Kyle Lukaszek
1113798

CIS 4820 Final Project: Sprite Fight - Kyle Lukaszek

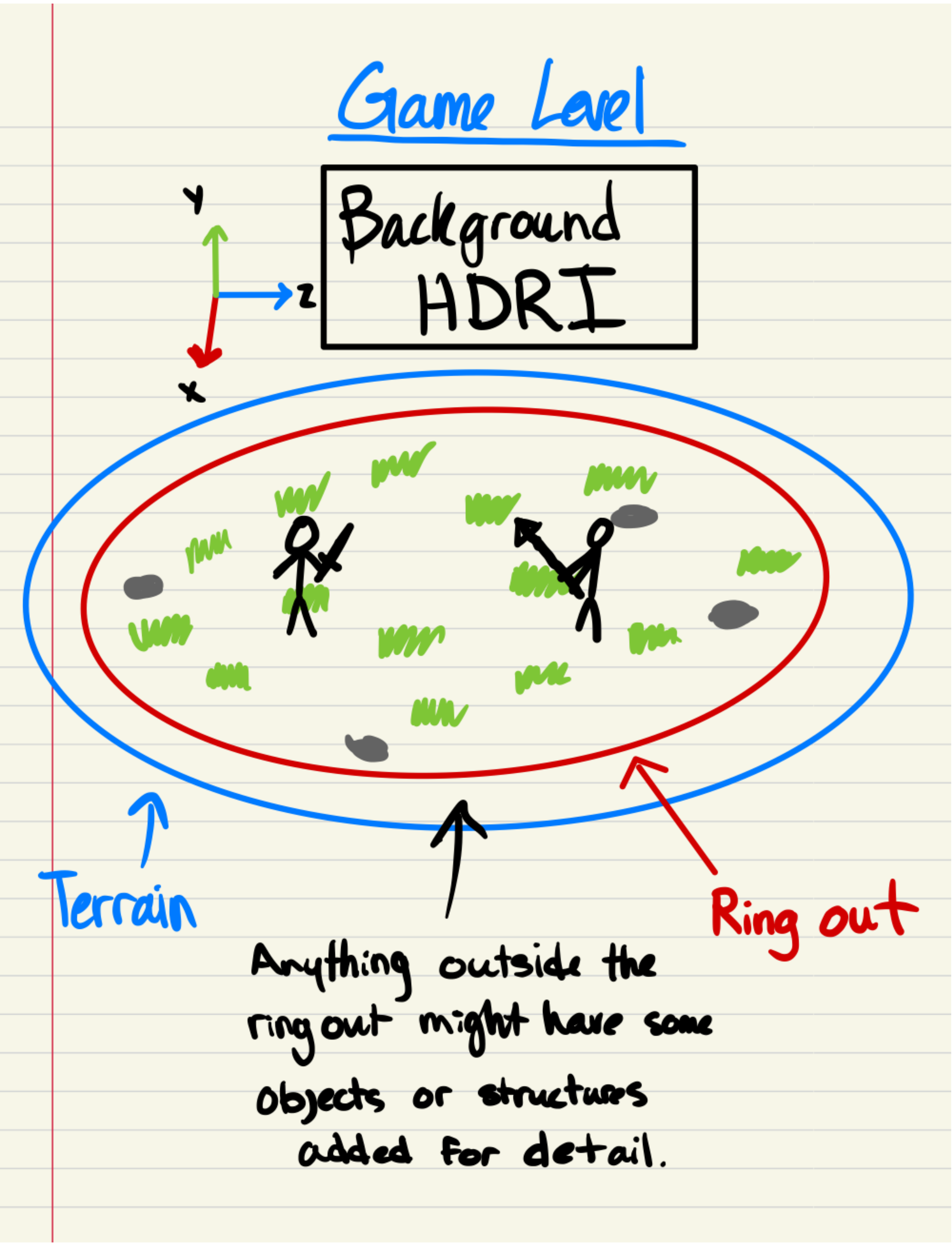
Game Overview:

