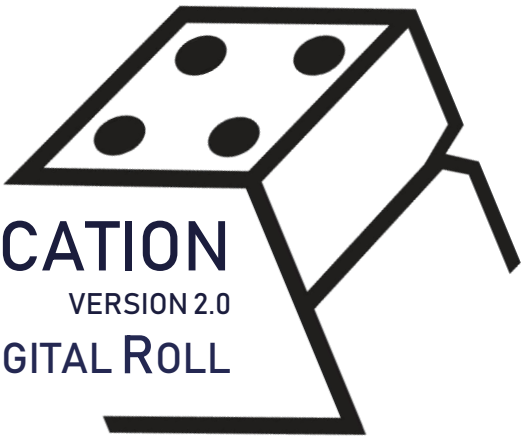
DECEMBER 13, 2019

Sponsor: JASON ROBINSON OF BEAUTYMARK DESIGN STUDIO

Mentor: ISAAC SHAFFER



REQUIREMENTS SPECIFICATION

VERSION 2.0

DIGITAL ROLL

Tyler Boice Caleb Johnson Tyler Malmon Brandon Measley

Accepted as baseline requirements for the project:

For the Client: _____ For the Team: _____

Table of Contents

1. Introduction

Tabletop Role Playing Games (TTRPGs) started in 1972 with a game called Dungeons and Dragons. What started as a small community of people and one game, has blossomed into millions of players worldwide, with many different games and versions. Today, there are an estimated 5.5 million people who play TTRPGs in the United States alone. The reason that so many people play these games is because it's a fun and creative outlet that can be experienced with friends. These games have become so popular that they have spawned many web applications (e.g. Roll20, which has over 4 million registered users). Since TTPGs can be played online, players don't have to be in the same location anymore. Technology has allowed TTRPGs to expand how it can be played and who can play together.

When people play TTRPGs they role play characters within a fictional world. When these characters need to accomplish a task or use an ability, the probability of their success is determined by a dice roll. Since the amount a character can do is limitless, these games need to use more types of dice then a standard size sided; therefore, these games use polyhedral dice such as a four, six, eight, ten, and twenty-sided dice. Currently there are two ways to roll dice as an online player. A player can roll their own physical set of dice and convey the result to the group (e.g. message, video, picture, verbally), or they can use an online dice generator that simulates a dice roll.

Our client, Jason Robinson, is one of the millions of people who play TTRPGs. He has always been captivated by how and why we tell stories which drew him to the medium of TTRPG's. Mr. Robinson is a web UI/UX designer and founder of the company Beautymark Design Studio. Beautymark is a small a small company of less than 10 employees, who focus on creating expressive interfaces that convey meaning to the user and lead them through a webpage.

Mr. Robinson host his own TTRPG games on his spare time, where some of the players are online. While hosting these games Mr. Robinson found a conflict with the methods of rolling dice that online players must use.

If a player wants to roll their own physical dice and convey that to the group, the group may not believe that was the real result. Players can easily cheat: taking a picture doesn't prove it wasn't placed there instead of rolled outcome, a video doesn't prove the player didn't roll multiple times before till they got a good result, stating your result vocally or through message, doesn't mean it was the result that was rolled. Live feed could be a solution, but live connections can sometimes have poor quality or be unreliable, making it hard for others to tell what was rolled. The other method of rolling, online dice generator, eliminates the need for physical dice.

At first glance this is a perfect solution because it almost eliminates cheating. However, by eliminating the dice from the game, it also removes a fundamental part of the game that has been there since its origin. Rolling a dice set feels like you have control of your character rather than a computer algorithm. People have their own set of unique dice and these dice may have more than monetary value. In addition, people can be superstitious, like having lucky dice or a strategy to roll their dice. Removing the dice from a player eliminates an aspect of the game that may result in the game being less enjoyable. There needs to be a way for online players:

- To roll their own physical set of dice
- To have that result sent to the rest of the group
- To have their result be trusted by the rest of the group

2. Problem Statement

Mr. Robinson wants to broaden Beautymark Design Studio's focus to expand into TTRPG's. His goal is to create an application that will solve the dilemma online placers face

when rolling dice. His vision is to allow online players to roll their own physical set of dice and have the application recognize their roll so it can send the result to the rest of the group. This will allow online players to use their own set of dice, while keeping them honest by having an unbiased program send the results to the team instead of the player.

Currently the application he has envisioned has not been created. Mr. Robinson requires a key component: a feature that uses machine learning to detect polyhedral dice and their result. Mr. Robinson needs this key feature to be in the application before he begins development but does not possess the knowledge or resources to create it. There are currently programs that detect regular pip dice, but since TTRPG's use dice with numeric values and various polyhedral shapes, these programs will not work for his solution. Therefore, this key feature must be created and implemented, then Mr. Robinson will begin the creation of the application and add additional features.

