Lab5 – AI Agents and Steering

CSE 3541 Games and Animation, The Ohio State University

Guidelines

- The grader will NOT overcome compiler or debugging errors for you.
- All make-ups for lab must be accompanied by a documented and verifiable excuse well before the deadline. Given the severity of the emergency please inform me as soon as possible.
- Lab submissions will NOT be accepted via email to me or the grader.
- ***All suspected cases of academic misconduct will be reported to the University Committee on Academic Misconduct for review.***

Objectives

The purpose of this lab is to learn more about behavior-based animation techniques by implementing algorithms such as spatial agent vision, movement, and interaction.

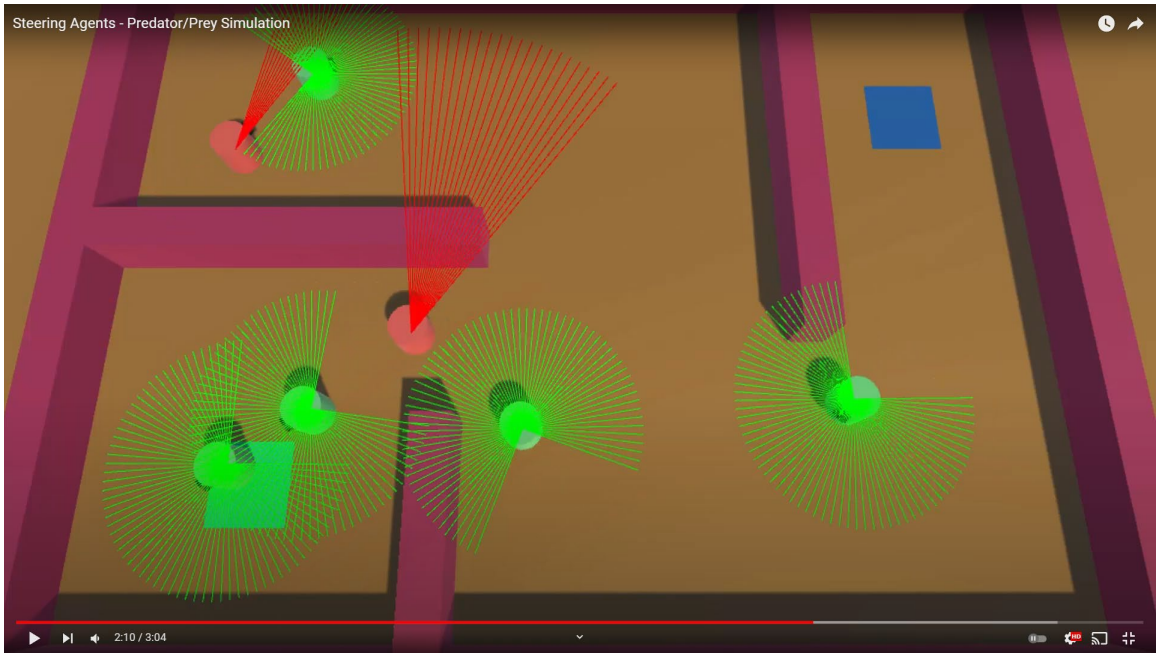

Figure 1. Screenshot from: https://www.youtube.com/watch?v=CdSzlp0OSzk

Assignment

You will grade will be based on original code. Any code that is not yours must be referenced in your solution and should be minimal.

The implementation details for this lab are open ended. The general task is to implement a 3D animation of agent's motion in two dimensions (in other words, use 3D GameObjects or models but only have the agents move in the x and z dimensions, then place a camera above the scene to give a top down/bird's eye view). The scene will contain two types of behaviors: predator and prey. Prey wander about the scene, when a predator sees prey the predator chases it (use seek or pursue behaviors), and the prey flees.

Start by implementing two classes, one for a predator agent and the other for a prey agent. At a minimum, each should have a current position (x, 0, z) and heading (either an angle or vector). You may use forces or kinematics to control the motion of the creatures. Give the two groups different capabilities (vision, acceleration, turning speed, etc.) to make the resulting animation interesting. Whenever a predator gets within epsilon of a prey (i.e. close or collides with it), the prey is eaten (disappears). You should include spatial limits on the environment (i.e. do not let the creatures outside of a closed arena).

You may want to **test** your implementation for predator and prey by having a player-controlled agent where in one scenario the agent has a predator behavior, and the prey interact with it. Then in another scenario the agent has a prey behavior, and the predators interact with it. This is not a requirement but would be great for testing.

You may use Unity components such as Colliders, Triggers, and possibly RigidBodies (although they shouldn't be needed).

Grading

Points will be allocated based on whether the scene displays the following features:

- Scene (10%)
- Spatial agents with behavioral motion (40%)
- Vision models for spatial agents (40%)
- Additional features, described below (10% each)

Note: You must do additional features to receive full credit. While the lab is still graded out of 100%, you can earn up to 110% by doing 3 additional features.

Point breakdown: since some implementation details are harder than others, here are some details on gradations for the tasks.

- Scene: Create an environment with obstacles that the creatures cannot move through. There should be some open areas, some "dead-ends", some choke-points and a procedurally animated set of moving obstacles – 10%

- Agent motion – 40% possible (collision detection between predator/prey and bounding box must be supported in all cases)
    - Predators and prey both wander randomly – 10%/40%
    - Predators chase prey when detected, prey ignore predators – 30%/40%
    - Predators chase prey when detected, prey try to avoid predators – 40%/40%
- Agent vision – 40% possible
    - Omniscience for predators and for prey – 10%/40%
    - Proximity sense for predators and prey – 20%/40%
    - Field of vision (cone and range) for predators, nothing for prey – 30%/40%
    - Field of vision (cone and range) for predators, your choice of vision model for prey – 40%/40%
- Additional features – 10% each up to 30%, i.e., 3 additional features. **Document each one in your report.**
    - Visible Field of vision for an agent – 10%
    - Add flocking or some other ruleset to have the prey interact with each other – 10%
    - Add a goal position or object that the prey tries to move toward (wanders towards) – 10%
    - Add a visualization that accurately shows the field-of-view of the agents. Note: it must be viewable in "Game" mode (e.g., Unity's scene display of a raycast doesn't count). To display the field-of-view, consider creating a triangle, cone, or several line segments. – 10%
    - Use Unity Record (or some other recording software) and create a video highlighting what you did, upload it to Youtube and include a link in your report. – 10% (Note: You will want to do this for lab6 as well)

Suggestions

Ideally, the placement of creatures and obstacles should be such that we can run your animation and tell what you've implemented just by looking at it. It might be difficult to have one scene with pathfinding and flocking, feel free to make multiple scenes with different scripts for different additional features. Alternatively, create a larger scene with different rooms (possibly closed) that show off your features.

Submission

Similar to what you did for Lab 1, you will submit this work on Carmen as a Unity package. Include your name in the filename of the package. Make sure the package includes all scripts and scenes (and images/textures and models if applicable) in it. Also include a pdf report describing the tasks you completed along with several screen shots. Also include any user input controls. As in previous labs, you can test to determine if your package can be graded by making a new Unity project, importing the package,

opening the scene file, and running the animation. You must submit work that runs in order to receive credit.