KAIJU AGENTS IN UNITY

COMP 4770: ASSIGNMENT 1

Presented by Pelumi Towuru

Overview

- Sensors
- Actuators
- Agent Controllers
- Code Walkthrough
- Demo
- Questions

Sensors

For this assignment, I programmed two sensors: FloorCleanerSensor, which extends the KaijuVisionSensor with Floor parameters, and FloorGroundSensor.

FloorCleanerSensor stores the observedDirtyTiles as a Generic Collection of type Floor, provided that there is a Floor and it is Dirty. However, we don’t want our Agent on an aimless wander, so we calculate the nearestDirtyTiles for each tile in observeDirtyTiles and find the nearest based on them. A Debug function was added during initial testing to ensure that all the observedDirtyTiles were being detected and stored properly. All Floor Tiles became dirty, and the Update Method was commented out.

.

FloorGroundSensor detects tiles around the Kaiju Agent using Physics.Raycast so that not all tiles are magically seen once the game is running. It stores a bool reference to onFloor, indicating whether the MeshCollider is dirty. We continuously update the floor beneath the Agent using OnFloor {get; private set; } through the Sense() method by performing a downward RayCast and retrieving information about the FloorTile.

Actuators

The Actuator performs the Cleaning operation on the Dirty Tiles that the Kaiju Agent has sought to and is currently standing above. There is no instantaneous clean as defined by the CleaningDuration for human-like behaviour. We only clean if we are above the tile, the tile is dirty, and it hasn't been cleaned yet.

There are two routines responsible for cleaning: tryToClean() ensures this tile meets all the requirements above, and CleanRoutine(), which performs the cleaning operation. CleanRoutine() ensures that if our KaijuAgent abandons the operation, then the tile is not registered as Clean, therefore not removed from our collection, and can be cleaned again.

Agent & Controllers

The Controller, FloorCleanerController, is the brain of our Kaiju Agent and extends the KaijuController Class. The FloorTiles are given three states and a short memory time to remember which state a tile is in and to remove invalid entries.

During testing of the Controller, I noticed that the corners of the Floor were often missed due to the search distance. To combat this, I decided to have the agent move in a circular pattern from the outer corners to the inner corners while wandering, and to stop when the Sensors no longer detected any dirty tile.

If this mechanism fails, a lower-level Wander() method prompts the Agent to seek randomly and perform the same functions, such as updating memory based on vision, removing stale entries, invoking the actuator to clean, and then continuing its random search.

Use of AI

The learning curve for using Unity and coding for this Assignment proved quite steep. In the first weeks, I focused on learning about the KaijuVisionSensor, testing RayCast distances, and iterating over the information from the visions. Once that was complete, I used ChatGPT for the Controller to fine-tune my Kaiju Code and implement the circular mechanism, as my initial model had a worldview that caused the Agent to pause multiple times due to the amount of information from the vision. This led to the addition of the

1. Stale Memory Timer
2. Fallback Method to wander randomly if SweepSearch failed

AI was also used for the FloorGroundSensor, as I initially had my sensor focused on my Kaiju and it didn't detect any tiles until all were dirty.

CODE WALK THROUGH

DEMO

QUESTIONS

THANK YOU

For your attention