**2.1 Describe what each of these technologies are and what they can be used for:**

**2.1.1 Scriptable Build Pipeline**

It is a preview package of Unity which changes the way in which the content of asset bundles is managed in its construction and loading process, it can be used instead of the traditional system of construction and loading of assets in the traditional way.

**2.1.2 Scriptable Build Pipeline**

It is a technology that allows modifying the parameters and rendering flow of a Unity project using the scriptable object architecture in which, all the parameters can be managed from scriptable type files and can be used to create and modify different types of quality and exchange this information as if they were cards, in addition to speeding up the rendering processes, the clearest examples of the use of the scriptable render pipeline are HDRP and URP.

**2.1.3 Addressables**

It is a technology that allows to efficiently manage the resources of a game, which are not included when generating the build of a project and are loaded in real time, the resources can be textures, audios, materials, 3D models, scenes among others. The operation is very similar to that of asset bundles although it is simpler, lighter and faster to implement. It can be used in compiled weight reduction processes and in DLC or similar download processes.

**2.1.4 IL2CPP**

In a general perspective, this technology allows become the C# code in a native C++ compiled with the implement of the second layer of IL (Intermediate language) applied in this conversion. The process only includes the data used by game target platform and allow Access to the many C# features.

**2.1.5 Nested Prefabs**

Nested prefabs belong to the new work flow of the prefabs in unity, now can create big prefabs that contain other little prefabs in nested hierarchy. The little prefabs can be used of the independent way, this feature contributes to granularity of the system.

**2.2 Mention at least two problems of Unity’s non-incremental Garbage Collector.**

* The non-incremental garbage collector is executed in specific moments with a method, this process can take a long duration in main thread.
* This process stops the system while the garbage collector is running and this can be generated non uniform FPS fluxes and lag perception.

**2.3 Explain which of these is better and why? Unity LTS, TECH release, Beta or Alpha**

Tell which is the best version of Unity is relative, because the alpha and beta versions are important for the grow of the engine and the participation of the community in the validation process and early detection of the bugs. These versions are recommended by game developers and enthusiasts that wish to know the new features in developing, otherwise, for game developers and companies that want to create prototypes using new features validated in alpha and beta versions, it can be use the Tech version. Finally, for companies and game developers that want to create production Projects it is recommended to use de LTS version because it has long term support.

**2.4 What is your preferred version control system and why do you prefer it over others?**

I have been using only three version control systems, Subversion, Git and Unity Collaborate. I like Git because it is more flexible and more complete than Subversion and Collaborate, it allows managing of branches, conflict solve is very good, big files handle, and more.

**2.5 What is your favorite IDE and why?**

I have used some IDE software, in my perception I believe that the IDE to select is according to the language and the target platform of the Project though it also depends on the perspective of the developer and his wishes, for example, if I want to program in Java for web or desktop platforms, I like to use Eclipse, but if I want to write Web scripts or Python, I use Visual studio Code, while to work in Unity, I prefer Visual Studio Community.

**2.6 What issues or limitations have you recently experienced using Unity?**

I don't remember any specifically.

**2.7 What strategies or best practices can be used to optimize the CPU and GPU usage in an application made with Unity?**

There are different techniques that can be used to optimize the consumption of resources at the level of memory, CPU and GPU, among them is the use of design patterns in programming, something like the pool system to avoid consumption by creation and destruction Instances is great, while in integration processes, using static batching, GPU instantiation in the materials, use of LOD systems help to keep rendering times low, audio compression and texture override according to the platform contribute to the weight of the binary and ease of resource management on the different platforms.

**2.8 How do you catch and investigate crashes happening in a released game?**

To detect errors, I use the Unity connection with Visual to debug step by step, I use the printing system with Debug. Log in simple cases, I use a profiler to detect bottlenecks and in build cases the use of logs. In the specific case of Android, I use the Android Studio console to do the debug trace directly in the build.

**2.9 Compare the following function and macro definitions. In what cases will they produce different results and/or side effects?**

In C # you cannot use macros as such, but in C ++ the difference is that macro definitions are preprocessed and independent of the compiler, while functions are compiled to be used, on the other hand, the definition is not a function like such as it tends to "replace" text with an expression which can be translated at runtime as text, function, operation etc.

**2.10 What is the package manager in Unity and what is the alternative way of adding a package than via the package manager UI?**

The package manager is a system that allows the import of assets to unity projects, such as unity packages (shader graph, addressable, Input system, etc.), some Unity preview packages (Jobs, ECS, etc.), purchased packages through the asset store and even packages using Git URLs. Another way to carry out this import is through thew unity package files .unitypackage in the Assets -> ImportPackage -> custom Package menu.

**2.11 Examine the following function. What does it accomplish?**

int someFunction(int i)

{

int n = 0;

while (i)

{

i &= i-1;

n++;

}

return n;

}

Numero Binario Resultado

3 = 11 2

5 = 101 4

7 = 111 6

9 = 1001 8

11 = 1011 10

13 = 1101 12

27 = 11011 26

51 = 110011 50

55 = 110111 54

According to my experience, the function doesn’t execute because in C# an integer cannot be used to evaluate a While cycle. However, the line with the bwise operator “&=” analyzes the value of the variable in binary representation, the result shows the result of this analysis, for example.

In some tests generated to this line can be conclude that operation (i &= i - 1) was executed when the value of the i starts in binary format with 1 and ends with 1 in the last part of the representation.

Numero Binario Resultado

3 = 11 2

5 = 101 4

7 = 111 6

9 = 1001 8

11 = 1011 10

13 = 1101 12

27 = 11011 26

51 = 110011 50

55 = 110111 54