Introduction

This week, we’re finishing Wiz with AI and pathfinding. But also, we’ll begin brainstorming for your final projects! Yay, exciting! (we hope...)

As part of this checkpoint, please fill out this Google Form (Also linked on the website at “Final Project Idea”) with a rough idea of a game you’d like to make. It's not binding, and everyone should submit one, even if you already know you want to work in a group. Be creative! You don’t need to know every detail of your idea, but give us a general idea of the gameplay/genre you’re going for. Think about games you’ve played in the past that you’ve enjoyed, or your dream crossover game, and think about what makes a game fun for you. For this week, this idea submission and the design check will both be worth 0.5 points.

We’ll also pitch these in class next week (October 25), so come to class if you’re able to!

Design Check

- Which AI framework will you use? What engine-side classes do you need to implement this? What behaviors will you need for your game objects to interface with those classes? When will the AI code actually be called?
- How will pathfinding be integrated into your engine? How will your AI make use of pathfinding?

Primary Requirements

Primary Engine Requirements

- Your handin must meet all global requirements.
- Your handin only crashes under exceptional circumstances (edge cases)
Your engine must implement an AI library using either behavior trees or goal-oriented action planning.

**If using behavior trees:** You must include a behavior tree, selector, and sequence class. You must also include a node and action/condition interface.

**If using GOAP:** You must include a game state class, an action class/interface, and a condition class/interface. You must be able to search over an action set using a predefined start state and goal condition.

In both cases, your engine must have an A* implementation of pathfinding.

### Primary Game Requirements

- Your game should have a map containing passable and impassable tiles.
- Your game must have a unit that can be controlled by the player.
- Your game must have at least one enemy unit that moves around deterministically (i.e., the same actions by the player result in the same enemy behavior). There must be a visible reaction when the player and item collide.
- The player-controlled unit must not be able to leave the map.
- The enemy unit should use your engine’s AI framework. All of the AI tools included in the engine requirements should be used when constructing your AI. All of the behaviors defined for your enemy should be visible at some point when playing the game.
- It must be possible to start a new game without restarting the program.

### Secondary Requirements

#### Secondary Engine Requirements

- Your engine must meet all primary engine requirements.

#### Secondary Game Requirements

- Your game must meet all primary game requirements.
- The enemy unit should move according to a path generated using A*.
- Your game must meet at least two of the extra game requirements (that weren’t used for last week’s game requirements).
Handing In
Handing in works the same way as it did last week! Feel free to refer to the handin reference for a walkthrough of each of these steps. To hand in...

1. Push your final handin commit
2. Create a release for this handin with Wiz2.0 as the tag
   a. You should have separate releases for Wiz I and Wiz II!

Don’t forget to upload a demo video to the demos slack channel!