Arborea

Group Members

- Johnson Chau 101211212
- Kayla Gingras 101166181
- Ryan Peckham 101114328

Overview

The theme of the game is fantasy, and the setting is a magical forest arena. The characters and environment are all magical and whimsical. The main objective of the game is to destroy the enemy tree while defending your own. Destroying the enemy tree will result in a victory, whereas having your own tree destroyed will result in a defeat.

Units

We have 3 player units, each with different abilities and stats such as attack, attack range, and movement speed. All of the units are humanoid.

1. Gambler Cat - this unit can move, attack, pick up items, use a healing item, and has a blink ability.
2. Wizard - this unit can move, attack, pick up items, use a healing item, and has a heal ability where it can target itself or an ally to heal. It can also heal structures.
3. Dog Knight - this unit can move, attack, pick up items, use a healing item, and has an ability where it buffs its own attack and movement speed.

We also have a ‘wisp’ minion unit which can be spawned when the player or enemy has sufficient summoning crystals. The wisps act on their own to go attack the enemy structure.

The enemy units mirror the players’ units but with a red visual outline for clarity. Each of the enemy units have the same capabilities as the player units.

The static units we have are a protective shield structure, and a main structure that you need to protect throughout the game:

1. Shield Structure - this structure provides the main tree structure with a defense buff, regenerates its health over time, and is able to summon wisps that attack the enemy tree when given summoning crystals.
2. Main Structure - this structure gains a defense buff when the shield structure is alive, regenerates its health over time, and the game ends once either of them are killed.

Within the game you will also be able to find several resources:

1. Powerup: The golem in the middle of the map gives a powerup when killed
2. Summoning Crystals: Collected and can be used to summon wisps
3. Healing Potions: Picked up, then allows the use of a healing action

Technical Requirements

We believe that we have met all technical requirements outlined in the 1st and 3rd project check ins.

1. Our game has an isometric view with both lighting and shadows affecting everything in the game world.
2. The terrain, static and dynamic units, and resources that can be collected each have 3D models, along with the appropriate textures and materials.
3. We use different kinds of colliders where appropriate to allow for proper collision detection between objects as well as between objects and terrain.
4. Two potion resources and two crystal resources spawn upon starting the game. The potions and crystals respawn every so often.
5. The units are fully animated and each action properly affects the game world.
6. We use a NavMesh for pathfinding behaviour.
7. The minions in our game will pathfind to the enemy tree when they are summoned.
8. The enemies have AI, the code for which can be found in EnemyAI.cs. The enemy AI has a priority system determined by a few factors. It will pick up items, attack the player, retreat from losing combat and call allies, use its abilities, and attack the enemy structures.
9. We have implemented full game mechanics, as you are able to use all types of resources in the game, all abilities, and the player can win or lose the game.
10. Each interactable thing has an outline that displays the team that it is on (red for enemy, blue for ally, yellow for neutral). We have health bars for all attackable units and when a unit is attacked or healed, a number popup will occur (orange for ally damage, red for enemy damage, and green for healing). When certain abilities are used, pink text will show up to notify the player. When an ability is pressed and it is on cooldown, the cooldown will be displayed. If a player does not have an item and attempts to use it, a popup will appear telling the player that they do not have enough items.

Additional Features

We implemented a few features to our game that went beyond the specifications:

1. A start screen, controls menu, and victory/defeat screens.
2. Logic for creating new units (wisps).
3. Items and heroes respawn after time using the ItemRespawner.cs and HeroRespawner.cs classes.
4. The degree of polish in our game world and units - all of our playable units are animated with low-poly models that we think give our game a cohesive look.
5. We have included a lot of elements for player feedback, such as our health bars, outlines, popup damage/healing numbers, and cooldown messages.