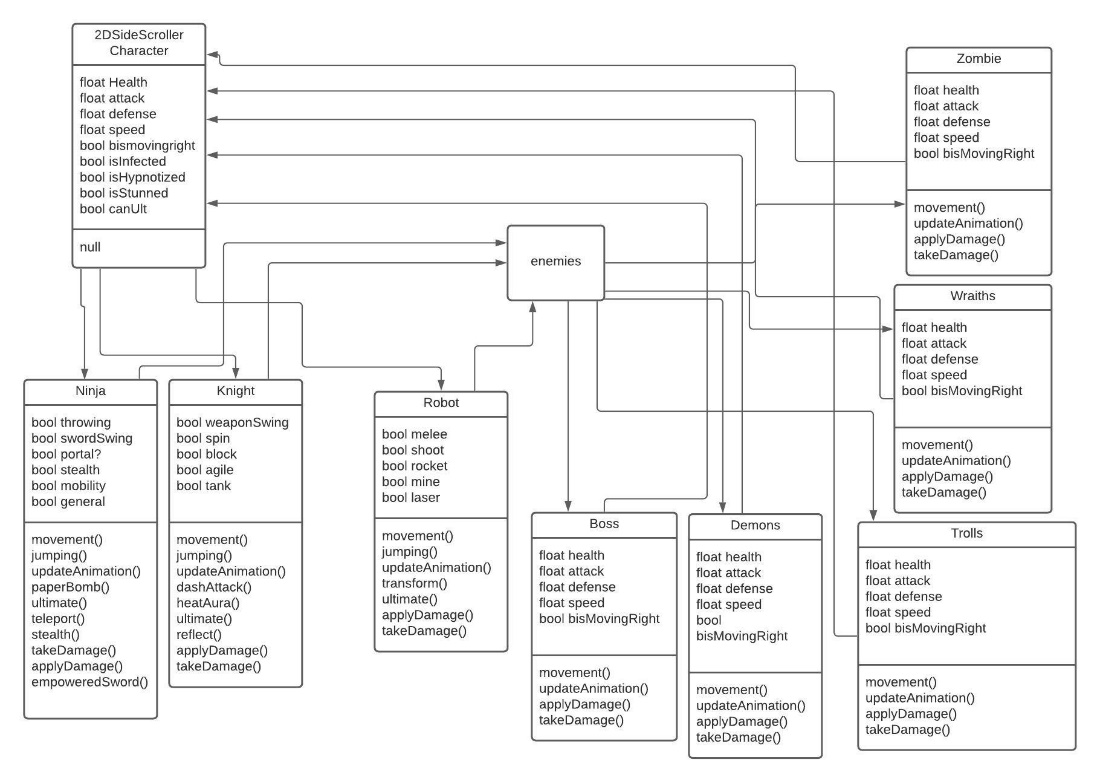
Skill Floor – The lowest possible skill needed to play a game with success.

Skill Ceiling – The highest possible skill needed to play a game successfully.

