**Department of Electrical and Computer Engineering**

**College of Engineering, Trine University, Angola, Indiana**

**Shrek: The Wizard’s Revenge**

Game Design Document

Developed by:

Matthew Graber

Dr. Kevin Wolverton

December 4th, 2023

# Table of Contents

[Table of Contents 1](#_Toc2118778309)

[Original One-Page 2](#_Toc1270291802)

[Summary of Original Design 3](#_Toc798525434)

[Plot Summary 4](#_Toc837372003)

[Mechanics summary 4](#_Toc57150153)

[Goal 4](#_Toc785980479)

[Design Intent 4](#_Toc735403064)

[Gameplay and Mechanics 4](#_Toc754468182)

[Basic Combat Mechanics 5](#_Toc673429526)

[Sailing and Ship Combat 5](#_Toc485674141)

[Magic Items 5](#_Toc1759675860)

[Enemies and enemy mechanics 5](#_Toc912578372)

[Final Design 5](#_Toc254990773)

[Core Gameplay 5](#_Toc704168)

[Game End 6](#_Toc1334687232)

[Sailing 6](#_Toc2037068556)

[Locations 6](#_Toc733527849)

[Starting Island 6](#_Toc675157954)

[Boss Island 6](#_Toc940460755)

[Random Islands 6](#_Toc839784043)

[Magic Items 7](#_Toc1106945728)

[Enemy Design 7](#_Toc309268267)

[Enemy Manager 7](#_Toc1581062652)

[Enemy Types 7](#_Toc82731741)

[Berserker 7](#_Toc435513378)

[Gunslinger 7](#_Toc1354655757)

[Shrek, the Pirate Lord 7](#_Toc312149557)

# Original One-Page

You are a wizard. You have been captured by pirates. They’ve taken all of your magic items from you, rendering you almost helpless. Unfortunately for them, one of those magic items is the Warp Sword, which you can summon at will and use to escape the prison they’ve put you in.

You are no longer just a wizard. You’re an angry wizard. And you have a magic sword.

Goal

You win the game by finding and defeating the pirate king. You lose the game if you die.

Core Gameplay Loop

The game begins with you breaking out of pirate jail and slaughtering the pirates that inhabit the island pirate jail is on. You then must take the pirates’ ship and sail away. You then use the ship to travel between the different islands. One of the islands is the lair of the pirate lord, who you must defeat to win the game. The others each have one of your magic items, guarded by pirates.

Over time the enemies will grow stronger, meaning the longer you wait to face the final boss the more powerful he becomes. Alternatively, I may instead add a countdown where after some amount of time you lose the game.

Basic Mechanics

This game uses basic first-person FPS mechanics, but with primarily melee combat. Ideally the combat should be fast-paced and feel acrobatic – staying still is a good way to die, but quick thinking and good tactics will get you through it. The other important mechanic is sailing between islands. Depending on time constraints, it will either be a fast-travel system or actually controlling a ship as it sails around islands.

There should be at least three different kinds of enemies, plus a boss enemy.

Stretch goals

If time allows, there are additional mechanics that I’ll add to the game. The first is real sailing as opposed to fast-travel sailing, with enemy encounters. Second, randomization of island placement. Third, randomization of additional magic items and extra magic items. Finally, larger enemy diversification.

# Summary of Original Design

## Plot Summary

You are a wizard who has been captured by pirates. These Pirates have taken all of your magic items and thrown you into a prison. Fortunately, one of your magic items is a sword with teleportation powers. You can summon the sword to yourself and then use it to teleport out of the prison. You then steal a ship from the pirates remaining on the island and begin your quest for revenge.

## Mechanics summary

Shrek: The Wizard’s Revenge (henceforth known as STWR) is a 3d first-person action game that focuses primarily on melee combat. Additionally, there will be some seafaring exploration and combat as you search for the islands that hold your magic items. It will also contain some rogue-like elements in the form of randomly generated island locations and items.

## Goal

You win the game by finding and defeating the Pirate Lord. You lose the game if you die.

## Design Intent

The game is intended to create fast-paced and strategic combat, with a meta-strategy about deciding what items to collect and when to fight the final boss. The game is intended to downplay “safe,” but boring strategies, such as camping, farming, or similar low-risk behaviors.

