Greg MacGown and Josh Rosenthal

December 3, 2019

C# and Beat Saber

Abstract: This paper will examine the history of the C# programming language, including a brief history, some key features of the language, and how it differentiates itself from other languages. Then, the paper will briefly discuss the programming aspect of the project, creating a Beat Saber mod and how C# is related to creating mods for the game.

Introduction

This paper is the culmination of the CSIS340 term project where we were tasked with studying a programming language outside of class in a similar depth to ones we have studied in class. The language that we chose to study is C#, as it is one of the prominent languages in game design and shares similarities and differences with Java and other C languages. The programming aspect of our project is using C# code in Unity to create a mod for the game Beat Saber, a rhythm game which uses virtual reality to slice through bricks that progress towards the player's screen in a pattern similar to the rhythm of a song playing in the background.

C#: A History and Why C# instead of other C Languages?

C# 1.0 was released to the public in 2002 as a modern, object oriented, type safe programming language. The most recent version of C#, 8.0, was released earlier this year (History of C#). C# was designed to be a more similar language to Java, intended to be a competitor to Java, as opposed to C and C++, but still uses C and C++ as a base for syntax. C++ and C# are also both compiled languages, as opposed to Java being an interpreted language. This means that C++ and C# create executable files when they compile, similarly to Java's class files which are run on the Java Virtual Machine. Despite these similarities, C# is very different from C and C++. C#, similarly to Java, has garbage collection, and has no need for memory allocation like C and C++. C# complies its code into byte-code, which creates a much larger overhead and can cause efficiency issues when it is compared to C and C++, which complies into machine code, making it a much more lightweight language. C# does not use pointers anywhere in its code, while C++ makes heavy use of them. C# also does not require the usage of header files.

C++ programs favor applications that work directly with hardware or that need better performance than other languages can offer. Examples of C++ programs include networking, gaming, server-side applications, and even device drivers for your PC. C#, on the other hand, is generally used for web, mobile and desktop applications (Marsh).

Beat Saber and How C# Ties In

Beat Saber is a virtual reality rhythm game that features the player slicing blocks representing musical beats with a pair of red and blue lightsabers. The player gains points based on how many consecutive note blocks they slice, the difficulty, and the accuracy. The game is developed and published by Beat Games.

Beat Saber mod code tended to be created and tested in the Unity Engine, which favors C# code due to its simplistic features, garbage collection, and protected nature. Beat Saber has a very active modding community so we knew there would be resources going into this project to help me when we didn't understand something. The Beat Saber Modding Group wiki has a tutorial on how to create a basic mod which displays how well you do in game scoring wise. We followed this guide to get familiar with Microsoft Visual Studio and get a basic understanding of what kind of code we would be writing.

Because Beat Saber is a game made by an official studio, the code of the game itself is quite difficult to understand without looking at game code previously. We used the program dnSpy to look at the game's code and see what we could reference from it. dnSpy shows every class the game uses and each method in that class. Patches can then be written to change the code of the game, modifying or completely changing the game's code.

The most important feature of C# utilized was the reference feature, which allowed us to reference other objects and files directly, something that other C languages cannot do as easily. This allowed us to reference many other dll files such as BeatSaberCustomUI, which allows us to quickly create a button in game with a specific function.

```
     public void OnSceneLoaded(Scene scene, LoadSceneMode sceneMode)
             {
                 if (scene.name == "MenuCore")
                 {

CustomUI.MenuButton.MenuButtonUI.AddButton("Quick Reload Songs",
"Reloads new songs", delegate {
SongCore.Loader.Instance.RefreshSongs(false); });
                 }
```

```
                }
```

This specific example adds a button in game, and when that button is pressed, the SongCore mod will reload all new maps put in the map folder after the game was launched. Being able to reference other mods was also a very helpful tool because the mods are usually publically available on GitHub where they can be examined.

The example code above, while difficult to read at first, ended up becoming simpler once the similarities to Java code, which we have worked with extensively in the past, became more apparent. Knowledge of features like garbage collection and the protected nature of C# assisted us greatly during the course of this project.

Microsoft Visual Studio was also very useful because it autofilled code suggestions and there is a community template available which can be selected to auto fill a basic plugin structure. Once a mod is finished, we can press the build button or use the shortcut to build the dll file. This file can then be put into the plugins folder of the game where the mod loader, BSIPA, will activate the mods and apply any changes they make. A logging capability can also be built into the mod to indicate where a problem occurs or when a specific step with the mod is finished.

Conclusion

This project has given us a valuable insight into the nature of C languages and game development as a whole. A learning goal for a future project would be how similar other games would be to mod, and the scope of the Unity Engine.

Citations

“Beat Saber Modding Wiki” https://bsmg.wiki/modding Access Date: 11/13. Multiple Authors

“C# vs. C++: Which Language is Right for Your Software Project” https://www.upwork.com/hiring/development/c-sharp-vs-c-plus-plus/ Access Date: 11/17 Author: Jennifer Marsh. Publish Date: 5/25/2019.

“Features of C#” http://ecomputernotes.com/csharp/cs/features-of-c Access Date: 11/24 Author: Dinesh Thakur

“History of C# Programming” https://www.c-sharpcorner.com/blogs/history-of-c-sharp-programming-language Access Date: 11/15. Author: Bikesh Srivastava. Publish Date: 1/5/2017.

“The history of C#” https://docs.microsoft.com/en-us/dotnet/csharp/whats-new/csharp-version-history Access Date: 11/19. Publish Date: 9/19/2017

“Unity Scripting Languages” https://www.lynda.com/Unity-tutorials/Unity-scripting-languages/365280/379703-4.html Access Date: 11/15. Author: Kelley Hecker. Publish Date: 3/24/2015.