

Immersive Media Programming – Lab 3: Unity scripting 2

In this lab, we continue practicing basic Unity scripting. We will keep using the same techniques that we practiced in Lab 2, and we practice these new things:

- Using CharacterController
- Using raycasting
- Applying explosion force

Submission instructions:

- For each task, record a short video that demonstrates all the required features of the task. Name the video according to the task, like Task1.mp4.
 - Place all videos in a separate folder OUTSIDE the project folder.
- Close Unity.
- Make a backup copy of your project folder.
- Remove unnecessary folders from your project folder: Build, Library, Logs, obj.
- Zip the project folder AND the video folder into a single ZIP file.
- Upload the ZIP file to your Google Drive shared folder.
- Submit a link to the shared folder in AjouBB.

Deadline: April 1 at 23:59

Maximum points: 16p

Task 1: Character Controller v2 (8.5p)

Open the CharacterControllerExample_template project (Lecture notes > Week 2) and finish it by following the lectures (end of 2020/03/14 and beginning of 2020/03/18) (3p)

Improve the scene:

- Change the pin game object's color when it is shot. (0.5p)
- Add a new game object to represent an enemy (e.g. find a model in Asset Store) (1p)
 - Enemies will be destroyed when they are shot
- Add another game object to represent a friend (e.g. find a model in Asset Store) (1p)
 - If you shoot a friend, it is game over!
- Create an obstacle course: different game objects that the player must run, jump and shoot through. (1p)
 - Add at least 10 shootable game objects (pins, enemies, friends, etc) and obstacles like stairs, slopes, walls, etc.
- Keep count of how many game objects the player has successfully "killed" by shooting. (1p)
 - "killed" means both destroyed enemies and knocked down pins.
 - Print out the number after each "kill".
- When the player is in the air (e.g. jumping, falling), use a raycast to measure how far the ground (or whatever is under the player) is. (1p)
 - Print out the distance to ground.

As before, be creative :)

(BONUS) Task 2. Timer (1p)

Continue your Character Controller v2 project:

- Add a countdown timer (e.g. 60 seconds) that starts when the scene starts playing.
 - When the time is up,
 - print a Debug.Log() message indicating how many of the game objects were shot.
 - Pause the game
 - Here is one way to do the countdown timer: <https://gamedevbeginner.com/how-to-make-countdown-timer-in-unity-minutes-seconds/>
 - This example uses the Text UI component to show the time. This is optional for you.

Task 3: More rayshooting (5.5p)

Demo: Lab3_Task3

Create new scene “Task 3”. In this task, you will implement “shooting” different Rigidbody enemies by using raycasting, like the demo video shows. Here are some basic rules:

- The camera is positioned so that the game view shows the game area. (0.5p)
- The player shoots enemies by clicking on the screen (raycast from mouse position). (1p)
- Enemies have different models. Download some nice assets, or reuse from previous labs. (0.5p)
- There are three type of enemies (Rigidbody) that respond to shooting in different ways:
 - BombEnemy: creates explosion force that affects the surrounding objects (see example in lecture) (1p)
 - SplitEnemy: destroys itself and instantiates smaller copies of itself (i.e. use a prefab). (1p)
 - When the smaller copies are shot, they are split into smaller pieces, etc.
 - Third enemy: up to you! Try to impress us :) (1p)
 - In my case, I created an enemy that jumps and changes its material or color (Renderer.material or Renderer.material.color)
- Implement different sound effects: (0.5p)
 - Shooting
 - Hitting each of the enemy types (different sound for each)

Hint 1: if small “children” of SplitEnemy fall through the ground, try this: ground > Rigidbody > Collision Detection > Continuous Dynamic.

Hint 2: the player script may need to access different enemies to call their methods. There are different ways to do that. Please see Class Notice about it. If you have questions or need examples, please let us know.

BONUS Task 4: Flying (1p)

Demo: Lab3_Task4.

Implement flying mode where the player (i.e. the camera) can rotate with mouse and move with WASD. Add a crosshair for easier aiming (like in Task1).

Hint: You can copy and modify the MouseLook script that you wrote in Task 1.

Task 5: Reflection (2p)

Write a short reflection text based on your lab experience (e.g. a text file or a Word file) using the questions below. The positivity or negativity of your answers does not affect scoring.

Please answer to these questions (at least):

- What did you learn? What did you know already?
- What was difficult in the assignment? How did you overcome it?
- What was good about the assignment?
- Do you have any suggestions to improve the assignment? Please tell us!

Save your reflections as Reflections.txt/docx to the root of your project.