Interviewer template

Patch 2b – Week 4 – CMP424

Dr William Kavanagh & Dr Naman Merchant

This submission consists of 15% of your overall module grade

TODO: Fill in all sections marked in **[red]**

**Interview Questions for Picture Particle**

*Either select one of the example jobs from week 1 (available on MLS) or a games job online with clear expectations, including a link to the listing.*

**Job Role**: Junior Generalist Programmer  
**Student Name**: Callum Myers

**Instructions:**

For the chosen job role, the following interview questions assess the desired technical and problem-solving skills, using real-world scenarios. Each question is accompanied by:

* An **example answer,** demonstrating an excellent response – i.e., given by a hireable candidate.
* **Prompts and guidance** to help the interviewee respond effectively, highlighting the desired skills and competencies.

*A suitable example is provided…*

**Example Question:**

In a top-down shooter game implemented in C++, which data structure and patterns would you use to store and manage bullet objects, and why?

**Key Skills Examined:**

Design patterns, efficiency, data structures.

**Example Answer:**

I would use a std::vector to implement an object pool to manage bullets. A vector provides dynamic array functionality, meaning it can be resized without significant memory overhead. Using indexing I can access elements in the vector in constant time ( O(1) ). Furthermore, vectors store elements contiguously in memory which is beneficial for cache locality and efficient iteration.

By using an object pool, I minimise the allocation and destruction of objects, returning them instead, which is a cheaper operation. This is particularly important if bullets are going to be frequently created and destroyed as it may heavily effect performance.

**Prompts and Guidance:**

* Suggest to the candidate they speak about complexity and big-o notation. Is it possible to do this is mostly constant time? (yes, it is).
* Ask about using an array, would it be suitable for this and if not, why not?
* Are there any design patterns the candidate would look to use for object management such as this?
* Try and get them to discuss the problem specific to bullets (creation/destruction) and what could happen if they were not managed properly.

**Question One:**

How would you go about planning and implementing a maintainable player character controller in Unity?

**Key Skills Examined:**

Class structure, modularity, efficiency, implementation

**Example Answer**:  
I would first assess the mechanics required to make the character controller, which would include input, movement, camera, world interaction and physics, as well as any abilities or mechanics specific to the game. I would then create scripts to handle each of these separately, since this would help improve simplicity if any expansions or changes were required for a specific aspect of the controller in future. These scripts would use modular systems such as the Unity Input System, which allows a singular variable to refer to multiple input sources, for example using if (“Jump” == true) could check for a touchscreen tap, keyboard key or controller button and have the same result for all. This would be especially useful if the game were to be released on multiple platforms later down the line since the basic input handling system would not have to be fully revamped to allow for other inputs.

I would use a state machine which constantly checks for a player’s action at a given time, for example if they are running, jumping, idle, on the ground or midair, etc. and this would be used to decide if the player’s desired next action is possible based on their current state. This would, for example, prevent the player from jumping infinitely while midair, or blocking whilst shooting. This can be especially useful when working with animations, since each animation can be linked to a specific state. It would also improve performance as not all checks would be required in every state, since any checks relating to entering the current state or states which cannot be accessed from the current one could be ignored.

Finally, I would iterate over my created controller once mechanics have been implemented and assess where performance could be improved. This could mean minimizing the number of times physics calculations occur as these can be computationally expensive, reusing objects or animations where possible to minimize the amount being loaded into the game or scene, and ensuring any components which are not required are disabled or unloaded, such as having a player’s gun be a separate component and disabled when not being shot.

**Prompts and Guidance**:

* How could you make sure it would be simple to change specific areas of the code in future?
* Unity has some built in systems which can be very useful when building modular code. Where and how could you use these?

[Describe the specific skill, knowledge, or competency you want the interviewee to demonstrate.]

**Question Two: [Insert question here]**

**Key Skills Examined:**

[Comma-separated list of up to five topics.]

**Example Answer**:  
[Provide a relevant, detailed answer showcasing strong knowledge, practical experience, and job-related problem-solving abilities.]

**Prompts and Guidance**:

[Describe the specific skill, knowledge, or competency you want the interviewee to demonstrate.]

Repeat as necessary until 6-10 suitable questions have been given.