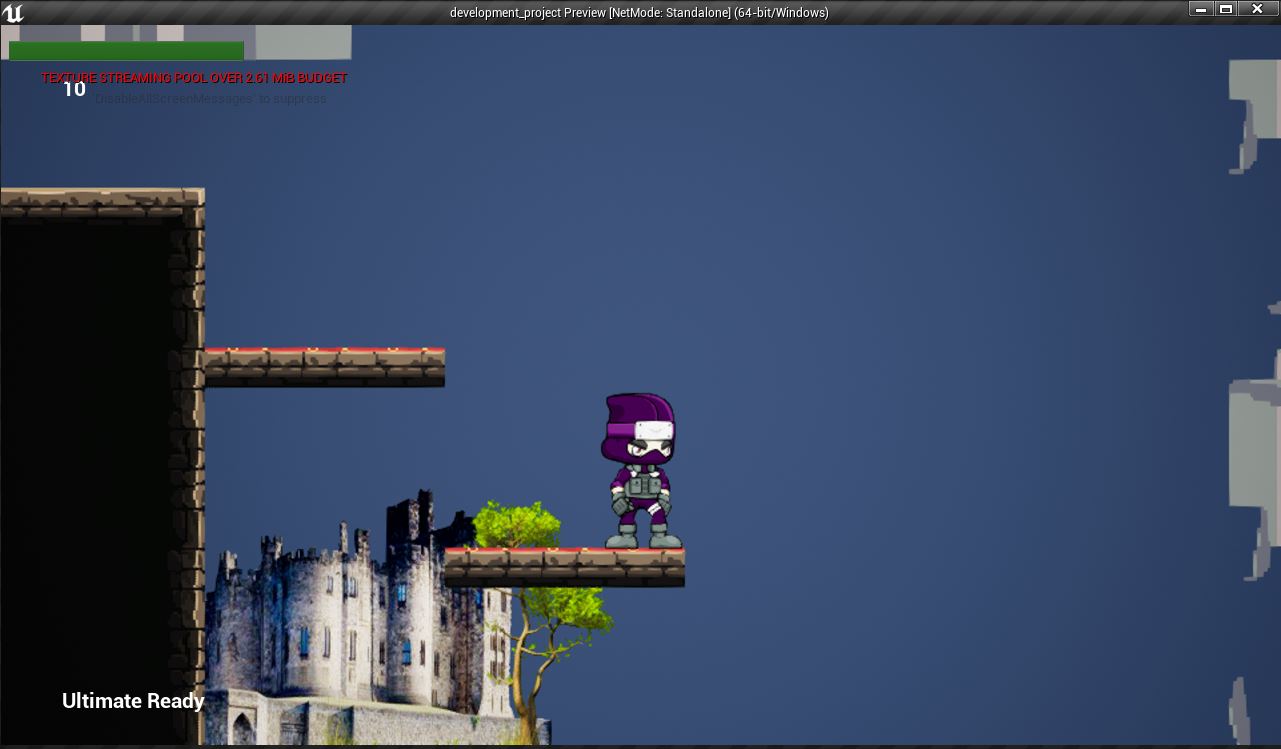
Crowd Control – Any ability that hinders the movement of a player or AI I.e stuns, slows, etc.

Game Mechanics – This includes but not limited to character abilities, rules of the game, how the player interacts with the environment.

