

Michael Gutierrez
Alex Katzfey
ISTA 451
6 December 2017

Recoil Final Report

Instructions:

To begin a game first hit play from the main menu. Having one player create a game and have the other player join the lobby by hitting list servers and selecting the game with the name that was just made. Once both players are in the lobby the creator can select the map and game mode. Once both player hit join a 3 second countdown will start and the game will begin. Once in game the players can use wasd to move around with the jetpack and the left-click to fire their weapons. The players must complete the objective to earn points and kill their opponent to prevent them for completing the objective. The objective will be different depending on the game mode. At the end of the timer the player with the most points will win (there is an overtime condition for the oddball game mode).

Game Design Document

Title: Recoil

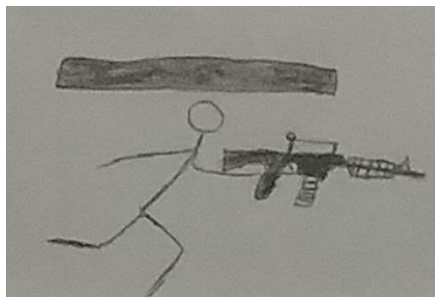
Genre: 2d shooter/platformer

Target Platform: PC

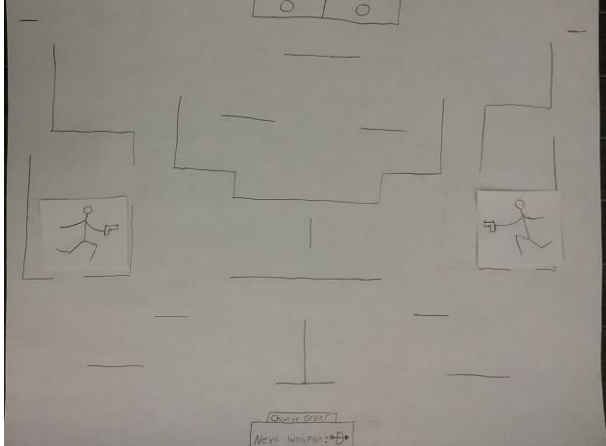
Target Audience: Competitive players who enjoy online multiplayer

Unique Selling Point: Your weapons recoil will propel you to move quickly around the map.

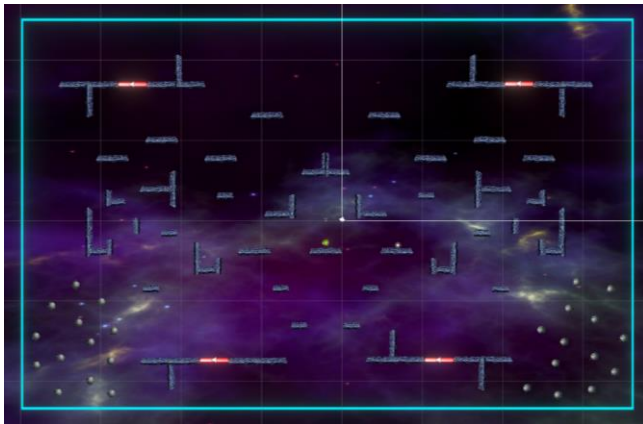
Character Design: The character used was the standard 2d character asset. Since there is no story to the game, there is also no backstory to the character. The original character designed looked like a simple stick figure seen below on the left. Ultimately, the standard asset was much better than anything we could create or find, thus we went with the robot boy (seen right).



Level Design: The main goal of the level design was to provide obstacles to move around and protection from bullet fire. To do this we wanted platforms scattered around the map. You can see the main idea of the levels in this early sketch.



Once implementing the full first map we made the level much bigger and added environmental elements to interact with. These elements can help add to player's strategy. Another thing we kept in mind while designing the map was making some parts of the map wide open and other area narrow. This way there is more diversity in how to play based on which are of the map you are on. Below is the final look of map 1.

