NOVEMBER 8, 2019

SPONSOR: JASON ROBINSON

MENTOR: ISAAC SHAFFER

TECHNOLOGICAL FEASIBILITY ASSESSMENT

DIGITAL ROLL

Tyler Boice Caleb Johnson Tyler Malmon Brandon Measley

Contents

1. Introduction

Millions of people around the world play Tabletop Role Playing Games (TTRPGs) as a fun way to interact with friends and express themselves creatively. These games are played using various sized polyhedral dice that represent various probabilities in the players' stories. This style of gameplay and storytelling has enthralled numerous players and inspired countless adventures throughout the years. These games have become so popular, they have spawned many web applications (e.g. Roll20, which has over 4 million registered users). Today, players do not have to be in the same room to play these games. One can play with anyone, anywhere. Technology has allowed TTRPGS to expand how it can be played and TTRPGS have become popular than ever.

Our client, Dr. Jason Robinson, is someone who has been captivated by how and why we tell these fantastical stories. Dr. Robinson is a web UI/UX designer and founder of the company Beautymark Design Studios. Beautymark focuses on creating expressive interfaces that convey meaning to the user and lead them through a webpage, as if they are unraveling a story. When combining his love for storytelling and TTRPGs, Dr. Robinson found a conflict; There are no programs that truly merge online with in-person player by allowing online users to roll their own set of dice. If an online player wants to roll their dice, they can, but must verbally tell the group the results. Since the rest of the group has no way to verify the result of the roll, they have to trust that the player is being honest. This may cause skepticism about a player's honesty and result in conflict within the group. There are programs that can generate a random polyhedral dice result; however, serious players cherish their own unique dice set and it is important they can roll them.

The focus of Digital Roll is to help resolve this conflict by developing a pipeline to create machine leaning (ML) objects that classify the value of various polyhedral rolls. The application of this pipeline means players could roll a dice, a camera would capture the result, the application would translate the results and broadcast it to the entire group. This keeps online players honest about their rolls, while also allowing them to use their own personal dice. Additionally, this solution will allow all players to have an accurate log of their previous dice rolls for their own reference. This project aims to merge the physical realm of rolling one's own dice, with the utility of logging what values have already been rolled

2. Technological Challenges

Before we start the project, we must perform an assessment of the technological challenges that we know we will face. After analyzing the project, we have determined that we facethe following four challenges:

1. Selecting a language to use as our pipeline wrapper class
2. Selecting a machine learning framework to assist in creating the machine learning model
3. Developing the ability to recognize the hardware and software the user's phone
4. Tracking the changes to our program as it is being built

3. Technological Analysis

After researching each of our main challenges, our team has selected potential solutions for each challenge and evaluated them based on objective metrics. From the metrics we have determined a solution for each of our technological challenges.

a. Pipeline Wrapper Language

In order to ease the development of machine learning algorithms, our team plans to create a workbench to facilitate the creation of Apple Core ML kernels for classifying pictures of polyhedral dice. A major component of this workbench is the language that will be used to create the foundation.. This is the pipeline wrapper language and there are numerous languages that work well with machine learning systems. In order to choose the most fitting language for the workbench, we have chosen to use the following criteria which are listed in order of priority:

- **Functional API**: Since our project will also implement a ML framework, the language chosen must have a supported application programming interface (API) provided by the machine learning framework chosen. This is a mandatory requirement as otherwise the language will not be capable of supporting the machine learning process.
- **Team Familiarity**: The language should be familiar to the team if possible. Languages that the team is not familiar with may slow down development and create an unnecessary burden; however, this requirement is not mandatory. We may still select a language the team is not familiar with, if that language is the superior option. Team familiarity will be judged by conducting a poll where each team member will rate their familiarity with the language between zero and five. The average of these ratings will serve as the languages' team familiarity score. Zero represents a language that is completely unfamiliar to the team, while five represents a language that the team is fully familiar and comfortable working with.

The languages that met the criteria of being updateable, and machine learning system compatible, were Python, C++, and Java. Each of these languages have supported APIs in

MXNet, TensorFlow, and Pytorch – our candidates for ML frameworks. All these languages are commonly used, updated and possess useful features for our pipeline Machine Learning.

