

Overall Design:

Our goal is to implement 2 AI controlled enemies that will perform different actions. We will implement this with a finite state machine where the actions will be in a particular state of the group and will change states based on various factors from the surroundings. We plan on implementing a thief enemy that can chase and steal food, retreat away from the player, and teleport from the player's direct attacks. We also plan on implementing a fighter enemy that can chase the player, attack the player, and dodge attacks from the player.

For the thief enemy, it'll detect the items around it to decide its actions. First, it'll detect if the player is within a certain radius and will run in the opposite direction (map allowing). Second, it'll detect if the player is attacking and will try to dash away and dodge the attack. Lastly, if it doesn't detect the player then it'll detect the nearest food item and move towards it with the intention of stealing it.

For the fighter enemy, it'll also detect items (mainly the player) to decide its actions. It'll detect the location of the player and go towards it (chasing the player). It'll also detect when it's within a certain distance from the player and will be able to attack. Lastly, it'll detect when the player is also attacking and will try to dash out of the way to avoid getting killed.

We will debug unexpected AI behaviors with a few methods. First, we will look at the source code to see if we were mistakenly implementing something or if there were simple mistakes made. Second, we will look at the expected vs the observed behaviors to see if there is a pattern detected in the comparison. Lastly, we will tune different parameters in order to see their effects and to overall see the way the AI interacts with different situations and contexts.

Software Architecture and Plan:

We plan on using a mix of polling and event listening to determine the behavior of our AI. To get the target for our AI's pathfinding, we will use polling to get the player and food positions. This will also be used to determine if the fighter enemy wants to attack or if the thief enemy wants to run away. To determine if the enemies want to dodge, we will use a mix of an event driven and a polling system where, if the enemy is in range of the player (polling) and the player attacks (event driven) the enemy will dodge. The fighter's dodge will be a more traditional rolling out of the way of the player. The thief's dodge will be a teleport, moving it to a random position in the world. Setting up the 2 agents themselves hopefully won't take too much time (only 1 day or 2) while most of the work will be in the actual AI behavior.

We will implement a finite state machine to regulate enemy behavior. We will use a graph of different states connected by edges that have conditions attached to them to transition between the various behavior states. For each enemy type, it will contain 3 states. For the fighter: chase player, attack, and dodge. For the thief: chase fruit, run away, and teleport. We will most likely have classes for each behavior and have 2 different finite state machine classes that are connected to the 2 individual agents. We will utilize the functions from these behavior classes to control the AI. This will be a main bulk of the time since it requires many different factors to be considered in each behavior state and has many inputs all interacting with each other in different ways for a single decision for the AI. We estimate each behavior will probably take a day to implement (but are splitting the behaviors up so it will hopefully only take each person a day for each behavior resulting in only 2 or 3 days for the behaviors).

We plan on using the built-in Godot A* pathfinding algorithm. The thief will have to choose which fruit it wants to chase. If the player is near the fruit or between the thief and the fruit, it will have to pick a different one. This means that the thief will need knowledge of both the fruit and player positions. This will most likely take the

most amount of time (2-3 days by itself) since it is the foundation of the movement and will need to be implemented before we consider the other extra behaviors.

Division of Labor:

Our plan is to have all 3 of us work on the AI equally but on different sections. We plan to all set up the foundations of the enemy nodes so that we are all very familiar with the basics but then plan to expand separately implement the more unique features. We will all work on the movement and navigation together most likely but then do the behaviors differently.

Silas will most likely focus on the fighter enemy's chasing the player as well as the thief enemy's retreating from the player. These are similar in that they detect the location of the player but act in opposite ways.

Josh will work on the thief enemy looking for and eating the food and the fighter player trying to attack the player when within a certain distance. Both of these rely on detecting the distance of the item (food and player) to act accordingly (stealing and attacking).

Aidan will work on the dodge mechanics of both the thief and fighter enemies. The dodge of the 2 are similar in that they detect the player attacking and react accordingly but are different in that they dodge the attack differently.

We plan to meet up at least once each day. This meetup will be a quick check in and report of any blockers that we can work through on some days and working together on a call on other days. Overall, the goal is to meet up frequently so that we are all kept up to date with each other's progress.

Milestone Demo Code:

The demo code will be included in this repository in the GitLab. The controls are the same as the last assignment at the moment since we haven't implemented the player attack action yet. These controls are displayed on the GUI of the game but are basically WASD for basic movement, space bar for jump, shift and H for ledge interaction, and G for gliding. Rotation mode can be toggled with R and music/sound effects can be toggled with M and the comma (,).

The main new pieces are the fighter enemy and the thief enemy. Both enemies are created and now exist in the game. They move towards their goals (player and food) respectively although have a few bugs at the moment. After we finish smoothing out the movement and navigation, we can start adding in the extra behaviors.