His envisioned solution for the dice detection feature is to have a phone propped up against a household object, so that the phone is angled to look at a small flat surface. He does not want a top-down view because he wants the setup to be quick, painless, and require no additional components rather than a smartphone and the household object to prop it up. The user can then roll their dice in view of the camera so the app can detect when the dice has stopped rolling and output the result. The result should be the number of dice, type of dice, and the numeric value of the top face of each die.

3. Solution Vision

The solution we have envisioned for our project is a tool to help Mr. Robinson in creating his own machine learning kernels. Originally, the project had requested that we create our own ML kernels for recognizing dice rolls, as well as build the app that the ML would fit into. This

means we would build the player GUI, and the ability to connect to multiple players; however, this was deemed to be an ill fit for a capstone project. The main issue with the original project was that the complexity and research required for the creation of the machine learning models was not a software-based problem and would take up a sizable portion of the project time. The alternative -was to turn the creation of machine learning models into something accomplishable through software. Therefore, our project was altered to a new version. Now our project will be to create a pipeline so that Mr. Robinson and the people at Beautymark Design can create their own ML kernels and have a consistent way to collect and use the data. This pipeline will make the creation and research process of machine learning easier by using a workbench and API. The pipeline will include code for an application program interface (API) that takes in images and their metadata from a phone camera to create a unified piece of data. This data will then be used in a workbench designed to create and train Core ML kernels or in an app developed by the user running a model created in the workbench. The workbench will also validate the kernels as more data is processed through the pipeline. This project can be described in two parts:

- API
    - Takes in an image from an iPhone
    - Produces data in a consistent format
    - Must be fast, feel almost instant
- Workbench
    - Takes in data from the API
    - Outputs an Apple CoreML model and results of validation

Figure 1 shows a sample data collection app that displays the accelerometer data. The blue, green, and yellow boxes are the x, y, z coordinates of the accelerometer data, and the purple

box at the bottom is the upload button. The upload button will trigger the conversion to the JSON format using the API, before sending that data to a database for use in the workbench later.

Figure 1:data collection app

The API will take the metadata input for each image which may include: the phone orientation, the current angle of the phone, the type of camera lens, and the amount of light taken in. Mr. Robinson has requested that this metadata be displayed to the screen so the user can see their current conditions and adjust it for optimal accuracy. This API is vitally important because it standardizes the input information that will be used in training and validating the kernels. The workbench portion of the project takes in the API data and images and uses them to train and validate ML kernels. The more images and data supplied to the workbench, the more accurate

the kernels. This will create accurate classifiers for the images. In the case of Robinson's program, the classifiers will be the type of polyhedral dice and their value.

The product created will be executed by Mr. Robinson and the personnel at BeautyMark Design Studio; therefore, the product needs to be easy to learn. Luckily, the BeautyMark personnel have computer experience, so the pipeline can be run from the terminal in a machine with macOS or Windows 10. The images will be provided by Mr. Robinson; however, our team will provide an example set to show how to input these images into the pipeline. We will test the pipeline on this example set to ensure the product works to the satisfaction of Mr. Robinson.

4. Problem Requirements

In order to create our product for Mr. Robinson it is crucial that we first break down and compartmentalize how we will approach the design of the product. Through our discussion as a team and meetings with our client, we have fleshed out a set of functional, environmental, and performance-based requirements that we must solve while developing our product.

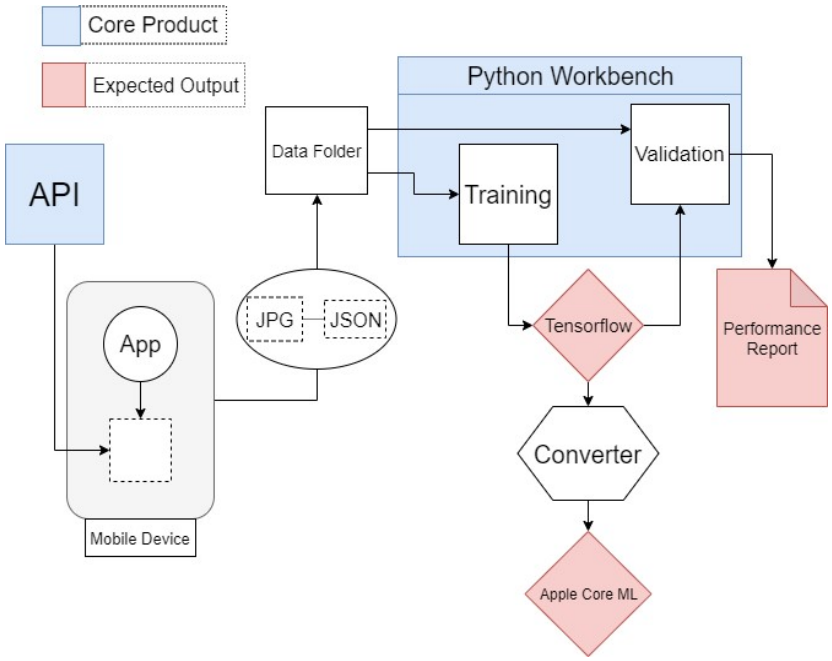


Figure 2: Workflow of the Pipeline

4.1 Functional Requirements

The pipeline we create for this project is primarily split into two parts: An API used to standardize the image format and a workbench used to create classifiers as seen in Figure 2. Both are necessary tools for ensuring we have consistent data that leads to the creation of accurate classifiers.