3.1.1 Python

Python is an object-oriented language featuring numerous libraries. Python features interactive and interpreted programming and is usable as an extension language for applications not written in Python. Python's latest version is Python 3.8.0 which was updated as of October 2019.

Python is the basis of prototyping of TensorFlow before it moved to C++, making it the best choice for compatibility and efficiency in that system [2]. MXNet also has a Python API that serves as the package's main language default option [3]. Python also runs at effectively the same speed as C++ as tested by coders during May of 2019 [1]. Python received a team familiarity score of 4.25 which means the majority of the team find Python to be very familiar.

3.1.2 C++

C++ is an extension to C which adds classes to the language. The design of C++ focuses on embedded systems and system programming with efficiency and flexibility. C++ is standardized by the International Organization for Standardization and its latest standardized version C++17 was ratified in December 2017 [4].

C++ has the advantages of being very streamlined making it useful in systems with limited resources which could allow our pipeline to run very effectively inside mobile devices as well as on regular computers. MXNets main API is C++ which ensures compatibility as well as efficient performance [5]. C++ is also directly supported by theTensorFlow API but is marked as experimental and does not have all the features of other supported language APIs such as Python.

MXNet does have C++ support through a built in API of the system. C++ received a team familiarity score of 2.25 which means that most team members do not feel very familiar with C++, but that they have used it before.

3.1.3 Java

Java is a class based general-purpose object-oriented language. Java utilizes a virtual machine to run itself allowing for write once run anywhere usage. Java's most recent version is Java 13.0.1 which was updated as of September 2019.

Java's feature of a virtual machine that can run on various devices is incredibly useful for our pipeline which may utilize mobile devices and non-mobile devices. MXNet as of 2019 features a stable Java API though its effectiveness needs to be tested [6]. TensorFlow also features a Java API, but it is marked as experimental and unstable. Java's team familiarity score was 4.50 which is slightly higher than Python's score and shows that one teammate was more familiar and comfortable with Java than Python.

3.1.4 Our Choice

Table 1 reflects a summary of our candidates for our pipeline wrapper and the objective metrics collected. Using Table 1, we have decided the best choice for the pipeline wrapper language of our product is Python. Although Java is slightly more familiar to the team Python offers much more advantages thanJava. In addition, Java only has experiment support for TensorFlow while Python is the main supported language for TensorFlow and MXNet which grants the team flexibility with either choice [9].

Language	Team Familiarity	TensorFlow support	MXNet support	Additional Features
Python	4.25	Full support	Full support	Simple to use. As fast as C++ for MXNet. Main supported language for MXNet and TensorFlow.
C++	2.25	Experimental only	Full support	High efficiency and low resource consumption.
Java	4.50	Experimental only	Full support	Write once, run anywhere.

Table 1: wrapper language candidates and their objective metrics

b. Machine Learning Framework

To create our pipeline, we need a tool that can help us create a machine learning model. This is what a machine learning framework does. Due to the advancement of machine learning in the last five years, there are many open-source machine learning frameworks that can be used to develop this functionality. The challenge is selecting the best framework for our project. The following criteria, in order of importance, is how we will measure the best framework for our pipeline:

- **Compatibility**: The framework that we choose must be compatible with the language that we have chosen to write our pipeline wrapper. In addition, one of our environmental constraints is our product must have the ability to be converted to Apple's Core ML. Therefore, the framework should be easily converted to Core ML. We will measure ease of convertibility by whether the framework has an Apple certified converter.
- **Validator**: It is very important that the framework has a built-in validator. If we select a framework that does not have a validator, then we would need to build one to validate our results. Therefore, the framework must have a built-in validator

- **Popularity**: When a framework is popular and has more mainstream appeal, then it is likely to have more support, tutorials, and content. Popularity is also important because it means that there are likely lower-level developers with the same skill set as our group is using this framework. There are many ways to judge popularity; however, we must learn the framework we select in detail, so we decided the most important metric in determining popularity is the quantity of online tutorial articles. We will also feel that the number of contributors on GitHub is also a good representation of popularity, because it demonstrates the size of the community.
- **Efficiency**: Ideally, we want our product to be as fast and as light as possible. To measure this quantity, we will evaluate the benchmark tests on the training and inference performances of the frameworks in question. Since our product should be faster at inferring than training, we will focus more on inference speed and utilization. Our benchmarks will be performed by the same computer with each framework. The fastest framework will execute the test it in the least amount of time. We will also take into consideration the percentage of CPU usage to test how intensive the framework is to run. The most efficient framework will have the lowest speed and usage. If the framework does not have both, we will focus more on the speed and less on the usage.

