THIS IS MY JOURNEY: A DEVELOPER'S PORTFOLIO

ABOUT ME

Hi! I'm Can. 6+ years old programmer (since from uni I'm developing something) and 2+ years old in professional in games. I always had strong interest for game programming and the learning the development stages of a game. Then, I started to develop and design games. I love making games. I love creating art. I play my guitar when I'm not working and listen music a lot. I love fitness and I like to read books.

I'm not only making games. I also developed real world applications using Flutter and other technologies such as Spring Boot, Java, Python Flask and C.

My First Journey: Travel MG

February 6, 2022

ABOUT THIS GAME

Travel through Europe's most beautiful cities and put your memory skills to the test with this fun and addictive memory game! Match pairs of cards featuring stunning photos of European landmarks, streets, and more. With varying levels of difculty, this game is perfect for all ages. Play now and start exploring Europe in a whole new way!

Playstore Link:
https://play.google.com/store/apps/details?id=com.cmindgames.travelmg&hl=en_ZA&gl=US

DEVELOPMENT

This game is published just a few weeks ago. It's a simple memory card game. It is developed by GameMaker Studio 2. GameMaker Studio 2 was much more powerful than Unity for small sized games. It was completely an old school tool. Later, we went on deep about graphics. We wanted to keep it simple but sweet. Textures were a problem because most of the game graphics were must be directly downloaded in-game. In-game textures, which would eat up to lots of GB's from memory because you can't organize textures and cut the whole atlas in pieces while in game, we would have to find a work around. We came up with something that could first download whole things then parse .PNG fİles and after that it could create a whole 1024x1024 blank sprite that would have been used as a texture page. Then, we put whole downloaded small sprites into it and called our sprites directly from that big sprite. This solved our texture page eating too much RAM problem and game was working fine between 8-10 MB RAM on Windows. Because of Dalvik in Android, it was around 30-40 MB.

Travel MG - Gameplay

While in development process, we have found that GameMaker's drawing system was just simply a wrapper of OpenGL. We focused on OpenGL to explore and what can we do more for our game. I went down with OpenGL for lighting and rendering. Then with shaders. I have written so many shaders as experiment for our game. We have used Naughty Dog's Uncharted 2 HDR rendering system. In main menu and gameplay, we have used different kinds of lighting shaders in backgrounds. It was a small sweet game but we put our souls into it. Whatever I'd be doing, I'd be doing it's best way!

Graphics programming with GameMaker is a bit more complex than Unity. We were decided to create this game for 1080x1920 screens fİrst but then later changed it to 1080x2376. These were the measurements that we used them as basis for other screens. Here is the key part was how we were going to handle aspect ratio problem. We did our own research and found a solution to it. Now, the only thing that left was designing screens. We first designed them in Photoshop then we put sprites and objects and etc. exactly to their location. We had to test it so many times and I don't want to even tell the whole truth that when we change something in screen we would have to re-calculate coordinates of things to make sure everything was in correct place. It consumes almost a day that if we would have used a scene editor, this game would have finished earlier. We even thought of writing our own basic scene editor as a plugin to GameMaker but this would take too much time and we couldn't cut the development progress in it's half. I think one of the most downs about GameMaker is that the engine has no scene editor. If you want it, you have to write for yourself.

Travel MG - Gameplay

For GUI and texts, it would be so bad to have too many draw calls. It would affect badly to performance. So, then we decided to use graphic files as texts. So simply, this would have decrease the number of draw calls. A bunch of glyphs that creates a text graphic was simply just a one draw call! This was meaning using disk much more but whole files were 20-30MB in total or maybe less.

CONTACT

E-mail: cpehlivan544@gmail.com