Multi-Agent Motion Planning
Check-in: Mar 16
Due: Mar 30 (11:59pm)

Overview
In this assignment, you will implement a basic motion planning framework that you will use to animate groups of agents navigating in complex environments with a variety of obstacles. The simulated agents should avoid collisions with both obstacles and other agents, while still reaching their goals. Additionally, you will explore methods for simulating group behaviors. You will develop several animations that show groups of agents displaying interesting emergent behavior such as flocking, herding or lane formation. You should incorporate a global navigation structure to allow your agents to navigate through complex environment without getting stuck at local minima.

Check-in (*Required*)**:**
Consider the following scenario:

> A 0.5m radius game character is in a large 20m x 20m room. The character starts at the bottom left (-9,-9) and wishes to go to the top right (9,9). There is a single obstacle in the room, represented by a circle of radius 2 meters at coordinates (0,0). Use a probabilistic roadmap (PRM) to plan a path for the agent from the start to the goal.

Implement the above scenario, in a graphical environment, with a smooth, continuously animated agent (e.g., represented as a cylinder). You should also visualize the roadmap the agent is following, including the start and goal positions, milestones, and edges.

Crowd/Flocking Simulation (Required) (80 Points)
Simulate multiple agents sharing the environment as follows:
-Implement a local interaction technique (Boids, Helbing, RVO/ORCA, TTC, etc.).
-Find 2 or 3 scenarios showing interesting interactions between the agents
-Implement a global navigation strategy for the agents (PRM/A*, RRT, etc.)
-Your roadmap needs to account for the extent of the agents, and should support multiple obstacles in the environment
-Show 2 ~~or~~ 3 scenarios of groups of agents successfully navigating through environments with local minima
-Find a scenario where your overall simulation breaks and produces odd behavior

Additional Features

30 {
(10) Implement and compare two different group interaction techniques
(10) Implement and compare two different global navigation techniques
(10) Nicely rendered 3D scenes w/models to give context (2D navigation is okay)

(10) Support full 3D navigation (e.g., birds flocking around 3D obstacles)

70+80

User Interaction
20 (5) Allow the user to add and move obstacles at run time
(5) Allow the user to dynamic choose agent starts and goals at run time
(15) Allow user to control some characters or obstacles in real time, simulated agents should replan or react dynamically to the user

Better rendering and animation of scenario
(20) Animate the agent as a walking virtual character (using a walk cycle)
(30) Load and render complex environments (e.g. quake or doom game level), and plan a path through the level.
(50) Blend clips from a mo-cap database to drive complex character motions through the environment.

5 *Faster motion planning*
(5) Implement A* for graph search, must document performance improvement
(10) Implement a spatial data structure (such as KD-tree) or some other to accelerate the nearest neighbor search for the PRM construction or neighbors search in crowds, compare performance to the brute force method
(15) Implement a spatial data structure (such as a BSP-tree) to accelerate the checking of potential collision with obstacles for building roadmap links or obstacle-neighbor search in crowds, compare performance to the brute force method

Better motion planning [can illustrate on a single agent]
(5) Implement path smoothing (e.g., walk to furthest visible node on path)
15 (10) Implement an RRT. Briefly compare the RRT to the PRM method in terms of generated path, and ease of coding.
(15) Implement the Optimal RRT algorithm (RRT*)
(20) Allow agents to rotate, implement a scenario where the agents must rotate to reach to goal. (Smooth the rotation for a small bonus.)
(30) Implement D* Lite, Lifelong A*, or any method where agents must explore the world as they navigates to their goal (the agent should only see nearby obstacles).

Game & Dance Contest *[you'll get points for only one or the other]*
(5) Make a game-like scenario involving the planning tool; best game is 10 points.
OR
5 (5) Animate agents flocking together in an artistic fashion; best dance is 10 points.

Scoring
-*Undergraduate*: Grade is √(totalPoints * 100) [e.g., 100 points will be full credit]
-*Grad students*: Grade is √(totalPoints * 84) [e.g., 120 points will be full credit]

What to turn in
Make a submission website, submit *just a link to this webpage*. The page should contain:

- Images & Videos of your animations
- Code you wrote & List of tools/libraries you used
- Brief write-up explaining any difficulties you encountered

You must call attention to anything you did which you are expecting credit for.

Hints
-There are a lot of little things that could go wrong. The concepts behind motion planning are easy, but debugging in hard, please start early!
-Circles navigating around circles (and rectangles with rectangles) have very simple configuration space obstacles. You may want to be flexible with how you represent your agents depending on what they are avoiding.
-A basic group interaction should be relatively easy to code, and you can likely build off code from Assignment 1 (particle system). The difficult part comes in finding parameters that lead to nice animations; make sure to save time for tuning!