The game is intended to be replayable through the use of randomization. It is also intended to be fairly challenging, so that players likely won’t win on their first run.

## Gameplay and Mechanics

### Basic Combat Mechanics

The Game will feature basic first-person shooter mechanics, except that you will be fighting primarily with a sword. The only item you are guaranteed to have each game is the Warp Sword, which you can use to teleport during combat.

You will have two resources during combat: Health and Manna. If you run out of health, you die. Mana is used to power your magic items. Mana regenerates when you damage enemies with melee attacks, but health only regenerates by using consumable items. If you have no mana, you fuel your magical abilities with your health instead.

### Sailing and Ship Combat

You will be able to sail a boat around. You won’t be able to use your ships cannons until you have your unseen servant, but enemy ships will be able to shoot at you. Your ship will have a limited amount of health, and if it is destroyed it will begin sinking.

### Magic Items

The game will contain several unique magic items which you can acquire by traveling to different islands and fighting enemies there. You will be able to use these magic items by expending mana.

## Enemies and enemy mechanics

Every enemy will have a couple attributes: Their hit points, attacks and damage, speed, and preferred distance from the target. Enemies will attempt to maintain a certain distance from the player, periodically using their attacks. When they reach zero hit points, they die.

# Final Design

The core of the game, including the plot and basic mechanics, remained the same from the original design to the final product, but there are some key differences.

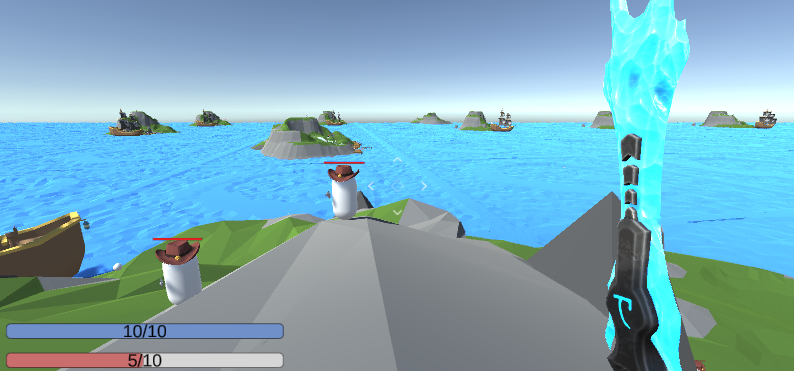
## Core Gameplay

The Game uses basic FPS mechanics, but with a sword that you can throw and teleport to. You can a small amount of damage by throwing your sword into an enemy, a moderate amount by swinging it and hitting them normally, and finally you can do an “execution” by teleporting to an enemy while they have your sword embedded into them, which deals a significant amount of damage.

Additionally, you have mana (mp) to spend on a few special abilities. You can perform midair jumps by spending 2 mp, dash at triple speed by spending 1/second, or heal yourself at a rate of 4hp/sec for 8mp/sec. You recover mp passively when not using other abilities at a rate of 1mp/2sec, and you also recover 2mp whenever you hit an enemy with a melee attack.

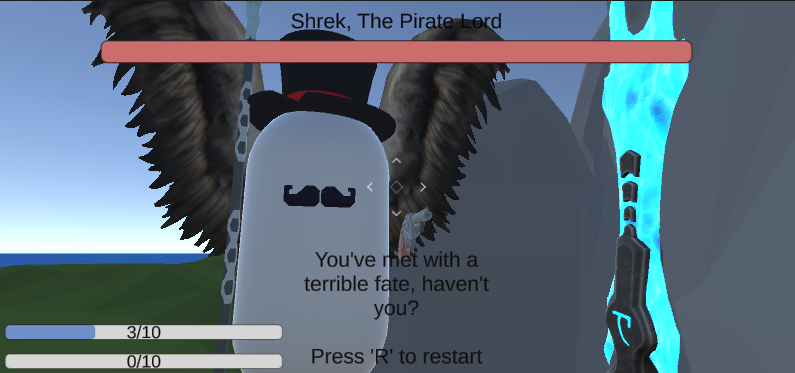
This design differs from the original in a couple ways:

1. There was originally not going to be passive mp regeneration, to prevent players from camping. I changed this because I found that it made the player too conservative with their mana.
2. Things like the extra jump and the sprint were originally going to be magic items you would acquire in the game, but I found they were so useful to the gameplay feel that it made more sense to give them to the player outright.
3. Originally you were going to need health pickups to heal, but I removed those in favor of giving the player the ability to heal themselves with mana, as this creates an eb and flow of the action.



