Sunday Unity Integration Developer Test

The project is built for Android and the apk “FinalBuild.aplk” is located in the folder “sunday-unity-integration-developer-test\Build”. I had to verify the api target and minimal version because I was getting some errors.

Critical Issues approached

MyEventSystem Class

MyEventSystem class is unable to find GameAnalytics required to send level start and finish events. Although the GameAnalytics unitypackage containing the script for handling events has been imported, it is not functioning as expected.

Solution

To find the issue, I had to first get the project to compile. For that, I commented out the lines in the script that were using the missing package. I tried setting up a Game Analytics account and completing the general setup but it didn’t solve the problem. I started looking inside the project files for any hint of something unusual or out of order.

I found an assembly file in the same folder of the class script that was blocking the plugin for Game Analytics because it wasn’t specifying the plugin. I tried to get a reference to the plugin dll but I couldn’t find a way to do it. My solution was to delete the assembly definition as it’s only compiling one file and it’s not necessary at this point. If there were more files in the compilation hierarchy, it would be necessary to correctly reference all plugins and assemblies.

Performance Issues

Despite the game having minimal objects and scripts, performance issues persist on mobile devices.

Solution

I checked every script developed for the project and it’s true that coding is minimal, so there is no optimization issue on that side. What’s left is to reduce quality settings in the project settings. I did that by selecting Medium Quality as default for Android. This will reduce shadow, textures and LOD quality and general rendering costs. I verified with the Stats tab and the FPS setback when a level is loaded was reduced so this is working as well.

Frame Rate Dependency

Controls behave inconsistently depending on the frames per second (FPS) the game is running at.

Solution

This problem is typically associated with the Update() method for Physics related routines. In this case, the solution was to change the routine to FixedUpdate() so that we don't calculate the ball position depending on the frame rate. This way, we do it with a fixed period.

Git Repository

There are issues with the git repository as numerous irrelevant files are included in pushes.

Solution

Git problems are related to the inexistence of a .gitignore file. When I created the GitHub repository, I made sure to add a .gitignore file to avoid pushing unnecessary documents. I also had to add a Large File System for the latter issue with the Firebase Analytics package.

Bonus issues approached

Firebase Analytics Integration

Firebase Analytics integration is missing. I need to send the same events (start, fail, and finish) tracked by Game Analytics to Firebase too.

Solution

I followed the documentation and tutorials to add the firebase analytics into the game. I also had to update the Log Events in the MyEventSystem methods so that we use both analytics packages.

Level Management

Adding and managing new levels within the project is challenging.

Solution

The level management was suboptimal. It relied on manually indicating the current and next level in the inspector of every object in every level. What I did was to use the SceneManager.GetBuildIndex to create a level sequence. Now, both Enemy and FinishTarget scripts get a reference to the current level and look for the next one in the build list so that we can replay the same one or go to the next level if we hit the target. In the case of the last level, I included a boolean that makes the game go back to the first level instead of looking for the next one in the build list.