Homework (Project) I: LR^{2}H

Due Dates: 10/3/19, 11:59pm (Submission on Blackboard)
Points: 5
Assignment Type: Paired

In this assignment, we will practice the basic single agent movement algorithms by creating an animated story or a game. A sample Little Red Riding Hood story annotated by these movements is attached. You don't have to follow this story; you can make your own or make an interactive game.

Requirements

- Though fancy graphics are not required, your overall project should be no worse than what is in the Assignment #2 Framework.
- As before, it should always be possible to tell where each of the character is facing -- which can be different from the direction they are moving. This is already in the Framework.
- You will need many of the Steering Behavior methods that you have already written, plus one new one:
    - **path following**.
- Add the Path Following this way:
    - Build a zig-zag path of some sort for Little Red Riding Hood to follow.

  - When it is time for Red to appear, make this path visible on screen and have Red follow it.
  - Her movement should be smooth and fluid, not robotically glued to the path or making hard, neck-snapping turns.
  - You will find a path (called *Paths*) already set up in Framework #2. You can use that, edit it if you like, or set up your own,
- Do something with the trees:
  - Disable the random tree generations but put a bunch of trees in the game (say, 20) in fixed locations.
  - From a coding standpoint (if you instance them in code), it is better to read their locations in from a file (CSV or JSON), so you can edit them easily, than to "hard code" their locations.
  - But it will be okay if you simply place a bunch of trees in the level via the Unity editor.
  - The main reason for doing this is so that random tree generation doesn't mess up the walls or Red's path.
- Wall and Obstacle avoidance should be in use.
- You can use the MapStateManager mechanism for stepping through the story.
  - As each stage ("scene") of the story completes, simple move to the next map state and present the next scene.
  - The whole story should play out from beginning to end once it is started, or, if clicking a number button, play that scene.
  - If 0..9 are not enough for the buttons, use A-F (see the story outline below).
- You need to display the names of algorithms when they are applied in your application, same as before.
- For each of the algorithms, try to add visual cues to demo how it operates. For example, for arrive using reduced speed, you need to draw the circle which when being hit, the character will start reducing speed. For pursue and evade, you need to indicate the expected location of the target. For path following, you need to show the path. For obstacles in your environment, show the ray casts or "whiskers" on the characters.
- These are the minimal requirements; you can add other behaviors to suit your game.
- Name your submission "LRRH_*your name*".
- If too large to upload, host it on DropBox or Google Drive and upload the link to LMS.

Little Red Riding Hood: The Game

As noted above, each of the items below are “scenes” that play out in sequence. Once the conditions for a scene are met (either by collisions or by using a timer or some other in-game event), advance to the next scene. Note: you will need to use “timers” to stUse the “Narration” UI to tell the viewer what’s going on.

1) Hunter appears (at a randomly-determined position of the screen)
   - After a few seconds Hunter wanders in the woods, avoiding trees and walls
2) Wolf appears (at a randomly-determined position of the screen)
   - After a few seconds Wolf starts wandering in the woods, also avoiding trees, etc.
3) Once Wolf and Hunter are close enough …
   - Wolf evades from the hunter
   - Hunter pursues the wolf
   - (play with the max speeds of both to get a nice result)
   - Both Hunter and Wolf disappear, by colliding
4) Red appears and walks to Granny’s house (<u>visible</u> path following)
5) Wolf appears …
   - Wolf pursues Red
   - Once they meet, somewhere on Red’s path, they both stop
   - (They talk for a few seconds), then
6) Wolf runs to Granny’s house (dynamic pursue with arrive using ***slow radius***), while Red continues on her path to Grandma’s
7) Hunter appears again (at a randomly-determined position of the screen)
8) Hunter runs to Granny’s house (dynamic seek with arrive using ***slow radius***) at a higher speed.

…

9) Have whatever scary surprise ending you want.