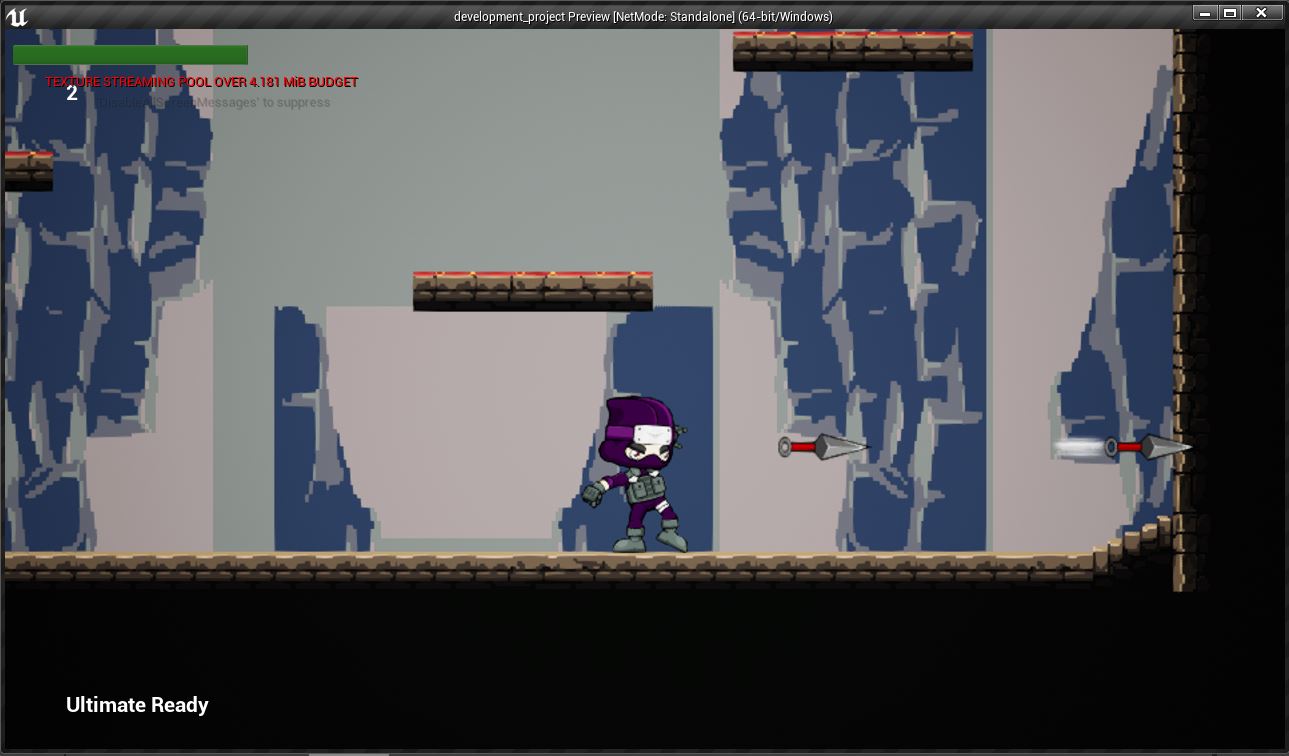
Appendix: Models



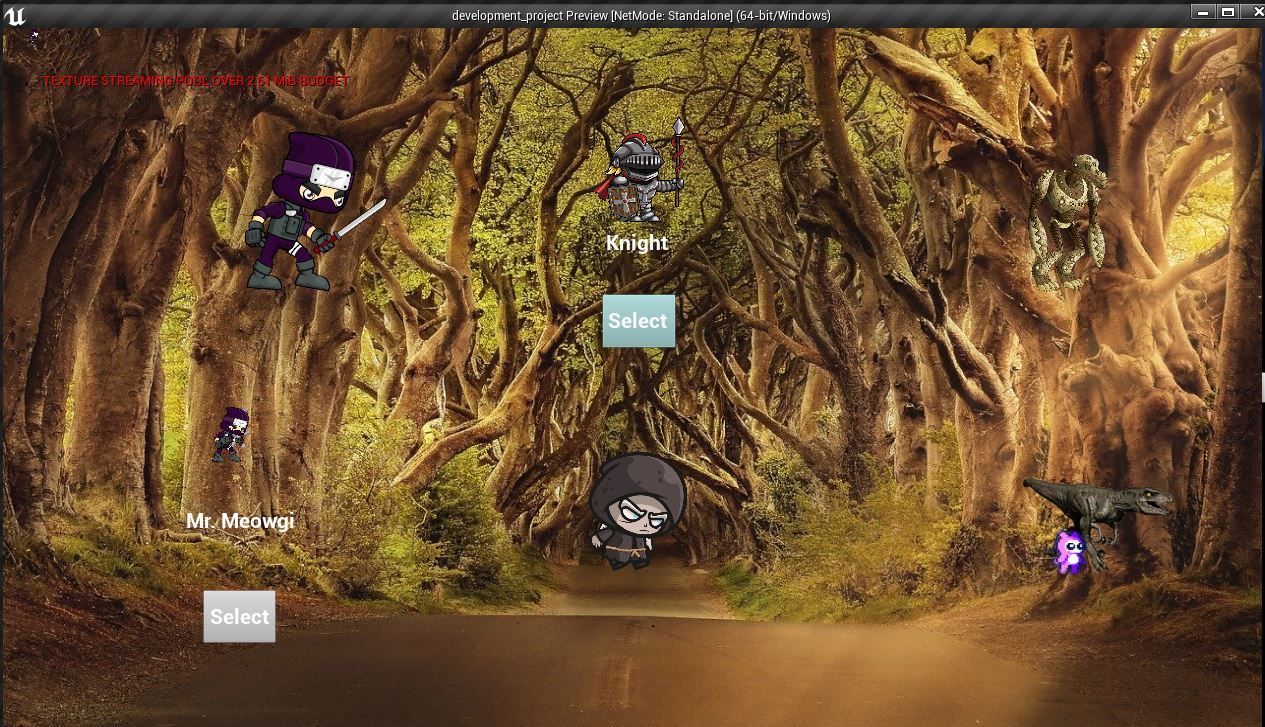
Appendix: Prototype

Ninja Character and Level Design Prototype