I plan on making a combat-focused isometric 2.5D sprite-based fighting game with health, stamina, and a "ring out" mechanic. The player can play as 1 of 2 classes based on fictional warrior archetypes (whatever assets I find look best and are easiest to work with). Each class will play differently and have its own strengths and weaknesses. For example, a sword user might attack slower than a spear user, but their attacks will deal more damage and apply more knockback. Each class will have 2 attack options, light attack, and heavy attack which have different stamina costs when performed. Each class can also roll and block which will cost stamina as well. Holding block costs stamina per second and block cannot be performed again until the block animation has been completed to discourage block spamming. If the player or NPC blocks at the right moment, they will perform a parry which will return stamina allowing them to combo further. To replenish stamina, the player or NPC must not perform actions for approximately one second for the bar to begin increasing. If the stamina bar becomes depleted, the player or NPC becomes tired for 2 seconds and becomes incapable of performing any actions until they recover. This makes stamina management important as you could easily lose by spamming attacks or blocks and then becoming immobilized which could lead to getting knocked out of the ring, resulting in a loss.

Like most fighting games, the players/player and NPC are always facing each other, so as the entities move around the ring they will never have to worry about an attack missing as long as they close the gap or the enemy does not roll at the right time. The game will consist of selecting a class and then battling a random class. If all goes well in development, by the end I hope to add atmospheric lighting, and other post-processing effects to emulate an "Octopath Traveler"-like graphical aesthetic for the game.

Kyle Lukaszek
1113798

Game Level:



Anything outside the ring out might have some objects or structures added for detail.

Kyle Lukaszek
1113798

Player Controls and Actions:

W - A - S - D / Left Stick: Movement

- W and S will move the player along the X-axis (seen in the **Game level diagram**).
- A and D will move the player along the Z-axis (seen in the **Game level diagram**).
- Movement speed is uniform (will have to try rates to see how this feels).

W - A - S - D + Spacebar / Left Stick + Circle/B: Roll

- Roll and gain invincibility frames (i-frames) for a short duration.
- Costs a quarter of the player's max stamina.
- Rolls take time to complete and can be punished if you miss the i-frame window.

Left Mouse Button (LMB) / R1: Basic Attack

- Fastest attack option with cheap stamina cost.
- Weakest attack and does little damage to enemy stamina.

Right Mouse Button (RMB) / R2: Heavy Attack

- Slowest attack option with expensive stamina cost.
- Strongest attack and does 1-quarter max damage to enemy stamina.

Left Shift / L1: Block / Parry

- Reduces incoming damage to health and stamina by 75%.
- Holding block costs stamina per second (will have to try rates to see how this feels).
- Slow down by 50% while blocking.
- 25% knockback reduction while blocking.
- Blocking as an attack connects results in a parry that negates chip damage and does not cost stamina. It even refunds stamina to the player to allow for combo potential.

Kyle Lukaszek
1113798

Animations for both the player and the enemy:

Required states: Idle, Left, Right, Light Attack, Heavy Attack, Block, Roll, Tired, Hit, and Death

Unity Packages: 2D sprite and 2D Animation packages from Unity will be needed to manage sprites in a 3D environment.

Notes:

- Since players can play as any class, the player and the NPCs will have the same animations as they share the same states.
- I plan on using open license sprites from Luiz Melo from the Unity asset store.
- Some of the animations will have to be edited or created, such as Light Attack, Heavy Attack, and Block.
- Some of the animations will have to be created based on the existing sprites, such as Tired and Roll for some specific classes.

Animation State Machine:

	Idle	**Move**	**L.A**	**H.A**	**Roll**	**Hit**	**Block**	**Tired**	**Death**
Moving (float < 0.1)	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Idle	Death
Moving (float >= 0.1)	Move	Move	Move	Move	Move	Move	Move	Move	Death
Light Attack (trigger)	L.A	L.A	L.A	L.A	L.A	L.A	L.A	L.A	Death
Heavy Attack (trigger)	H.A	H.A	H.A	H.A	H.A	H.A	H.A	H.A	Death
Rolling (trigger)	Idle	Roll	Roll	Roll	Roll	Roll	Roll	Roll	Death
Take Hit (trigger)	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Hit	Death
Block (trigger)	Block	Block	Block	Block	Block	Block	Block	Block	Death
Tired (trigger)	Tired	Tired	Tired	Tired	Tired	Tired	Tired	Tired	Death
Dead (trigger)	Death	Death	Death	Death	Death	Death	Death	Death	Death

Kyle Lukaszek
1113798

Enemy/Class Descriptions:

- Enemies will share the same AI logic for combat but they will have different stats.
- Enemies can perform the exact same actions and animations as the player. See **Player Controls and Actions** as well as **Animations for both the player and the enemy**.
- Enemies can all be based on a character controller script that has adjustable properties that can be tweaked to match the class descriptions.

