
Mid Term Test

Fall 2022

Student Name:	Section: 001
Student Number:	

Course: COMP-396

Date: Wednesday, Oct. 26th, 2022 Time: 2:30am – 4:20pm

Examining Instructor: ARBEN TAPIA

- **The 50 marks on this exam count for 25% of overall course marks (as per syllabus).**
- There should be 2 pages for the practical test (including this cover sheet).

MIDTERM TEST - PRACTICAL

COMP396 Fall 2022

Student Name: _____ Student ID: _____

. (2 pts) Create a new 3D Unity project named MidTerm396_<FirstName_LastName> (where you substitute your First and Last Name for <FirstName_LastName>). In addition to the given Scenes folder, add the folders , _Prefabs, _Materials, and _Scripts; create PlayerMat, EnemyMat and GroundMat in _Materials folder (choose suitable colors/textures)

. (2 pts) Add a plane named **Ground** with scale (2,1,2); add a Capsule named **Player** positioned at (-9,0,0); add a Cube named **Enemy** positioned at (9,0,9); add tags Ground and Enemy and apply them as well as the already existing tag Player to the respective game objects. Select a suitable position for the camera to view the whole ground from above.

. (8 pts) Add a C# script **Approaching** to the Enemy to make it "Approach" the player if the player is "visible" (consider the player "visible" from the enemy when the angle between the forward direction of the enemy and the direction to the player is less than 90 degrees) and closer than 8 meters. (Hint: Have a goPlayer property and either drop the player GameObject on its slot or use FindGameObjectWithTag method of the GameObject. The color of the Enemy should become **red**).

. (2 pts) Save the current scene as **ApproachingScene**, and as **GettingAwayScene** (effectively making another copy) in the Scenes folder; in the **GettingAwayScene**, remove or disable the Approaching.cs script from the Enemy and add a new C# script named **GettingAway**.

. (8 pts) Implement the GettingAway behavior in the **GettingAway** script for the Enemy: if the player has PowerUp (Boolean is true) and the enemy is less than 5 meters away from the player. The color of the Enemy should become **yellow**. Make Enemy a Prefab in the _Prefabs folder.

. (8 pts) Add an empty game object named EnemySpawner with an **EnemySpawner** script which should spawn Enemies "waves" every 10 sec (using the Enemy prefab). Make the number of enemies spawned per wave configurable. The enemies will either Get Away or Approach depending on whether the Player "has the powerup" or not.

. (8 pts) Add a **PlayerController** script to the Player and implement WASD movement of the player on the plane. Add a Boolean property named **HasPowerUp**; attach the script to the player.

. (8 pts) Outline how would you implement the above functionality with a FSM (4 pts) and add a drawing of your State Transition Diagram - either doodle it, or use an app like Visio or Draw.io or Graphviz.net (4 pts).

. (4 pts) Answer the following by filling in the blanks (underlines) (hereby: include this word document with your submission) (2 pts each):

. (1 pts) Give the definition of Big-Oh notation: **Given a problem of size n and two algorithms with costs $f(n) > 0$ and $g(n) > 0$ each, we say that $f(n)$ is in $O(g(n))$ when** _____.

. (3 pts) We have found an algorithm for problems with size n with cost f(n) as below. Find the "best" g(n) for which we can say f(n) is in $O(g(n))$:

. (1 pts) $f(n) = 1000000 * n$ \Rightarrow **g(n) =** _____.

. (1 pts) $f(n) = 2000 + n * \sin(n)$ \Rightarrow **g(n) =** _____.

. (1 pts) $f(n) = 5 + n * (n-1) / 2 + \log(n) * n$ \Rightarrow **g(n) =** _____.

Bonus points if you can prove (as in class) at least one of your claims above based in the definition of Big-Oh notation.

Zip the workspace and the .docx with the answers to the last question above and submit to eCentennial.

MTT Total Points: 50