Michael Isaza

Misaza.net

Michael.Isaza1@gmail.com

WORK EXPERIENCE

Programmer for Platforms and Platform Tools, Hi-Rez Studio - Alpharetta, GA | November 2017 - Present

- Worked on Smite, Paladins, Realm, and RogueCompany creating and documenting engine changes and tools.
- *Automated Character/Map Performance Test* - Wrote code for each game in C++ that would capture the performance of every character skin, weapon, and map. It was developed for all our performance PC's as well as Xbox, and Switch. Once the reports were made, the data would then be formatted into an easy to read way in python and sent out to all the character artists, working on that game title, using javascript The reports drastically improved the character and environment artist's abilities to identify which skins or maps needed to be optimized. After seeing these reports the UE3 and UE4 teams launched an optimization pass on the skins and maps that the performance tool reported as over budget.
- *HLSL code* - Modified the base UE3 HLSL (DirectX) code to expand the use of our vertex shaders.I had to learn how to read and debug HLSL code, while working on this project. Despite this, I was able to modify the vertex shader to work with a random instancing, in a little over a week. I worked with the build engineer to implement my changes in a seamless manner. The next day after completing this task, our team and the artist were able to create new and performant materials. My team and I helped the artists implement the material inputs in 3DS max due to time constraints.
- *Debugged UE4/UE3 Engine Code* - In the past our team would have issues tracking down engine errors without the help of an engine programmer. I learned how to debug the C++ Unreal Engine 4 and 3 code base, so that our team can solve very specific problems like a texture streamer memory overflow or fix broken light map generations. Once I was able to solve problems like that on my own, our team could share those fixes with other game teams in an effective manner. Learning this increased our team's overall efficiency and platform teams production output.
- *Artist Developer Tool* - Coded a windows application in python called SCut that helps speed up the workflow for our artists. The tool was a series of buttons that would open developer apps found using a Perforce library. The apps you could open were dependent on the current workspace and game the user set which could be changed within the app itself. Before this tool, many of the leads had to spend time teaching an artist how to use the command line and which specific arguments to add for certain applications. After this tool was deployed almost all artists switch over to using this tool and all new artists onboarded use this tool from the start.
- *Playtest Performance Reports* - Originally one member of our team would setup a performance playtest one to two times a month with QA. This task was time consuming to coordinate a playtest for four games and format all the data collected. I created a tool in Jenkins, python and powershell that would do this task for every playtest. The lead QA would launch the program on Jenkins and data would be collected automatically . After the tool finished, it would send out a playtest performance email to all lead's for that game. Having this tool saved us from spending weeks of planning and data formatting, while providing more runtime performance data for the teams to look at.
- *Optimized Existing Code* - Some of the previous code was high on execution time, especially in javascript where we format and send out performance emails. Performance reports would be missed, because the amount of data we processed went over the execution window. I solved this issue without removing any data and created a clever way to bypass the execution time window.

The Last Pixel Gameplay Programmer, Spiregg - Marietta, GA | May 2019 - November 2019

- *Enemy AI* - Developed the AI for the starting 2D world and assisted with the final boss AI. The AI has editor parameters that let our designer change the way an individual AI instance behaves. We could place these instances anywhere on the scene to quickly design a level in a matter of minutes.
- *Player Controller* - Our game has three different sections with three different player controllers. I wrote the player controller and physics for one of the 2D worlds and the 3D that works with keyboard and controller. Our designer was able to modify parts of the player controller for each scene to give variety to the gameplay without needing to access the code. We were able to test and change things very quickly with this setup.
- *Shader Code* - Godot's shader code is similar to GLSL ES 3.0. I wrote the shader code for our materials in our 3d world. I made the shader code as accessible as I could for our team, so that we could access and modify input variables in the gd script. This allowed us to rapidly test material parameters without going back to the shader code and compiling them.

Programmer, Puzzles by Joe - Marietta, GA | September 2015 - February 2016

- *Clutter* - Our team took an existing game made in C++ and created a web version in C# with Unity.
- Revived old C# code that our contractor previously had someone write. We used some parts of this code for the user input, but I changed how puzzles were made and solved. With these changes we were able to create 20 puzzles in minutes rather than hours and puzzles generated faster..
- I managed a team of three people. I divided up tasks for each of us to work on and we would help each other pick up slack if we fell behind our timeline. We ended up delivering the project earlier than the client requested.
- Our client also wanted to learn how to use Unity and C#. We would set up a meeting where we would go over our code and teach him our workflow. After we finished the project, the client was able to understand how Unity worked and could create games in it.

Programming Languages

C++, Python, Javascript, Batch, C#, Java, HLSL, Powershell, Assembly, Lua

Skill Sets

UE4, UE3, Unity, Godot, Perforce, 3DS Max, Maya, DirectX

Education

Kennesaw State University *Marietta, GA*
Masters of Computer Science, *Spring 2019*

Kennesaw State University *Marietta, GA*
Bachelor of Computer Game Development, December *2016*

Extra

- Did programming consulting for Emory's UE4 VR research.
- Won the Center for Disease Control sponsored Game Jam in 2014 with a Unity platforming game.
- Won the Virtual Reality/Augmented Reality tract with Unity at the Kennesaw State Game Jam in 2016
- Presented Unity Hololens game at SiegeCon Atlanta, in Georgia 2016