After researching the following criteria, we were able to select three ML frameworks that we felt would be best given our requirements. These frameworks were selected because they each have a built-in image classification validator, are relatively popular, and can be easily converted to Core ML. The following are the frameworks we thought best fit our project, in order of best to worst fit.

3.2.1 TensorFlow

TensorFlow was developed by Google to accelerate machine learning and deep neural network research. TensorFlow gets its name from the multi-dimensional arrays, or tensors, it takes in as input. Since its release in 2015, Tensorflow has been constantly evolving and updating [24]. The most recent change being the creation TensorFlow 2, released on September 30, 2019[22]. For this reason, TensorFlow is used by beginner software developers as well as big companies like Google, Intel, GE, Deepmind, and Twitter [12].

Compatibility

TensorFlow is compatible with Python and in addition has an Apple supported converter from TensorFlow to Core ML; therefore, it is easily convertible to Core ML [21].

Validator

TensorFlow does have a built-in validator [12].

Popularity

In late 2018, data analysis was conducted using 11 data sources across 7 distinct categories to gauge framework usage, interest, and online tutorials [16] Figure 1 and 2 are the two categories that we will focus on (GitHub contributors and articles written). these two figures, we can see that TensorFlow is the most popular ML framework for both categories. Tensorflow has 1,642 contributors in GitHub and over 6,200 online tutorial articles [13].