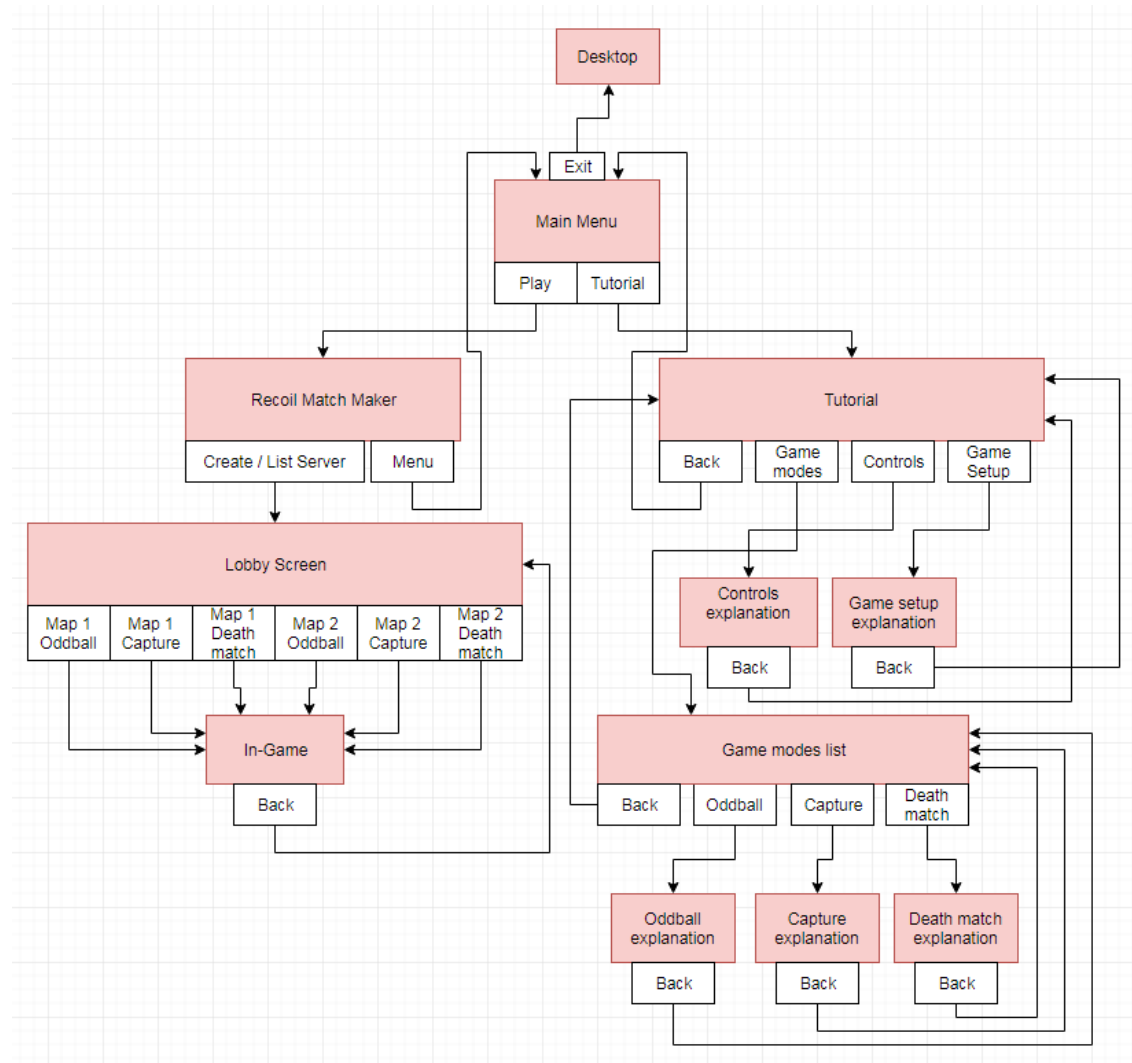


We also decided to add a 2nd map which would be a circle that rotates. However after playtesting it a bit, we found it was too chaotic with the rotating so we decided to scrap that idea. Instead we have a new map that follows the same level design idea as map 1 but has a much different layout. Below is the final look of map 2.



Menu flow board:

(Begins at main menu)
 Red Squares - scenes
 White Squares - Buttons



Full Game Description:

Recoil is a 1v1 2d competitive shooter/platformer. What is unique about the game is that while you can use wasd to help control your movements; allowing your guns recoil to help you move around the map is usually going to be the more effective option. The goal of each game is receive earn more points than your opponent by the time the timer runs out. This is accomplished by playing the objective for each game mode. There are 3 different game modes in recoil which are all 1v1. The game modes are oddball, capture, and death match. In addition, there are two different maps to play on. On these maps are going to be weapons, health, explosive barrels, trip mines, laser, and fire boundaries. Weapons will spawn around the map randomly every few seconds which players can then pick up and use. Health packs, barrels, and lasers will also spawn at certain time intervals. Players can use all these map elements to their advantage in order to get a leg up on their opponent.