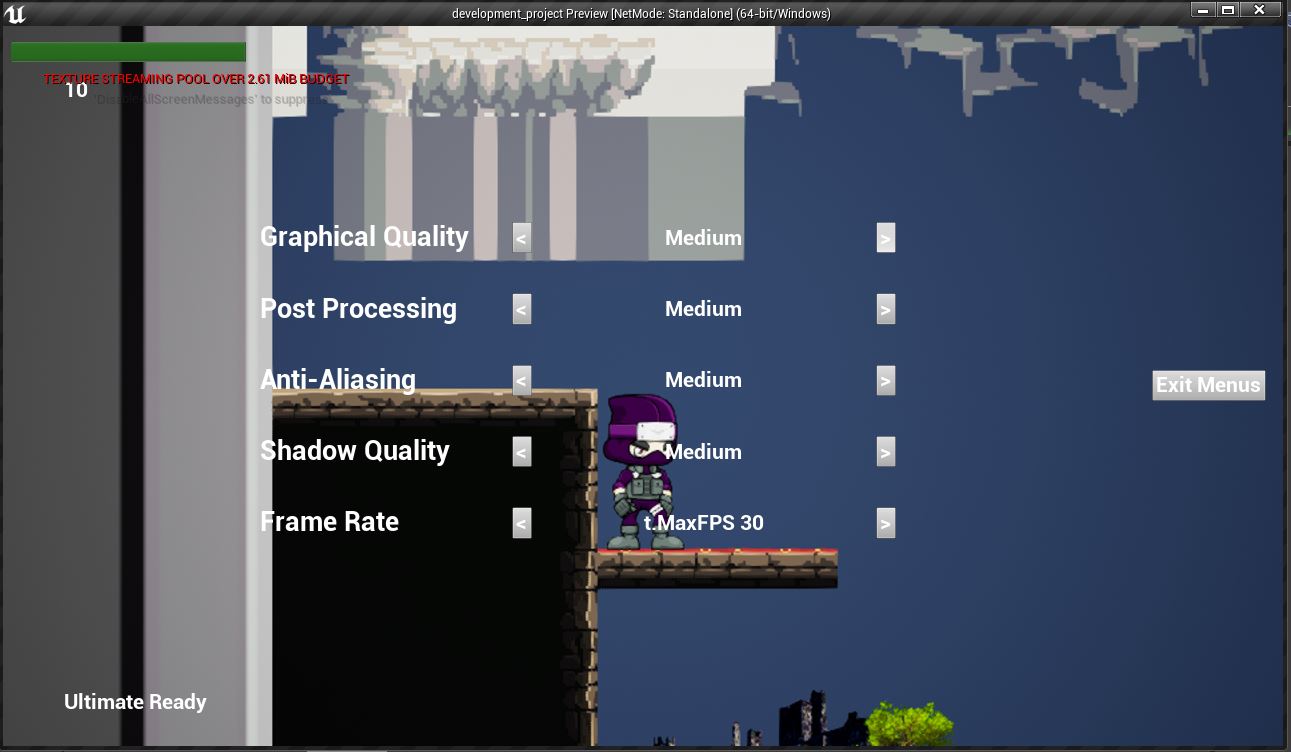
Ninja Throwing Kunai Ability



Character Selection Screen Prototype



Game Options Menu Overlay Screen



Portal Creation