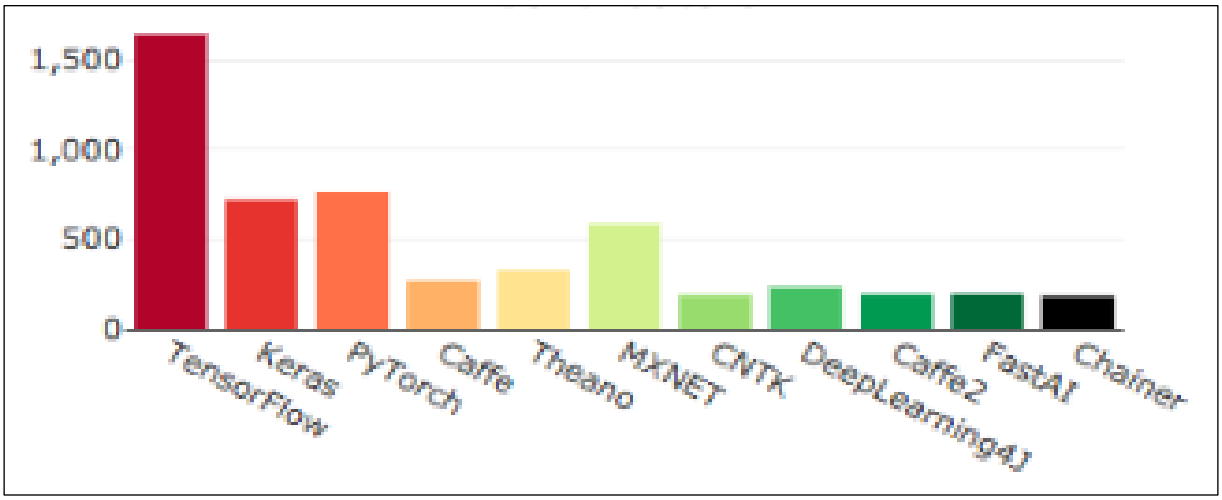

Figure 1: Total contributors on GitHub as of December 2018

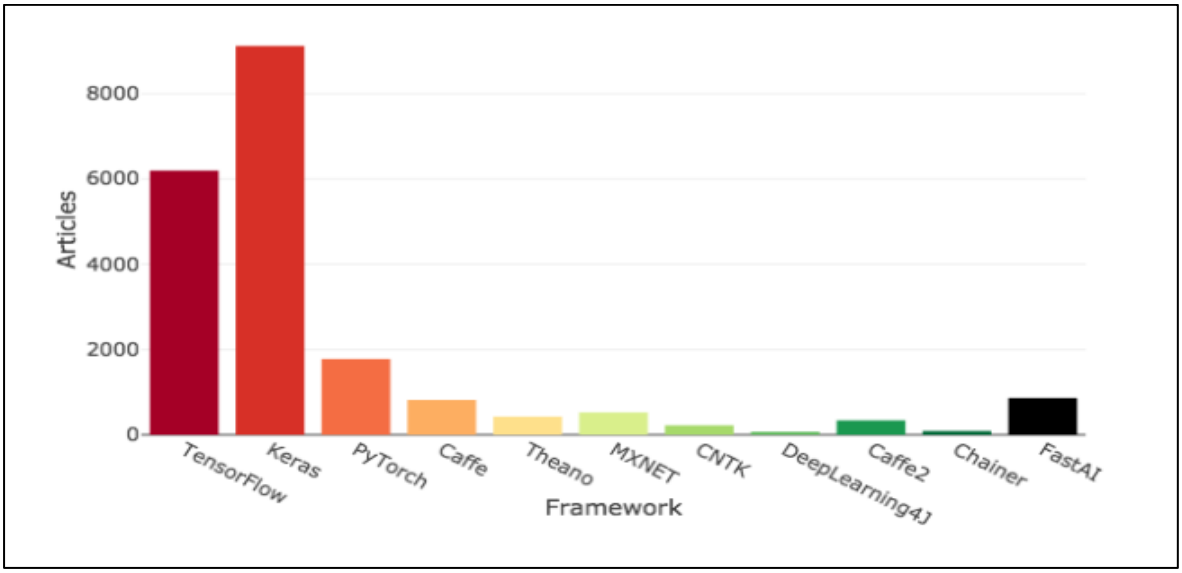

Figure 2: Total amount of online articles as of December 2018

Efficiency

Tables 2 and 3 show the Inference and training performances for TensorFlow, PyTorch and MXNet, which was ran using the computer. Looking at the inference data, TensorFlow uses a small amount of CPU at 0.84%. TensorFlow also ran the inference benchmarks at a rate of 8.93

seconds an image. TensorFlow also has impressive Training benchmarks with 0.72 CPU utilization and a speed of 3.92 steps a second [16].

	GPU utilization	Memory Utilization Time	Memory Used/Total (MiB)	CPU Utilization	Memory Utilization	Memory Used/Total	Speed (images/sec)
TF-1.4	71%	44%	5110/24190 MiB	0.84%	45.12%	7210/15703	8.93
PyTorch-1.0	48%	17%	7188/24190 MiB	4.3%	22.01%	3455/15703	22.22
MXNet-1.3	91%	39%	10396/24190 MiB	0.99%	23.36%	3693/15703	6.85

Table 2: Inference performance benchmark test on ML frameworks, using the same computer

	GPU utilization	Memory Utilization Time	Memory Used/Total (MiB)	CPU Utilization	Memory Utilization	Memory Used/Total	Speed (steps/sec)
TF-1.4	73%	32%	9714/24190 MiB	0.72%	38.51%	6028/15703	3.92
PyTorch-1.0	74%	31%	3502/24190 MiB	3.12%	45.51%	7146/15703	8.04
MXNet-1.3	88%	25%	5780/24190 MiB	0.55%	25.43%	3963/15703	2.85

Table 3: Training performance benchmark test on ML frameworks, using the same computer

3.2.2 MXNET

Apache's MXNet is a strong contender for our project because like TensorFlow, it is widely used. Amazon has chosen MXNet as its deep learning framework of choice at Amazon Web Services. MXNet is supported by public cloud providers including AWS and Microsoft Azure [6]. MXNet's first stable version was released at the end of 2017 and have since updated to its most recent stable update, 1.5.0, released June 8, 2019.

