# Assignment 2.2 - Task Report (Individual Component)  
## Cover Sheet  
\_\_\_  
### Team Name  
- (ID) Name  
- (ID) Name  
- (ID) Name  
...  
Class Time: Day-Time  
Tutor Name: Name  
\_\_\_

# Section 1: Skill/Task Summary  
- Q: Which skill/task is this report for? (Copy/paste the skill/item code here. It  
should really only include one skill per report. For multiple skills, have multiple  
reports)  
> A: PR08: Timers and delays: Make an object delete itself after a few seconds after being created. Implement this two ways: one with coroutines/threads/timers provided by the engine. The other way with a float variable that counts down based on time passing.

- Q: Did you have any experience or assistance relating to this task?  
> A: I had experience destroying an object, and I found out how to do a timer

- Q: How easy/hard did you find this task? Was it how you expected/was it weirder  
than you thought, going in?  
> A: It was relatively easy with having to count down and then destroy the object after the countdown reaches 0

- Q: Attach a screenshot of the Version Control system showing that the work assets  
have been ‘pushed’ to the server (i.e. all git commit/push logs, perforce logs,  
etc.) - the work can be finished/unfinished. If you also did time tracking on this  
task, put a screenshot of the time logs here as well.  
> A:

A screenshot of a computer

AI-generated content may be incorrect.

\_\_\_  
# Section 2: Logs and Notes

I first had to figure out how to make a countdown, I did this by searching online. After the countdown reaches 0, then a function needs to happen and within that function, the object needs to be destroyed. This is done by tagging the cube and then finding the tag and destroying it.

A trouble that I ran into was that the countdown couldn’t use an integer for the countdown and instead had to use a float.

A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

For the next part, I had to make a coroutine, this was tricky because it has its own way to be called and was a little more difficult to find resources on compared to the float countdown. However, eventually it worked.

# Section 1: Skill/Task Summary  
- Q: Which skill/task is this report for? (Copy/paste the skill/item code here. It  
should really only include one skill per report. For multiple skills, have multiple  
reports)  
> A: PR02: Object manipulation. React to user inputs (such as mouse click or keyboard/gamepad) to do each of the following, with different events for:: A) Hide/ show a game object. B) Create/instantiate a game object. C) Delete/destroy a game object.

- Q: Did you have any experience or assistance relating to this task?  
> A: I used various resources to assist me in doing the task

- Q: How easy/hard did you find this task? Was it how you expected/was it weirder  
than you thought, going in?  
> A: It was a little hard to wrap my head around it, I ran into a few problems but eventually found a solution.

- Q: Attach a screenshot of the Version Control system showing that the work assets  
have been ‘pushed’ to the server (i.e. all git commit/push logs, perforce logs,  
etc.) - the work can be finished/unfinished. If you also did time tracking on this  
task, put a screenshot of the time logs here as well.  
> A:

A screenshot of a computer program

AI-generated content may be incorrect.  
\_\_\_

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

# Section 2: Logs and Notes  
A problem that I had was that initially I tried to make the object invisible by using SetActive. Unfortunately, by doing this, it made it so that it wouldn’t be able to be turned on again. I overcame this by turning the renderer on and off. After this, I found that it was impossible to control if the object was turned on or off due to update working every frame and meaning that 1 click had to be less than 1/300ths of a second as it was running at 300fps. I overcame this by instead of using getkey, using getmousebuttondown.

For the later parts of instatiating and destroying, a part that I struggled with was trying to instantiate a capsule that didn’t exist, to get around this, I had to create an empty object that would exist in the scene and from here it would instantiate the object.   
A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.

# Section 1: Skill/Task Summary  
- Q: Which skill/task is this report for? (Copy/paste the skill/item code here. It  
should really only include one skill per report. For multiple skills, have multiple  
reports)  
> A:  UI Programming. Set up a basic canvas, and hook into the button click events (with the mouse, but you’re welcome to explore keyboard or gamepad events too).

- Q: Did you have any experience or assistance relating to this task?  
> A: In a previous task, I had to instantiate an object so using that information, I could already do half the task

- Q: How easy/hard did you find this task? Was it how you expected/was it weirder  
than you thought, going in?  
> A: I had trouble getting the button to work in the first place, after that it was making sure that after a button click, making sure a function takes place.

- Q: Attach a screenshot of the Version Control system showing that the work assets  
have been ‘pushed’ to the server (i.e. all git commit/push logs, perforce logs,  
etc.) - the work can be finished/unfinished. If you also did time tracking on this  
task, put a screenshot of the time logs here as well.  
> A:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.\_\_\_

# Section 2: Logs and Notes  
  
I already had previous knowledge on creating UI elements so finding out how to create a button wasn’t too difficult, after I created one, I ran into some trouble making the button be able to respond to clicks. After some research, it turns out that the canvas that the button is on didn’t have an Event System component on it so after that was added, the button could be clicked.

The script was a little different to what I have made in different tasks however was pretty simple to figure out. By creating a function that could be referred to by the button, a cube could be spawned.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

# Section 1: Skill/Task Summary  
- Q: Which skill/task is this report for? (Copy/paste the skill/item code here. It  
should really only include one skill per report. For multiple skills, have multiple  
reports)  
> A: AU01: Import sound files into the engine. Figure out a way to trigger one-shot sounds, and have looping sounds. You will also need to figure out how to trigger the sounds when the user interacts in the game with either a keypress or mouse click.

