

CS SENIOR CAPSTONE TECH REVIEW

NOVEMBER 14TH, 2017, FALL 2017

CDK GLOBAL: NO MORE TOUCH. NO MORE KEYBOARD. BRING IT ALL TOGETHER. USING TECHNOLOGY TO TEACH HUMANS.

PREPARED FOR

CDK GLOBAL

TREVOR MOORE

Signature

Date

PREPARED BY

GROUP 9



LOOK BOSS, NO HANDS

BRANDON DRING

Signature

Date

NIPUN BATHINI

Signature

Date

CARL BENSON

Signature

Date

Abstract

Here we will look at 3 pieces or technologies rather, of things going into our project.

CONTENTS

1	Server Side Scripting	2
1.1	Overview	2
1.2	Criteria	2
1.3	Node.JS	2
1.4	Java	2
1.5	Python	2
1.6	Discussion	3
1.7	Conclusion	3
2	Communications	3
2.1	Overview	3
2.2	Criteria	3
2.3	Polling	3
2.4	Websockets	3
2.5	RabbitMQ	4
2.6	Discussion	4
2.7	Conclusion	4
3	Graphics Engine	4
3.1	Overview	4
3.2	Criteria	4
3.3	Unreal	4
3.4	Unity	5
3.5	CryEngine	5
3.6	Discussion	5
3.7	Conclusion	5

1 SERVER SIDE SCRIPTING

1.1 Overview

The server is the unsung hero of this project, while everyone might be entertained by the voice assistant use, and the VR immersion, the server is what really ties everything together. Because for any interaction to come from the virtual reality headset, or the voice assistant, the computing and instructions have to be done on the server in the cloud before being sent to a local machine to be played back out.

1.2 Criteria

- Server side scripting language should be flexible and robust enough to be used everywhere.
- Easily learned or used by all.
- Vast ecosystem of modules to install easily, instead of having to code our very own utility modules.
- Support in the AWS Cloud
- Proper documentation and guides regarding any packages used.

1.3 Node.JS

Node.JS is a server side scripting language implemented in Javascript, part of the JavaScript everywhere approach. One of the big perks of using NodeJS, is that it comes with its own package manager that is incredibly simple to use, NPM. There are currently about 500,000 NPM packages that do everything to building an HTTP server to using frameworks such as ReactJS. Most things in NodeJS are also abstracted away from the developer, instead of worrying about the boilerplate code that needs to be written to create a Http server, you merely just call the package itself and start your server. On top of all that there is superior documentation for running JavaScript servers on AWS, more specifically the Lambda section in which the Alexa computations rely on.

1.4 Java

Java is a programming language owned by the Oracle Corporation, and is probably one of the most battle tested languages in the world. Along with being the most popular programming languages in the world for a variety of languages. The main reasons being it is entirely Object Oriented, fast, runnable everywhere, and secure. However the Java language is also very verbose to most people, to do simple things like access an object, one has to wade through object notation. It also comes with the overhead of having to compile the language before running it. And its not the simplest compile process either, it comes with a complex directory structure, byte-code files, and .JAR files as well.

1.5 Python

Python is an open source general purpose programming language used to really do anything from server side scripting to statistics to data visualization. Python is also known for being one of the more english-like programming languages as far as readability is concerned. Meaning that the team could both use it, and learn it relatively quickly. Python is also a mix between Object Oriented like Java is, more so than NodeJS, but also less than Java in general. However, Python is also known for being pretty slow in terms of computing speed, due to the fact that it is an interpreted language.

1.6 Discussion

NodeJS in comparison to both Python and Java, is sort of a middle ground to them all, it is faster than Python, slower than Java. More readable than Java however, less so than Python. Again, in contrast to Python and Java, NodeJS is the middle ground of complexity and general overhead of general use. All three of these languages can be ran on the AWS cloud and can do the tasks that we need, its more of a matter of ease of use and comfortability.

1.7 Conclusion

We chose NodeJS due to largely the reasons listed above, its a middle ground between the perks of both Python and Java. Largely though, we are choosing it because there is abundant documentation and guides on how to use NodeJS inside the AWS/Lambda cloud, such that Amazon actually even recommends it. Another perk, all though not a deal breaker is that all of us have rather extensive experience in NodeJS. Thus, cutting out the learning curve of trying to pick up a language and implement with it.