Compatibility

MXNet is supported for many languages including our wrapper language, Python. In addition, MXNet also has an Apple certified converter to Core ML; therefore, it is easily convertible to Core ML[21].

Validator

MXNet does have a built-in validator [6].

Popularity

Using Figure 1, we can tell that MXNet is one of the more popular ML frameworks. As of December 2018, MXNet has 587 GitHub contributors and 260 online articles [13]. Since we are applying more focus to the online articles, MXNet does not have a lot of online tutorials compared to other ML frameworks.

Efficiency

Using Tables 2 and 3 it is clear to see the MXNet truly does have the best speeds for both training and Inference performances at 6.85 seconds an image and 2.85 steps a second. MXNet also has an impressive CPU utilization at .99% for inference and .55% for training [16]. Although we are focusing on the inference benchmarks over the training, it is good that MXNet has good results in both categories

3.2.3 PyTorch

Released in January 2016, Facebook's PyTorch is an increasingly popular framework due to its similarities to TensorFlow and its ease of use [11]. Many developers have noted that it is far easier to use than TensorFlow and that with every version, it is gradually better and faster. PyTorch's newest update 1.3, is supposed to be TensorFlow's biggest competitor [14].

Compatibility

PyTorch is supported by Python. In fact, since it was built for Python, it can leverage all the services and functionalities offered by the Python environment [11]. Unfortunately, it is not easily convertible to Apple's Core ML. This because PyTorch does not have an Apple supported

converter. Instead would have require the use of a third-party system to convert PyTorch to ONNX, then to Core ML [23].

Validator

Pytorch does have a built-in validator [11].

Popularity

Using Figure 1, we can see that PyTorch is a very popular ML framework. Since December 2019 it has 1,560 online tutorials written about it, and 760 GitHub contributors [13].

Efficiency

Using Tables 1 and 2, it is clear to see that PyTorch lacks efficiency. Using the inference test data from Table 2, PyTorch more than quadruples the CPU utilization of its competitors at 4.3%. In addition, it takes over three times as long to test at 22.22 images a second. [16].

3.2.4 Our Choice

	Compatibility		Validator	Popularity		Efficiency with Inference Tests	
ML Framework	Python	Core ML	Built-In	GitHub Contributors	Online Articles	Speed (sec/Image)	Usage (% CPU)
TensorFlow	Yes	Yes	Yes	1,642	6,200	8.93	0.84
MXNet	Yes	Yes	Yes	587	260	6.85	0.99
PyTorch	Yes	No	Yes	760	1,560	22.22	4.3

Table 4: Results for ML Framework assessment

Table 4 reflects the overall results of the evaluated criteria for our ML framework. As we can see, all the frameworks have a validator, and are compatible with our wrapper language, Python.

We chose TensorFlow as our language because it beats the other two frameworks in every category but speed. Another compelling factor of TensorFlow is its stable amount of popularity. All our research indicates that TensorFlow will have a substantial amount of online support and tutorials that will assist us with this project.

3.3 Model Image Processing and Standardization

Our pipeline will need to read in images taken on an iPhone and use those images to act as templates for building classifiers. Apple's Vision framework provides us with plenty of tools to tackle the problems surrounding analyzing images and passing them off to a classifier. Vision's features are integrated with the Core ML Modeling and can quickly translate images taken from a phone camera to a core ML classifier [18]. Therefore, it is imperative to investigate this framework and understand what standard formats can be created to fulfill the following goals:

- **Minimize image distortion:** The default settings on an iPhone takes images in a 4:3 aspect ratio. However, Vision, while processing images will only take square images [19]. Due to this limitation, Vision has built-in image modifiers that change a standard portrait or landscape photo into an even square. This image modifier though may distort images while modifying them, so it is necessary we determine which crop and scale method causes the least amount of distortion.
- **Orientation Data:** The metadata of an iPhone does contain some orientation data that is used to determine the rotation of the image. Additionally, the iPhone accelerometer can be accessed at any time to determine the position of the iPhone. It is necessary to examine these methods for accessing orientation data as it relates to the orientation of the dice in the image.