Spear:

- Fastest but most frail class
- Lowest health, lowest stamina.
- Attacks have the lowest frame counts but deal the least damage out of the 3 classes.
- Knocked back further (makes up for increased speed)
- Most invincibility frames on roll.

Sword:

- Average speed and survivability
- Attacks are slower than the spear class'
- Attacks deal average damage.

Note: I ended up removing the heavy class since I could not find a decent enough sprite that matched the original description. Making my own sprite and sprite sheets for animations would have been a nightmare and development was already harder than anticipated as one person so I skipped the third class to make sure the two implemented classes work as intended.

Kyle Lukaszek
1113798

Enemy State Machine:

- I will develop a way for fighters to interact in the game with an ActionManager and an NPC controller class to manage NPC state logic.
- States are ordered by the highest to the lowest chance of occurrence. Chance-based states will be operating on a clock to avoid changing states too erratically. For example, if the 'Defensive' state is selected, the NPC might go backward and we don't want that to happen for 1 frame so we set a minimum duration.
- There is no death state since death is managed by the Fighter class for both players and NPCs.
- There is no tired state either since being tired is also managed by the Fighter class for both players and NPCs.

	Neutral (Random Move / Approach)	**Combat (Light Attack, Heavy Attack)**	**Defensive (Block, Roll, Random Move, Back up)**	**Desperate (Combat, Roll, Attempt Parry, Random Move)**	**Bad Spot (Approach Origin)**
Outside Range	Neutral	Neutral	Defensive	Neutral	Neutral
Tired Recovery	Bad Spot	Bad Spot	Bad Spot	Bad Spot	Neutral
In Range & Healthy	Combat, Defensive	Combat, Defensive	Combat, Defensive	Desperate	Bad Spot
In Range & < 50%hp	Defensive, Combat	Defensive, Combat	Defensive, Combat	Desperate	Bad Spot
In Range & < 20%hp	Desperate	Desperate	Desperate	Desperate	Bad Spot
Within 1 unit of ring out	Bad Spot	Bad Spot	Bad Spot	Bad Spot	Bad Spot
Successful Parry	Combat	Combat	Combat	Combat	Combat
Opponent Tired	Combat	Combat	Combat	Combat	Combat

Kyle Lukaszek
1113798

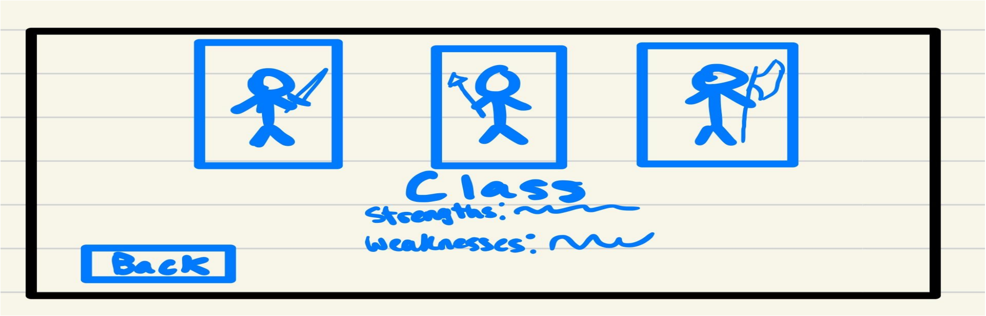
UI Elements:

Main Menu:

- The main menu will need a “Play” button, an “Options” button, and a “Quit” Button.
- “Play” takes the user to the character select screen.
- “Options” will have a resolution dropdown, fullscreen toggle, and audio slider.