## Game End

Like the original design, the player wins the game by defeating the boss, and they lose the game by dying. However, unlike in the original design, there isn’t a time limit for the player.

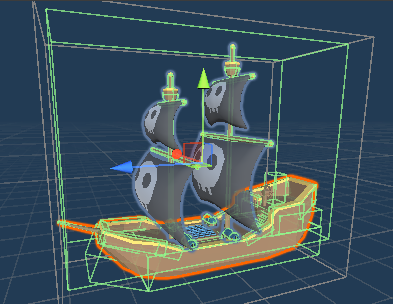


## Sailing

Sailing is where I cut the most things from. In my original vision, there would be enemy ships patrolling the world that would attack you if they saw you. They would fire cannons at you, potentially sinking your ship, and if you boarded them they would attack. This was cut due to time constraints. Now sailing is a method of transportation and nothing more. When you get on a ship you can go to the helm and press ‘F’ to take control of it. When you press ‘F’ again, you release control and it continues as far as its momentum will take it.



There were a few unexpected challenges in making boats work that caused me to spend a significant amount of time working on them. First, you cannot attach rigidbodies to concave mesh colliders. I tried several workarounds, but eventually had to manually insert lots of box colliders on the boat to give it the proper shape and allow for physics to effect it. Second, the process of making the player move with the boat proved to be difficult. I ended up putting a large collision box around the boat so when the player entered the box, the boat would start moving the player as it moved, and when they exit the box they would stop.

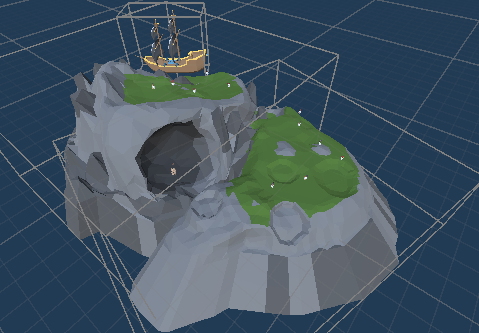


## Locations

At the start of the game the world is randomly generated.

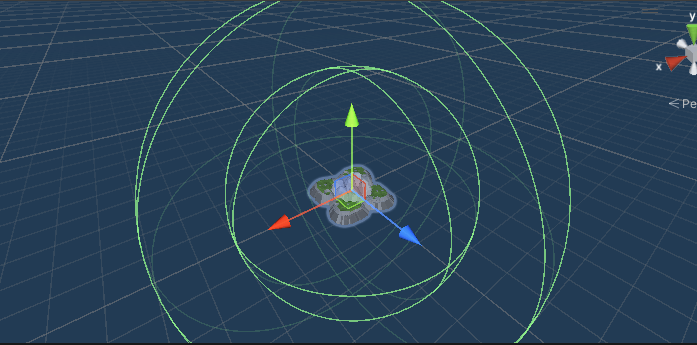
### Starting Island

The player spawns in this island, trapped in a cage. There is a ship by this island for the player to commandeer.



### Boss Island

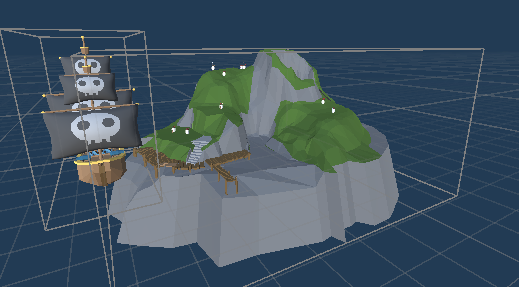
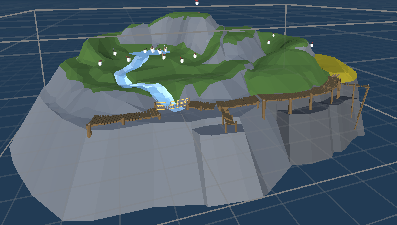
The boss spawns at this island. This island always spawns in the center of the map. When the player enters a certain distance from the boss island, the main music fades out. When they get a little closer, the boss music begins, the light dims, and the boss activates.