3.3.1 Scale and Crop Methods

Vision supports three different methods for naturally cropping iPhone images into a square image. Each of these methods cause some amount of distortion in order to focus on the classifiable objects.

Figure 3: Example Crop and Scale Options for Vision

Center Crop

Center Crop requires that the object for each model be in the center or relatively close enough to not be cropped out when the image is being down scaled. This method starts by taking the shorter side of the image and scaling that side to the standard size set by the classifier. Then it removes pixels from the edges of the image until it has the center of the image in a square [21]. This method means losing the outer edges of a standard image

Scale Fill

Scale Fill works by changing the aspect ratio of an image to match that of a square. This method will elongate or downscale one of the sides of the original image to match the square format. The stretching can cause a lot of disturbance for low light images due to pixels bleeding into one another when the image is stretched out or compressed by this method.

Scale Fit

Scale fit retains the original aspect ratio of an image but fills in the leftover square edges with black pixels. This means the relative size and shape of any object in the image will stay the same, but the image itself will be somewhat compressed due to the downscaling.

3.3.2 Orientation Data Storage

EXIF Metadata

Exchangeable Image Format (EXIF) is a file format comprised of all of the metadata for a given image on an iPhone. This data includes an orientation variable that tells the phone at what rotation it should display the image [28]. This orientation is only saved as a direction (up, down, left, and right) relative to the position of the phone when the image was taken.

Core Motion

Core Motion is the standard Apple framework for reading out accelerometer and gyroscope measurements from an iPhone. The package reads out the current roll, pitch, and yaw (as x, y, z coordinates respectively) which can then be exported and saved in a separate file alongside the image [20]. Core Motion reads out accelerometer data as a constant stream of data that gets updated at a certain interval, so the position data would need to be exported at the moment the image is taken or the interval right after.

3.3.3 Our Choice

We chose to use a center crop method for dice image models along with an output of Core Motion data to accurately determine the orientation and direction of the dice. Since dice are relatively the same size in all standard polyhedral shapes, it is easy to place them inside a

square towards the center of an image for modeling. We anticipate this will also help future classification as the core ML will likely draw square bounds around a dice when identifying it. Further we decided that it would be necessary to read in accurate accelerometer data in order to correctly determine the orientation of the phone. This would tell us the rotation and direction of the top face for any dice in an image.

3.4 Code Repository/ Version Control

When developing this product, we will need to control the versions of our project. A version control manager (VCM) will allow us to have a repository of our code. Version Control also allows us to keep track of any updates made to the project and who made them. We are going to be using this version control setup constantly throughout the project, so it is important that it fulfills a few core needs:

- **Team Familiarity**: Team familiarity is an important requirement because if the team is unfamiliar with a VCM, then it would slow down development. Unfamiliarity could also result in mistakes due to a foreign program. Team familiarity will be judged by conducting a poll where each team member rates their familiarity with the VCM between one and four. The average of these ratings will serve as the team familiarity score of the given VCM. One will represent a VCM that is completely unfamiliar to the team, while four represents a VCM that the team is fully familiar and comfortable working with.
- **Documentation**: It would be incredibly helpful for us, the client, and any potential future users, to have easy access to any documentation that we put out regarding our project. The ability to add documentation alongside codebase updates is important in order to keep the documentation current with coding changes

- **Number of Features**: The service we chose for our version control is not required to have additional features beyond the basics of being able to handle version control. However, having many available features may allow our team to integrate tools and streamline processes, improving workflow and speed.

3.4.1 GitHub

Familiarity

After conducting a team familiarity poll for GitHub, it received an average familiarity rating of three. With this rating, it is clear the team is familiar with working with GitHub and there should be little to no impact on our workflow or speed.

Features

GitHub has many useful features that are likely to be quite helpful. Some of the notable features that GitHub boasts and might be helpful for our project are [25]:

- An integrated issue tracker within the project
- Milestones and Labels within the project
- Branch comparison views
- Security uses of SSL, SSH, and https for all data transmissions
- Syntax highlighting
- API integration for a number of third-party tools for other potentially useful features.
- A High Storage Limit
- A built in Kanban Board
- Built in Wiki
- Free 100Gb storage space to all repositories
- Seamless Integration with the Git VCS, or Version Control System

Documentation

GitHub allows for any public or private repository to host a wiki page within the repository. This wiki page may contain a variety of information about the software including a project roadmap,, the current project status, and additional project documentation details.

