Assignment 4

Due date: Tuesday, December 3, 2019, by 6:00pm

Note: Unless otherwise specified, late assignments will only be accepted with prior written permission of the instructor. You must explain all answers and show all work to get full marks! Please make sure your code is in a professional style: well-commented, properly structured, and appropriate symbol names. Marks will be very generously deducted if not!

Description

In this assignment you will develop some procedural content within a Unity simulation. This involves developing behaviour for different kinds of agent, based on the principles described in class. All agents should avoid collisions with obstacles and other agents, and move using some combination of steering forces. Use the same vehicle model of all agents, with turning and acceleration forces capped lower than braking.

Do not use external assets that provide steering behaviours. You must implement all agent movement yourself.

Note that the simulation builds in complexity. Provide one solution and include clearly indicated options to demonstrate the different sub-solutions.

1. A basic game level consists of a mall-like environment, in which agents will move based on steering behaviours. Build a game level following the overhead view of the design shown below. This consists of a large rectangular area with a few larger obstacles in it, a central “food court,” and “shops” on two opposing sides.

   The food court area should consist of an arrangement of 3–4 tables, several chairs, and 2–5 planters. The exact locations and arrangements of these objects should be generated randomly for each playthrough. In each case, however, it should look like a reasonable seating area for an eating area. Specifically, all seats should be reachable, seats should be near/around tables, tables should have seats near them, and planters should be placed throughout the space while still ensuring reachability. You may use any algorithm/approach you like for setting up the food court, and while the exact number of objects is not critical there should be a sufficient number to make pathing through it non-trivial.

2. Some shopper agents move through this space. They spawn on the left side of the mall, and move at slightly varying speeds, taking several seconds to cross the mall and generally aiming to exit on the right side of the level. With 50% probability, a spawned agent simply traverses the space, despawning once they reach the other side of the mall. The other 50% shop and eat: they should choose a random shop to visit, navigate toward the shop, enter it, pause for about 1s, leave and attempt to find a free seat in the food court. Once successfully “seated” (ensure no more than 1 agent per seat at a given time) they wait 2-3s, then leave, and head to the right to despawn.

   These agents need to somehow get to their intended destination(s), but their actual motion should be entirely controlled by steering forces, and this includes obstacle avoidance. In addition, however, you should use Unity colliders to guarantee (all) agents never overlap with each other or other obstacles.
Provide an input field/slider in the editor to change the spawn rate for such agents (lower bound of 0, upper bound up to you, but it should allow the simulation to be quite dense).

3. Other kinds of agents, advertisers, also exist in the simulation. The mall has a fixed size population of advertiser agents, created at random locations (not within the shops or food court, and not overlapping other agents). They move randomly through the mall area (and may traverse through shops or the food court), slightly faster on average than shoppers. They may go anywhere in the level, but should favour the relatively open, non-shop, non-food-court area. Advertisers dislike other advertisers, and thus try to avoid being too close to them, as an additional repulsive force beyond object/collision avoidance. Again, all agent motion must be a result of using steering behaviours.

Advertisers have two primary actions. The first action is to drop an advertisement (flyer), represented by a small object, at their location while outside of a shop. This can be done every $k$ seconds with probability $p$, and has no effect on advertisers, but a shopper who passes over (or within a small radius of) an advertisement consumes it, causing the shopper to instantly pause for about 2s before resuming their motion. During this period they are considered “flyered.”

A shopper who has been flyered becomes a target for advertisers. Any wandering advertisers within a (Euclidean) range of $s$ of a flyered shopper will attempt to head for that shopper to deliver their sales pitch—a process considered successful if the advertiser is able to get and stay within a distance $r$ of the shopper for more than 4s. Note that this is longer than the time the shopper pauses! An advertiser who successfully delivers 3 sales pitches despawns, and another advertiser may then be spawned to replace them. An advertiser who cannot reach (get within distance $r$ of) an intended shopper within 5s will give up and go back to wandering.

Use a visual representation for shoppers and advertisers that clearly distinguishes them, and which also indicates their state (flyered or not, and number of sales pitches given respectively).

Provide additional input fields/sliders in the editor to modify advertising parameters. This includes the observation distance $s$, sales pitch distance $r$, advertising rate $k$ and probability $p$, and the total number of advertisers in the simulation.

4. Provide a separate document that concisely and briefly describes which steering behaviours you use for each kind of agent.

**What to hand in**

Assignments must be submitted on the due date and time. Submit your assignment to MyCourses. Note that clock accuracy varies, and late assignments will not be accepted without a medical note: do not wait until the last minute.

For the Unity questions, hand in an exported project containing all files needed in order to reconstruct and run your simulations.

For non-Unity questions, submit either an ASCII text document or a .pdf file with all fonts embedded. Do not submit .doc or .docx files. Images (plots or scans) are acceptable in all common graphic file formats. You may submit a single document for such questions, as long as each answer is clearly delineated.

This assignment is worth 15% of your final grade.