### Random Islands

There are two different random island prefabs that will spawn throughout the world. They each contain an assortment of enemies and a chest for the player to open.



## Magic Items

Magic items also changed significantly between the original and final design. Originally there were going to be a few complex magic items, one of each in the world. The final design implements a few simple items, and any time you open a chest you might find any one of them. The magic items can give you +1 speed (you start with 10), + 1 mana (you start with 10), +1 health (you start with 10), and +20% damage.



## Enemy Design

Enemies function largely the same way as they did in the original design, but enemy navigation proved to be more difficult than originally anticipated. When the player strikes the enemies with melee attacks, the enemies are knocked back slightly, which requires the use of rigidbodies. However, controlling something with Navmeshagents completely overrides unity’s physics engine. In the end, I was able to disable the control from the navmeshagent and just get the intended destination. From there I apply forces to the rigidbody that will move the enemy in the direction the navmeshagent would have moved it.

### Enemy Manager

One thing that was not in the original design is the enemy manager. The enemy manager holds a list of enemies and locations. As long as the enemies are in a non-hostile state, the enemy manager will periodically command them to move to a new location. When one enemy enters a hostile state (by taking damage or getting close to the player), the enemy manager will alert the rest of the enemies to attack.

### Enemy Types

There are a couple different enemies in the game with unique functionality.

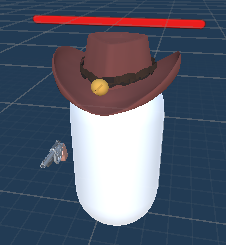
#### Berserker

The Berserker will move about as close as it can get to the player, and when it gets in range will swing a poleaxe at them. The berserker wears a Viking helmet.



#### Gunslinger

The gunslinger will only approach the player if they are far away from them. If the player gets too close the gunslinger will move away. If the player is at a mid-distance, the gunlinger will fire projectiles at them periodically. The gunlinger wears a cowboy hat.



#### Shrek, the Pirate Lord

Shrek is the boss. If the player kills him, they win the game. He can fly. He will move towards the player and fire bullets at them while they are at a distance. Once Shrek gets close to the player he will swing his scythe at them. Shrek Wears a top hat and has a mustache.



# Project timeline

I began working on the project the week of September 18th. I completed the following work each week:

1. (Sep 18th) Added a functional player and made the warp sword teleport the player.
2. (Sep 25th) Created sword animations, sword collisions with the enemies, and health bars for the enemies and player.
3. (Oct 1st) Added player Crosshair. Gave the player the ability to interact with objects, such as ships. Added a bunch of colliders to the boat.
4. (Oct 15th) Made it so the player can sail the ship and swim. Additionally, added the knockback to enemies who are hit with melee attacks.
5. (Oct 22nd) Added basic navigation to enemies.
6. (Nov 5th) Improved the physics of the enemies and created the Axeman (Berserker) enemy. Added animations for the axe and a script to damage the player when hit by the axe. Added gunslinger and bullets for the gunslinger to shoot.
7. (Nov 12th) Created random world generation. Enemies now detect players when they enter a certain radius and alert other enemies. Added the boss. Added basic music. Added loot boxes. Added mana dash and heal.
8. (Nov 19th) Added a popup message for the player when they acquire items.
9. (Nov 26th) Added player death. Various small additions, such as music transitions, displaying player health and mana as a number and giving Shrek wings. Added tutorial messages for the player. Gave the player a hat.

As a whole I enjoyed this class. In total, I probably put somewhere between 60-80 hours of work into the game. I’ve made a game in unity before and have done multiple game projects, so by now I have a decent idea of what a feasible amount of work is, but I think for most people it would be beneficial to create a timeline before being set loose.