3.4.2 BitBucket

Familiarity

After conducting a team familiarity poll on BitBucket, it received an average team score of one, indicating that the team has very little current BitBucket knowledgeHaving each member on the team learn a new VCM will take extra time and this learning curve may introduce errors due to the unfamiliar environment.

Features

BitBucket has quite a few useful features that are likely to be quite helpful. Some of the notable features that BitBucket boasts are [25]:

- Easy integration with Trello, Jira, and other Atlassian products
- Pull requests
- Code reviews
- Branch comparison
- Commit history view
- Built-in Continuous Delivery, Issue Tracking, and Wiki
- Supports Mercurial VCS along with Git
- Supports Git Large File Storage
- Functions with both Git VCS and Mercurial VCS

Documentation

BitBucket allows for any repository hosted through it to have a wiki, by simply enabling this feature. This wiki can be used similarly to GitHub's to store project documentation, the project roadmap and project status. BitBucket does not have a built-in Kanban board which is a feature to be utilized by our team as our project management tool. However a Trello board, a similar project management tool, can be easily be linked and available within the website's dashboard as an embedded board.

3.4.3 Our Choice

VCM	Team Familiarity	Features	Documentation
GitHub	3.0	11	Yes
BitBucket	1.0	9	Yes

Table 5: VCM objective metric analysis

After comparing both GitHub and BitBucket, we have decided to move forward using GitHub as our repository. Table 5 shows that both GitHub and BitBucket have solid options for documenting our project, and both also have an impressive suite of features. We ultimately decided that the familiarity that we had with GitHub outweighed the ease of use that BitBucket provided.

4. Tech Integration

Although we have determined the technologies we will utilize to tackle the issue at hand, we must also have a plan for integrating these technologies all into one project. Below is a diagram of our choices implemented into one package.