2 COMMUNICATIONS

2.1 Overview

The biggest hurdle of our project is merely just getting an interaction from our voice assistant, whether it is the Amazon Alexa, or the Google Home, to register to our VR headset and populate and environment. This is where the messaging comes in, what we need is something to connect from our cloud/server to broadcast down to the local machine with a signal on what to load or do next.

2.2 Criteria

- Support between programming languages
- Ease of connectivity
- Assurance of message delivery/receive
- Easy to read documentation
- Support in the AWS Cloud

2.3 Polling

Perhaps one of the oldest methods of communication, polling is nothing more than just a client asking another service if anything has changed. The perk of this is that it is incredibly simply to implement and develop on ones own. There are no libraries required, or anything external for that matter. You merely just set a loop to constantly keep asking the network for an update. However, as one could probably guess, this is an incredibly inefficient way to do things. There is also low overhead for the client to use polling, however the server that it is polling from may become overloaded relatively quickly.

2.4 Websockets

Websockets are a method of a TCP connection that is a HTTP Protocol. With a Websocket, each client that is connected to a server must go through a handshake which identifies the client to the server. Then the connection is upgraded to a binary direction protocol which then leaves the realm of HTTP protocols. From there the server will only give the client information as programmed. So one of us could programmatically tell the server to send a message to our local machines should an input come in from our voice assistants.

2.5 RabbitMQ

RabbitMQ is a messaging protocol software designed for message brokering. RabbitMQ specializes in the reliability of delivery a message from a host to a client. In our scenario it would be our server down to our local client machine. They also have the perk of putting messages on a queue should the connection drop off. So whenever the client or host reconnects, they will then get all the messages that were waiting to be received.

2.6 Discussion

Originally I had prior experience with websockets, but it wasn't until this assignment that I had heard of RabbitMQ. In contrast, the issue with websockets is should the handshake fail or the connection drops off, the default for websockets, or at least the version we were going to use is that they just result to polling. On top of that there isn't a guarantee that the messages will be delivered reliably.

2.7 Conclusion

We will choose RabbitMQ due to the reliability of the messaging protocol that it does have. Along with the added benefit of message queuing should the connection or handshake fail, that doesn't result in simple polling which can overload the server and cost money. But the biggest benefit of RabbitMQ is that you can monitor your connectivity, or at least can. While the actual RabbitMQ server can be set up in the cloud as well to ensure 100

3 GRAPHICS ENGINE

3.1 Overview

The graphics engine is the framework that helps the developer build a 3D video game. The graphics engine helps abstract away all the collision detection between objects, sounds from the game, frame rendering, etc. The proper graphics engine leads to increased productivity, and minimal learning curve.

3.2 Criteria

- Ease of use
- Learnability
- Low overhead (Memory, CPU, etc.)
- Support for HTC Vive
- Drag and Drop UI

3.3 Unreal

The Unreal Engine is the most successful video game engine, it is written in C++, and features a large degree of portability for the games it uses. It was originally used as a game engine for first person shooters, but since has expanded beyond shooters, and is used for MMOs or RPG games. Unreal Engine is also cross compatible between operating systems. They also have rich features such as AI within the engine, and being able to edit code inside a VR environment

3.4 Unity

Unity is framework written in C#, the perk of Unity, is that it is much more configurable and programmable than Unreal is. While Unity is also seen as easier to learn as well due to the complex nature of C++ used in Unreal. Unity has another perk being that it has an asset store which can be used to download packages to be used within the editor. Also, the HTC Vive that will be used has out of the box support for development with Unity.

3.5 CryEngine

Cryengine is another framework that is written in both C++ and C#, one interesting perk of using CryEngine is that is in Amazon already, just under the name Amazon Lumberyard. However the hallmark piece of the CryEngine is that it allows the developer to jump straight into the design and environment from the text editor.

3.6 Discussion

In comparison to both Unreal and CryEngine, Unity has their API written in C#. Unity is also naturally supported on the HTC Vive without having to jump through extra hoops in contrast to both Unreal and CryEngine. While Unreal is also known for its ease of learnability as well.

3.7 Conclusion

We're going to choose Unity because largely that is what is supported out of the box for the HTC Vive. So much so that the HTC Vive recommends Unity for development specifically. Unity also has the advantage of being written in C#, which connects with the .NET framework support of RabbitMQ as listed above. Making the communication aspect between the VR environment and the server that much simpler, not searching for some third party library in C++ to do the same. While Unity also has one of the more configurable and customizable APIs for building graphics. As our project incorporates data visualization, which to my knowledge hasn't been done yet. We could probably use all the customizations as possible.