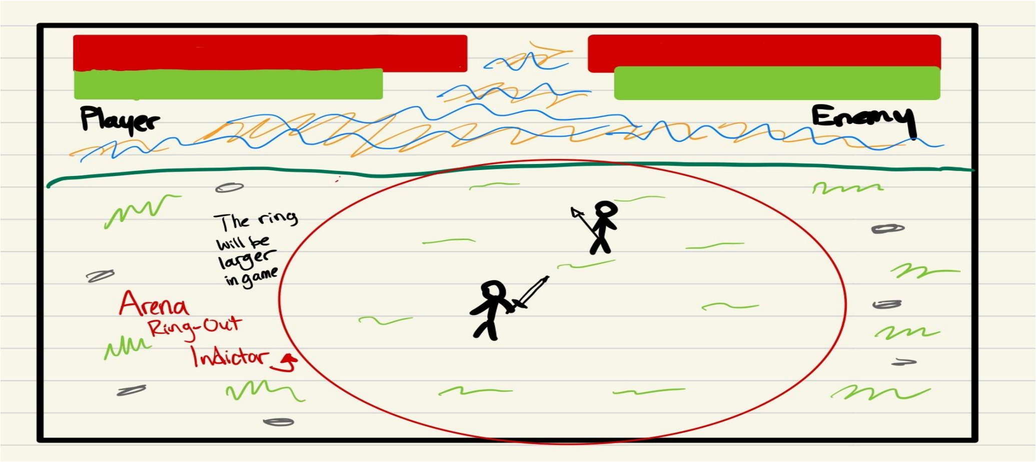
Character Select:

- The user should be able to select any of the 2 classes and see a brief description.
- There should be a “Back” button to return to the main menu.



Gameplay:

- There will be a health bar colored red for both the player and the enemy NPC located at the top of the screen.
- There will be a stamina bar colored green for both the player and the enemy NPC located at the top of the screen.
- Red arena ring-out indicator based on the sphere being cast over the area.



Kyle Lukaszek
1113798

Development Timeline:

March 11-18:

- Create a basic game scene with a working camera.
- Work on a player controller script with all the right controls and actions.
- Implement Arena Ring using spherecasting to check if an entity leaves the ring.
- Implement knockback by assigning the player and NPC the Rigidbody component.
- Implement the health bars and stamina bars as UI components.
- Begin work on an NPC controller script with working combat logic.
- Design very simple-looking sprite animations based on the existing sprites. They don't have to look good or make much sense whatsoever, just use the same colors.

March 19-26:

- Finish the NPC controller script if it is not already done.
- Implement the fight order based on the player's selected class.
- Continue working on sprites if still needed.
- Implement a main menu and a character select screen once the classes are tweaked.
- Properly animate the health bar and stamina bar UI components.
- Work on implementing sprite-based animations using the Unity packages 2D Sprite and 2D Animations. I expect this to be the most annoying thing to implement.

March 27-29:

- Fix any bugs that might remain and polish the game.
- Add more detail to the game scene. I'm thinking of grass assets and maybe some structures in the background.
- Maybe mess with Unity's example shaders to add shadows to the sprites and maybe some cool lighting. This would be the final thing attempted, granted it doesn't drastically reduce performance.

Kyle Lukaszek
1113798

Final Development Notes:

This can be done with 1 person. The player controller and the NPC controller are fairly basic and don’t stray far from what we have worked on in class. The hardest part will be creating sprites based on the existing ones and working with the sprites. Getting the animations to last the correct amount of time while an action is still active will most likely be a pain since the action duration must last a similar duration as the sprite animation frame count. The enemy logic might be a little difficult to work with but it will mostly operate on random actions while within range of the players since it does not know if the player will be ready to attack or if they will be expecting an attack.

Credits:

- **Sprites:** Luiz Melo
    - Huntress
    - Martial Hero 2 (Samurai)
- **Music:** 進撃 st hrn egt20130629 巨人 - Shigekin no kyojin, Hiroyuki Sawano
- **Sound Effects:** FreeSound.org
    - Victory Cry Reverb 1 - Chripei @ FreeSound.org
    - Metal Impact 5 {Soft} - Sophia_C @ FreeSound.org
    - Knife Swing 1_4 - Joao-Janz @ FreeSound.org
- **Fonts:**
    - https://fonts.google.com/specimen/Roboto
    - https://fonts.google.com/specimen/Nabla