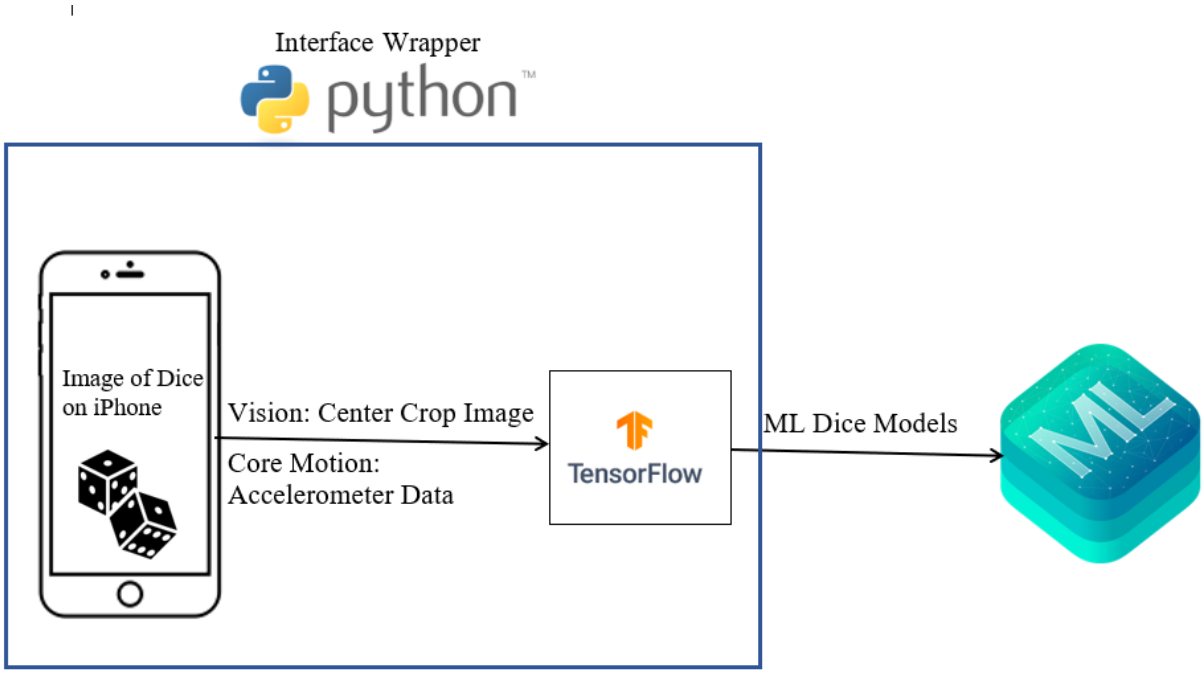

Figure 4: overall layout of project

5. Conclusion

Bridging the gap between online and in person TTRPGs allows for stories to be told more efficiently for many players of these games. Knowing what your peers have rolled in a game would remove a lot of stress and keep players honest. Also players would be able to play the game with their own dice giving them a sense of agency and connection to the game they are playing.

This document covers the technological difficulties and our team's choices to overcome those challenges. Ultimately, we believe that we have made strong choices in the technology we are using to tackle this issue. In the following table we summarized our issues, solutions, and confidence in oursolution.

Problem	Solution	Confidence (1-5)
Pipeline Wrapper Language	Python	5
Machine Learning Language	TensorFlow	5
Model Image Processing	Vision Center Crop with Core Motion Data	4
Repository/ VCS	GitHub	5

Table 6: Technological feasibility assessment conclusion

We believe we have a strong approach towards solving the issue at hand. We will continue to look for creative ways to solve any other problems as they appear. We believe that our pipeline will be an instrumental tool for creating dice classifiers. This pipeline would be one step towards the goal of keeping the focus of TTRPGs on the story and the players.

6. References

1. https://discuss.mxnet.io/t/run-time-is-different-between-python-and-c/4052/2
2. https://wiki.python.org/moin/
3. https://mxnet.apache.org/api 11
4. https://en.wikipedia.org/wiki/C%2B%2B
5. https://mxnet.apache.org/api/cpp
6. https://mxnet.apache.org/api/java
7. https://medium.com/apache-mxnet/introducing-java-apis-for-deep-learning-inference-with-apache-mxnet-8406a698fa5a
8. https://www.tensorflow.org/api_docs/java/reference/org/tensorflow/package-summary
9. https://en.wikipedia.org/wiki/Java_(programming_language)
10. https://planspace.org/20170320-tensorflow_apis_for_various_languages/
11. https://pytorch.org/
12. https://www.tensorflow.org/about/
13. https://towardsdatascience.com/deep-learning-framework-power-scores-2018-23607ddf297a
14. https://www.analyticsvidhya.com/blog/2019/03/deep-learning-frameworks-comparison/
15. https://medium.com/@mouryarishik/tensorflow-2-0-vs-mxnet-41edd3b7574f
16. https://medium.com/syncedreview/tensorflow-pytorch-or-mxnet-a-comprehensive-evaluation-on-nlp-cv-tasks-with-titan-rtx-cdf816fc3935
17. https://medium.com/apache-mxnet/mxboard-mxnet-data-visualization-2eed6ae31d2c
18. https://developer.apple.com/documentation/vision
19. https://developer.apple.com/videos/play/wwdc2018/717/

20. https://developer.apple.com/documentation/coremotion
21. https://www.raywenderlich.com/1320561-machine-learning-in-ios/lessons/3
22. https://www.tensorflow.org/versions
23. https://medium.com/@alexiscreuzot/building-a-neural-style-transfer-app-on-ios-with-pytorch-and-coreml-76e00cd14b28
24. https://www.guru99.com/what-is-tensorflow.html
25. https://www.upguard.com/articles/bitbucket-vs-github
26. https://www.elegantthemes.com/blog/wordpress/github-vs-bitbucket
27. https://www.thebalancecareers.com/what-is-github-and-why-should-i-use-it-2071946
28. https://www.howtogeek.com/354802/how-to-view-exif-metadata-for-photos-on-an-iphone-or-ipad/
29. https://developer.apple.com/documentation/uikit/uiimage/orientation