

ASSIGNMENT 4

INTRO, PURPOSE AND GOAL

1. To learn about pathfinding in games
2. To learn about raycasting
3. To learn about multiplayer games
4. To make a bigger project than you have done so far
5. You can work alone or in groups of up to 3 students in this assignment, bigger groups are expected to achieve more!
6. This must be a 3D game!
7. Be prepared to present your game in class

HAND-IN

Hand in before Wednesday 13th March, by pasting the URL to your github project next to your name in this sheet:

<https://docs.google.com/spreadsheets/d/1rekEFkgqSKu9matm9qia9eLW68W9Ph06gJlogIDCJ68/edit#gid=0>

USEFUL RESSOURCES

1. A*: https://en.wikipedia.org/wiki/A*_search_algorithm
2. A*: <http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>
3. A*: <https://www.ics.uci.edu/~welling/teaching/271fall09/InfSearch271f09.pdf>
4. Unity Navmesh: <https://unity3d.com/learn/tutorials/s/navigation>
5. Photon multiplayer: <https://assetstore.unity.com/packages/tools/network/pun-2-free-119922>
6. Photon tutorial 1: <https://doc.photonengine.com/en-us/pun/v2/demos-and-tutorials/pun-basics-tutorial/intro>
7. Photon tutorial 2:
<https://www.youtube.com/watch?v=phDySdEKXcw&list=PLWeGoBm1YHVgXmitft-0jkcTVhAtL9vG>
8. Unity Tower defense template: <https://unity3d.com/learn/tutorials/s/tower-defense-template>
9. Brackey's Tower Defense Game tutorial:
<https://www.youtube.com/watch?v=beuoNuK2tbk&list=PLPV2Kylb3jR4u5jX8za5iU1cqnpQPmbzG0>
10. Line Renderer: <https://docs.unity3d.com/Manual/class-LineRenderer.html>
11. Raycasting: <https://unity3d.com/learn/tutorials/topics/physics/raycasting>
12. Check for GameObjects within a radius:
<https://docs.unity3d.com/ScriptReference/Physics.OverlapSphere.html>
13. Tower Madness Zero:
<https://play.google.com/store/apps/details?id=com.limbicsoftware.towermadnesszero>
14. Playing of Tower Madness Zero: <https://www.youtube.com/watch?v=ZCm-cyBE28E>

ASSIGNMENT

GREEN (MINIMUM REQUIREMENTS & CURRICULUM)

1. Make a tower defense game. Make it from scratch or from some tutorial. Note that the tutorials linked here are quite complex and long, and parts might be deprecated. You will learn stuff from

how the tutorials do it, but you will also learn a lot by designing your own game. Also, note that even if you finish a tutorial, you are likely not done, you still need to finish all the green requirements listed below, so you will have to modify your template/tutorial project.

2. This is a classic tower defense game, so the player must be able to place towers anywhere on the map, and enemies can walk anywhere on the map. That means your game must be more like "Tower Madness Zero" than "Plants vs Zombies" or "Kingdom Rush".
3. If the player blocks all paths to the other side, enemies will just instantly destroy a tower so they can pass - or alternatively, the player will be prevented from putting down the final tower that blocks the path.
4. The player can make the enemies change direction by placing towers. In that case, an enemy must find another way around the tower(s), meaning the enemies must have adaptive pathfinding enabling them to recalculate the move path in runtime. You can use Unity Navmesh system (easier) for this or make your own A* pathfinding (quite harder). If you make your own A* implementation, then you don't have to make the recalculation feature.
5. The towers must shoot at nearby enemies.
6. Enemies start in one side of the map, and win if they can reach the opposite side of the map
7. It must be possible to upgrade the towers, so they shoot faster, harder, slow the enemies, or whatever you feel like the upgrade should do.
8. Players can sell or remove towers.
9. Make some visual effects, remember juice it or loose it
10. Now make a new game which must be an online multiplayer game. I don't recommend turning your tower defense game into a multiplayer game, just follow the standard Photon tutorial, or some other Photon tutorial. The tutorial you follow is up to you, it can be a simple pong game, a 3rd person shooter or whatever you like, just keep it simple. This is just to give you a chance to try to get some easy hands-on experience with multiplayer.
11. For multiplayer, I recommend you use Photon framework. You can also use Unity's own system called UNet, but it is being deprecated and replaced soon with a new system.
12. If your workload is too high, slack at the multiplayer part, not on the tower defense part.

YELLOW (GO FOR IT)

1. Try and make both projectile towers and laser towers. For laser towers, you could use LineRenderer and Raycasting
2. Go all in on the "juice" - make many tweens and cool animations, a beautiful UI with buttons, a main menu, a story/introduction, multiple waves of attackers, boss fights, etc.

RED (CHALLENGE YOURSELF)

1. Make your own pathfinding using A* or another algorithm, thus you don't use Unity's Navmesh system.