In oddball the objective is to grab the neutron core, which is placed in the center of the map at the start, and maintain control of the core for longer than your opponent. In order to grab

the neutron core the player simply walks over the core. A player will drop the core where they are when they die. The neutron core will be placed back in the center if the player dies outside the map. A player can die by having their HP go to zero by getting hit by their opponent's weapons or by being hit by environmental hazards. When a player dies they will have a respawn delay of 3 seconds and spawn back on the map. At the end of the timer if the person holding the neutron core is has more points, then the game ends and that player is the winner. If the person losing is holding the core then the game will enter overtime and will only end once the player holding the core dies. At that point the game will end and the person with the most points wins. If nobody is holding the core, then the game ends and the winners is the one with the most points. It is possible for the game to end in a tie only if the timer runs out and neither player is holding the core and the score is tied, or if it is overtime and the player with the core dies just as they tie up the score.

In capture, players must collect the neutron core more than their opponent. When a player collects the neutron core, they will receive ten points and the core will immediately respawn elsewhere on the map. By the end of the timers the player with the most points will be the winner. It is possible for the game to end in a tie if the score is equal at the end of the timer.

In Deathmatch, there is no neutron core. The only objective is to kill your opponent. When your opponent dies then you receive 5 points and the person with the most points at the end of the timer will win. It is possible for the game to end in a tie if the score is equal at the end of the timer.

Finally, it is an important aspect of the game to keep track of the arrow indicator of your screen. If the objective is ever off of your screen then the arrow will appear and point to the objective. For the oddball and capture game modes the arrow will always point to the neutron core. In the Deathmatch game more the arrow will always point to your opponent.

-----END GAME DESIGN DOCUMENT-----

Testing and Debugging Information:

- **Slow character with the neutron core** (changed): During the playtest this feature was already implemented however after receiving the feedback, we decided to make the slow more noticeable.
- **Make game setup simpler** (changed): We added a game setup tutorial screen to help players learn how to start a game.
- **Add visual boundary to the map** (changed): Not only did we add a blue boundary; we also added radiation noise when outside the map and also set the player on fire when outside the map.
- **Don't let players spawn next to each other**(changed): We adjusted the spawn system so that it check the other players distance before spawning and choosing a spot that is off screen.
- **Have an arrow which points to the neutron core** (changed): We added an arrow that points to the neutron core in oddball and capture and an arrow that points to the opponent in death match.
- **Display type of weapon you currently hold** (unchanged): The gun have very distinct looks when holding it so this idea seems unnecessary and would just clog up the UI.

- **Add ammo capacity**(unchanged): Using your weapons to move is vital to the game so adding a ammo cap would put too much limit on players. Instead we have a rate of fire variable to limit how much a player can shoot.
- **Make a tutorial** (changed): Added tutorial screens to cover all aspects of the game.
- **There is a bug where players can shoot through walls** (changed): Players can no longer shoot through walls.
- **Allow players to move with wasd/ smooth out gravity** (changed): We gave players slow jetpacks to make movement feel more natural. It was strange playing a pc game with just the mouse. It also allows for us to add guns like the black hole gun which does not propel to fast. These jetpacks are not meant to be the main way to move around the map, be simple you help you.
- **Add a barrier instead of killing players** (unchanged): We made the barrier clearer so this should not be a problem anymore.
- **Let player choose their character** (unchanged): We do not want to add skins to the game. Unnecessary cosmetics can distract from the main gameplay.
- **Add more maps**(changed): Added the 2nd map
- **Add more items**(changed): Added laser, trip mines, black hole gun, and rpg

Outside Assets:

- RobotCharacter prefab from 2d standard assets
- Disfigure - Blank : <https://www.youtube.com/watch?v=p7ZsBPK656s>
- Unity 2d standard assets
- Cartoon Hand Drawn Weapons and Backgrounds by BIGINDA
- Futuristic weapons set: <https://www.assetstore.unity3d.com/en/#!/content/15644>
- Asteroid Sprite: <http://www.bionicinterface.com/web/googida/art4ever/a0130/a02.png>
- Neutron Core Sprite:
<https://vignette.wikia.nocookie.net/mugen/images/3/3d/CN9.gif/revision/latest?cb=20110617222819>
- Barrel Sprite:
https://vignette.wikia.nocookie.net/clubpenguin/images/a/ab/Hazard_Barrel_sprite_001.png/revision/latest?cb=20160923221344
- Long platform: <https://marcodonoghue.files.wordpress.com/2014/07/long-platform.png>
- RPG sprite:
http://images4.wikia.nocookie.net/_cb20090604043527/gtawiki/images/0/03/RocketLauncher-GTA4.png
- Simple FX - Cartoon Particles by Synty Studios
- Death sound: <https://www.youtube.com/watch?v=2uhxrPhVO8Y>
- Fog of War Gun Sound FX Free by Into The Void Sound Design
- 3D Games Effects Pack Free by Creepy Cat
- Radiation sound:<https://www.youtube.com/watch?v=9BVckjdZIK4>
- Black hole sound: **LittleRobotSoundFactory**
- Planet texture: <http://www.texturesforplanets.com/texture-packs.shtml>
- Tutorial: <https://unity3d.com/learn/tutorials/modules/beginner/live-training-archive/creating-a-scene-menu>

- Tutorial: <https://gamedev.stackexchange.com/questions/125986/how-can-i-get-ui-element-over-which-pointer-is-in-unity3d>
- Itemize by scrampunk: <https://freesound.org/people/Scrampunk/sounds/345297/#>
- Map 2 song: <https://www.youtube.com/watch?v=1KqQQHpQc8w>
- Mine: http://clipartist.net/links/clipartist.net/rg_1_24_sea_mine_scalable_vector_graphics_svg_scalable_vector_graphics_svg_clip_art_wall_paper_background-999px.png
- Laser: <https://people.rit.edu/jxm1503/330/Project%202/media/images/laser.png>
- Black hole gun: https://vignette.wikia.nocookie.net/saintsrow/images/7/72/SRIV_Rifles_-_Alien_Rifle_-_Dominator_-_Default.png/revision/latest?cb=20131029214838
- Black hole: <http://www.computerkernel.com/content/img/blackhole.png>

Workload Distribution:

Michael: I did a majority of the work on the maps/UI. I worked in on creating the Hierarchy of the scenes and Alex made a lot of the networked prefabs. I also made the menus aside from the menu art. All assets were found, not made. For some of scripts that I made Alex edited them so that it can be networked.

Scripts I worked on:

- PlayerHealth: I made a few bug fixes
- NeutronCore(later turned into NeutronCoreNet): Allowed players to pick up core and increased score while holding core
- HighlightText: Make the button text bold if the mouse is over it
- HealthPack(later turned into HealthPackNet): Increases player health when the health pack is touched
- GameManager(later turned into GameManagerNet): The first iteration of oddball
- BarrelHealth: Show what the barrels health is currently at and checks for damage
- BlackHoleDamageNet: Player takes damage if in the black hole
- BlackHoleNet: Fires black holes
- BlackHoleSpeedNet: sends a black hole in a certain direction and checks for collision
- LaserActivator: Turns the laser on and off
- LaserBeam: Originally damaged player if player entered, Alex changed it to push players away
- TripMine: Creates an explosion if the mine is touched
- UIManager: Controls the UI

Alex: I did a majority of the work on creating weapons and networking the game. I also made the different game modes. I also added sound effects to most items.

Scripts I worked on:

- Everything related to networking
- Player controller code
- Weapon classes
- Game Management and Lobby managers
- Player Health

- Boundaries
- Map spawning items code

Challenges:

Michael: A big challenge we faced was getting assets for our game. Neither of us knew how to create any assets so we had to find a lot of them online. Creating the maps so that they are not cluttered with spawn point yet there is also enough was quite difficult. To solve this I made sure no spot had two different types of spawn points. The final challenge that we face was time.

There was a lot more we wanted to add but to solve this problem we had to prioritize. We had a list of things that needed to get done with most important at the top and least important at the bottom. This helped us focus on what we needed to get done like menus/tutorials and bug fixes rather than just trying to add more elements to the game. We were able to refine our game into a nice finished product.

Alex: Networking was the most difficult aspect of this project to me. I went into this class with a decent amount of game design knowledge and Unity use, and wanted to push myself to create a networked game. Figuring out how to deal with lag time between server and clients was a large difficulty for me at first, as well as figuring out how to segment the code to work on only server or only client. Similarly I had to learn how to sync variables between scripts, and ensure that players were only allowed to fire bullets on the server, as to avoid cheating.