Homework (Project) II: Single Agent Movement

# Due Dates: 9/20/17, 10pm (Submission on blackboard)

# Points: 5

# Assignment Type: you can work in groups of 2-3 people

In this assignment, we will practice the basic single agent movement algorithms by creating an animated story or a game. A sample Little Red Ridinghood 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, e.g. create something like Agar (agar.io).

# Requirements

* Though fancy graphics are not required, your overall project should be visually pleasant.
* I should be able to tell where each of the agent/character is facing -- which can be different from the direction they are moving
* You need to use the following algorithms in your animated story or game -- **dynamic pursue with dynamic arrive, dynamic evade, dynamic wander, path following**.
  + These are the minimal requirement, you can add other behaviors
* You need to display the names of algorithms when they are applied in your application.
* For each of the algorithms, try to add visual cues to demo how it operates. In particular, for arrival using reduced speed, you need to show the circle where the character will start reducing its speed. For pursue and evade, you need to indicate the expected location of the target. For path following, you need to show the path.
* No obstacles in your environment yet, you will work on them in your next assignment
* Name your submission “singleAgentMovement\_*your name*”
* You will get .5 additional points if you can compile your homework into a web plug-in.
  + If you use Unity, turn in your complete project on LMS, and in addition provide me a website where I can find your project.

Hunter appears (at a randomly position of the screen)

Hunter (dynamically) wanders in the woods

Wolf appears (at a randomly position of the screen)

Wolf (dynamically) wanders in the woods

Once wolf and hunter are close enough

Wolf (dynamically) evades from the hunter

Hunter (dynamically) purses the wolf

(let wolf’s max speed to be a little faster than the hunter)

Both Hunter and Wolf disappear

Red walks to Granny’s house (path following)

Wolf appears

Wolf (dynamically) purses Red

Once they meet, they both stop

(they talked for a second)

Wolf runs to Granny’s house (dynamic pursue with arrive using ***slow radius***)

Hunter appears (at a randomly position of the screen)

Hunter walks to Granny’s house (dynamic seek with arrive using ***slow radius***)

…