- Q: Did you have any experience or assistance relating to this task?  
> A: No

- Q: How easy/hard did you find this task? Was it how you expected/was it weirder  
than you thought, going in?  
> A: It was similar to some other tasks in some aspects but was totally different in others, I ran into some small issues that took a while to figure out.

- Q: Attach a screenshot of the Version Control system showing that the work assets  
have been ‘pushed’ to the server (i.e. all git commit/push logs, perforce logs,  
etc.) - the work can be finished/unfinished. If you also did time tracking on this  
task, put a screenshot of the time logs here as well.  
> A:

A screenshot of a computer program

AI-generated content may be incorrect.

\_\_\_  
# Section 2: Logs and Notes  
To start off, I imported a sound file into the unity project, after this, I created an empty game object to which I added an audio source component, the file was then attached to the audio source and then a script was made for the empty game object. After some research, I found the code that I needed to first reference the audio source and then play it. An issue that I had was that I had to define the audio source first and couldn’t just reference the audio source from start.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

To make the audio loop, I simply had to add [AudioSource Name].loop = true (in this case allclear.loop) and then play it. A screenshot of a computer

AI-generated content may be incorrect.

# Section 1: Skill/Task Summary  
- Q: Which skill/task is this report for? (Copy/paste the skill/item code here. It  
should really only include one skill per report. For multiple skills, have multiple  
reports)  
> A: VA01: Import 2D and 3D assets from an external program into a game engine, the asset must show in a level and be visible in the ‘game world’. 2D Image must show transparency. 3D Asset must be unwrapped, textured, and also show transparency.

- Q: Did you have any experience or assistance relating to this task?  
> A: I have previously used maya

- Q: How easy/hard did you find this task? Was it how you expected/was it weirder  
than you thought, going in?  
> A: It was more difficult than I initially thought, I had a problem with importing the textures in but I found the solution

- Q: Attach a screenshot of the Version Control system showing that the work assets  
have been ‘pushed’ to the server (i.e. all git commit/push logs, perforce logs,  
etc.) - the work can be finished/unfinished. If you also did time tracking on this  
task, put a screenshot of the time logs here as well.  
> A: A screenshot of a computer

AI-generated content may be incorrect.  
A screenshot of a computer program

AI-generated content may be incorrect.\_\_\_

# Section 2: Logs and Notes  
Tips:  
I had to first get a model that I already had so I got this banana model I made in maya last semester and added a texture to it. I then imported it into unity through the built in menu ‘send to unity’. However when it got into unity, there was no material on it so after some trouble shooting I had to check embed media but that still didn’t load any textures. I then found out you have to extract the material within unity and that then did work in loading the texture.

A banana on a computer screen

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a yellow object

AI-generated content may be incorrect.

For the second part, I had to import a 2D sprite. For this I found a grass billboard online and decided to use that. An issue that I had was that I couldn’t just drag an image in so I tried to texture a plane but that didn’t allow for the transparency. I then looked online and then found out how to make it into a sprite and that worked fine. A screenshot of a computer

AI-generated content may be incorrect.

# Section 1: Skill/Task Summary  
- Q: Which skill/task is this report for? (Copy/paste the skill/item code here. It  
should really only include one skill per report. For multiple skills, have multiple  
reports)  
> A: VA06: Create one permanent/looping particle system, and one ‘one shot’ particle system, within the engine itself. The particles must demonstrate transparency, no shadows, and a custom texture/image, and motion. This needs to be visible in the running game.

- Q: Did you have any experience or assistance relating to this task?  
> A: No

- Q: How easy/hard did you find this task? Was it how you expected/was it weirder  
than you thought, going in?  
> A: This one was surprisingly easy and intuitive to make happen.

- Q: Attach a screenshot of the Version Control system showing that the work assets  
have been ‘pushed’ to the server (i.e. all git commit/push logs, perforce logs,  
etc.) - the work can be finished/unfinished. If you also did time tracking on this  
task, put a screenshot of the time logs here as well.  
> A:

A screenshot of a computer

AI-generated content may be incorrect.  
\_\_\_

# Section 2: Logs and Notes  
First, I had to add a particle system as an object in the editor, after this I duplicated it to create one ‘burst’ one and one loop one. By default, the particle system loops and to make it burst, I had to untick loop, make the particle system as short as possible and tweaked the emissions to only throw out 1 burst.

A computer screen shot of a banana and plants

AI-generated content may be incorrect.

A computer screen shot of a banana and seaweed

AI-generated content may be incorrect.

VA02: Import and work with sprite sheet animations into a game engine. Sprite sheet  
must loop in its animation, and demonstrate transparency. Also look into how to  
make sprites not loop and only play once. Spritesheet must be visible when the game  
is running.

VA10: Explore ‘Rigid Bodies’ in the engine and build a structure that, when the  
game is played, has objects falling to gravity and show collisions and physics. Eg.  
A see saw/balance beam with objects of different sizes/masses falling on them at  
run time. Also demonstrate rigid bodies that are ‘kinematic’ (i.e. they do not move  
but things still collide with them.

PR04: Input handling. Most game engines have input managers. Instead of hard coding  
things like “if KeyPressed(Key.Space)”, use the input manager for your event  
handling (i.e. ‘if ButtonPressed(“Jump”)). You must be able to demonstrate how to  
get analog inputs (eg. X/Y/Z axis) as well as button inputs (ABXY etc.)

PR05: Physics programming: Manipulate and move an object/player character using  
forces in the rigid body system of the engine. Must be able to move around such as  
WASD, and also jump, and ‘respawn’/reset position.

PR07: Physics programming: React to a collision event and create an object at the  
collision location. A sound effect needs to also play when the collision happens