RUTGERS

Unity: Navigation

In this assignment you must create a simple crowd simulator (navigation system), where you can select a set of agents (using a mouse interface) and can click on a desired location in the environment to have them navigate towards.

Useful materials and readings:

1. Unity Navigation Tutorials: http://unity3d.com/learn/tutorials/topics/navigation
2. Unity Navigation documentation: http://docs.unity3d.com/Manual/Navigation.html

Follow these instructions and create your simulator:

1. Design a relatively complex environment using different obstacle configurations. Your environment must include a simple maze field, several rooms, bottleneck areas, at least 3 height levels (e.g. a couple of containers on top of which agents can move/jump is considered a second level), and connections between levels (stairs, bridge, etc.).
2. Compute a navigation mesh for the whole environment (2D planes on different levels).
3. Create an "Agent" prefab with NavMeshAgent component to allow agents to navigate in the environment while using the navigation mesh.
    a. The Agent can be a simple capsule with a capsule collider for now
    b. You can instantiate multiple instances of the Agent prefab to create a crowd of agents in your scene
4. Create a simple mouse script to select agents and specify which destination to navigate towards. Read about camera rays to understand how to detect a mouse click position in your scene: http://docs.unity3d.com/Manual/CameraRays.html
5. Setup a "Director" script that keeps track of all the selected characters and sends messages to each character's component to move to the specified destination.

6. Use NavMeshObstacle to create obstacles in the environment which can be moved around. Obstacles should be selectable, and movable using arrow keys. The obstacles must carve the navigation mesh (check the carve checkbox in NavMeshObstacle).
7. Use off mesh links to connect disconnected navigation meshes to implement jump between different planes on different heights (hint: off mesh links can only be created between two surfaces, that means you may have to define a null object on the surface of another object).
8. Explain the difference (in behavior) between carving and not carving option for a NavMeshObstacle? When and why should the carve checkbox be active/deactive? Describe the problem with these two situations:
    a. if we make all obstacles carving.
    b. if we make all obstacles not carving.
9. Describe a way for implementing how an agent can avoid obstacles with not-carving option? (hint: a way was discussed in details in class).
10. **Extra credit:** Create several obstacles that automatically move in your environment (e.g. some roaming devils!), and make it such that your crowd avoids them! If you make them carve the navigation mesh, would there be any problem with your crowd movement in this case? What is the problem? What is the reason?
11. **Extra credit:** Add some “Nazgûl” agents! These agents can be selected and moved like others, but the other normal human agents should always avoid them at any cost, by a rather far distance! So for example if one of them enters a room, all human agents in that room should run away to other places, and when human agents are navigating, they should avoid these agents! Remember: Nazgul agents must be able to move seamlessly, like any other agent. Mention in your documentation how you implemented this part. (hint: it is possible to use both NavMeshAgent and NavMeshObstacle on same agent, but it is tricky to handle that, very tricky! some good/smart scripting is needed.)

Deliverable: A web playable demo of an interactive crowd simulator where the player can select one or more agents and issue commands of where they must navigate by clicking on a location in the environment.

SUBMISSION:

Submit the following in Sakai for grading:

1. Your Unity project in a zip file which contains the Assets/ folder and includes all your C# scripts.
2. Brief documentation explaining what you attempted, and answers to any questions listed above
3. A web playable demo of an interactive crowd simulator where the player can select one or more agents and issue commands of where they must navigate by clicking on a location in the environment.
4. Video demonstration of project

NOTE: Extra credit will be given at the discretion of the instructor.

--

Good Luck Everyone!