There are Lots of You

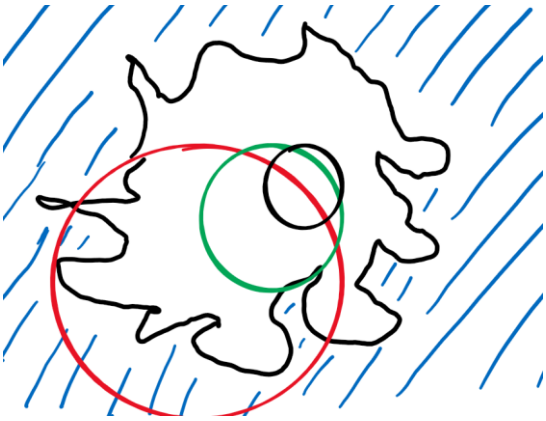
So kill everyone else.

Taylor Vo, Keegan Sanchez

Basic Idea

The core idea we want to create is a game that the entire class can play together at once. When thinking about (more) simple games that we could make with this in mind, the battle royale genre stood out immediately.

The game follows the tenants of the battle royale genre. When the game begins, players choose a location on the map to start at. Players are then free to explore and collect gear to become more powerful. As time progresses, a radius defining the “safe” area shrinks and moves across the map, and the players are force to follow it, or else die outside. The goal is to be the last player standing, so players that confront each other in this shrinking world will fight to the death.



This image shows a mockup of the map, with the “safe zone” shrinking and moving as time progresses.

Combat will be simple, with room for experimentation should time allow. The most important features are a simple slash which will attack in front of the player, and a dodge roll to allow for some counter play.

The game will feature an isometric world, with a locked camera. All animations will be created by us. Mechanically, the game will function as though it is simply top down, though visually it will have isometric graphics.

We have experience working with online games, and even if they’re a little laggy, and the mechanics aren’t great, there’s something very novel about attacking your friends, as long it's consistent. It creates a great sense of competition that is simple and enjoyable. This enjoyment is accentuated by a larger group, especially when they can all communicate with one another.

Development Strategy

Plan

Below is the general order the portions of the project will be completed in.

- Graphics
    - Attempt sprite stacking, settle to isometric if necessary
    - Keegan, Taylor
- Netcode
    - Attempt complex server-client architecture, settle to simple if necessary
    - Keegan, Taylor
- Combat
    - Add attack, roll
    - Taylor
- UI
    - Health, Mini-map
    - Taylor

By November 15th, we intend to have decided on what final approach we are taking for graphics and netcode. Regardless of which approach we take for graphics and netcode, these should be fully functional.

Pitfalls

We expect the most difficult part of this assignment to be the netcode, and following that the graphics. The graphics is something that can be tackled directly, while the netcode must be developed in tandem with the rest of the game's mechanics. Thus, we intend to start by spending some time working on the graphics. If it takes more than a few days to create a result we're happy with, we'll fall back to just a simple isometric world.

A similar approach will be taken with the netcode. The ideal netcode is a server client architecture, with lag compensation (client-side prediction, client rollback, interpolation). However, from experience this is difficult, and takes a lot of time, with many pit falls that can make it take much longer than expected. As such, we have the backup option of simply sending inputs to the server, and having it send the state back to the clients.

Thankfully, for the sprite stacking we have the work done previously with OpenGL to work off of. Though we have no code written in Java for Slick, we do have reference code for the netcode in C++. This game is actually similar to a previous project Keegan has worked on (https://github.com/Lurgypai/Stabby/tree/master/).

High Bar Items

Rotating Isometric World (10?)

The player can rotate their screen so that they can view what's around the player. This will involve sprite stacking.

Multi-Process, Complex (50)

Server Client architecture with client-side prediction, client rollback, and interpolation.

Mini Map (?)

Map will either be a preset layout of a map (separate image of map) or could be the map itself rendered at a smaller size.

Teams (?)

Players will be able to emote when near other players to request to be on a team with them. Teammates will be able to hurt each other, but will see what gear the other team members have where they are on the map.

Total: 60

Low Bar Checklist

Scrolling World (20)

Screen will be fixed on player that will scrolling along the world as they move.

Realtime Game (15)

Game will try mimic a battle royale style game where all players are fighting on same map in realtime.

Multiplayer (10)

Aiming for at least 6 players on the map, each of whom will be on their own system.

Multi-Process (50)

As a base case, we will implement a simple client server architecture.

Power Ups (10)

Gear consists of three tears for weapons and armor. That is, there will be three tiers of weapon, with ascending rarity and damage, and three tiers of armor, with ascending defense. Additionally, the player can collect health potions to regain health. Items will be scattered randomly on the ground and if time permits, in chests as well.

Art (10)

Custom isometric animations and textures.

Isometric World (25)

The world will be big enough where it will fill up the players screen, with a offset angle to give it an isometric view. If a player is behind an object on their screen, then the players outline will show through that object.

Total: 115