4.1.1 API Functional Requirements

The goal of the API is to handle the process of handing off images to eventually become classifiers. To achieve this goal, we will be converting an image to a JSON file. JSON (JavaScript Object Notation) is a file format that stores information as text. The files remain lightweight even when used to merge information from various sources. For this reason, our API will use JSON to merge an image encoded as text with accelerometer information from a phone in one file. Saving accelerometer data with the image couples them together and ensures every image has a related accelerometer position. This will allow us to better determine which way objects are facing in the image. By standardizing the output of the information from the phone we will also be able to always know what kind of data is being placed into our workbench.

4.1.2 Workbench functional Requirements

The workbench handles generating TensorFlow based classifiers from a given set of data that has been standardized via the API. The workbench will run from a command line and will be pointed towards the directory where images are placed. Once executed, the program will go through the images and begin creating a classifier for the dataset. Furthermore, the classification process should report a log that keeps track of how long the process took, how many images it processed, and any possible errors the program incurred.

4.1.3 Validation and Reporting

Validation and reporting will be handled through two forms of feedback, command line outputs and text log files. As the models are tested, the results will be shown to the user in the terminal as they are generated while simultaneously being printed to a log file files for further analysis.

4.1.4 Training

It is crucially important that this pipeline is usable by Mr. Robinson and any of his employees. He has clearly stated that after the product is complete, no matter how functional it may be, it is worthless to him if he does not know how to use it. However, this does not mean we have to create a graphical user interface (GUI) to use the pipeline. We have spoken to Mr. Robinson about this issue and he has stated that he has a basic understanding command-line interface (CLI) and is willing to work with a CLI. This means we will be working with Mr. Robinson closely to make sure that he understands what inputs are necessary for our CLI and how he should interpret the outputs. We will also write a well-documented tutorial on how to use our pipeline that we will leave with Mr. Robinson once our project is complete. It is crucial that Mr. Robinson's input on the CLI's ease of use be taken into consideration. If needed, we will conduct user tests with our pipeline to make sure that someone with little CLI experience can use and understand it within an hour. Although Mr. Robinson primary uses Apple products the pipeline will be executable on multiple platforms. This is especially important since TensorFlow does not support GPU boosted processing on Apple devices (without requiring an external GPU). If large datasets are being processed, we would recommend taking advantage of this feature on a device has the available hardware to support it.

4.2 Performance Requirements

It is necessary for both parts of our project to run efficiently and help facilitate the speedy creation of various classifiers. Our client has placed emphasis on being able to use our project to quickly create accurate classifiers.

4.2.1 API Image Conversion Time

The time it takes to convert both the image and snippet of accelerometer data to a json file format should feel instant and should take no less than half a second per image.

4.2.2 Building Classifiers

The workbench needs to be able to create classifiers in a reasonable amount of time for our client. The workbench will accept a range of 10 - 1000 images at a time all stored in one directory. It should take approximately 30 minutes to 8 hours for the entire process to complete. The completion time will vary based on the total number of images and the software/ hardware that the workbench is running on. For the most ideal times, a GPU is required as it greatly helps with the classification process.

4.2.3 Runtime

Mr. Robinson understands that training machine learning kernels can be a long process. The machine that Mr. Robinson runs the pipeline on will determine how long it takes the pipeline to train the kernels. Mr. Robinson would like to let this pipeline run for a standard work day (i.e. eight hours). Ideally the product can be given a certain number of images, and the user can return eight hours later to trained Core ML kernels.

4.2.4 Easy to learn tutorial

Along with our project we will include a technical tutorial that will walk new users through the process of using our tool to create classifiers. It should take a technologically literate person who is new to our system, at most an hour to setup, troubleshoot, and run our workbench. We expect a technologically literate person to be someone who knows how to navigate a file directory via command line and understands other basic command line-based features such as passing arguments to a program.

4.3 Environmental Requirements

Since this project is for our sponsor, it is important that he give specific requirements to best fit his needs. These are our environmental constraints and are non-negotiable. We must incorporate all these requirements in the final product.

4.3.1 Converting Models to Core ML

Models will need to be converted into Apple's Core ML once trained so that the models will be optimized for iOS devices. Our client wants this feature because he intends to use the models our product generates, to be used in an app for iOS devices. Thus, it is a requirement that our workbench support outputting Apple Core ML models. For the project we will be using TensorFlow which outputs its own .pb filetype models but can natively support conversion to Apple Core ML models. The product will first generate the TensorFlow models and once complete, convert the result into Apple Core ML.

5. Potential Risks

There are several risks to the viability of this project as seen in Table 1. The main risks that have been considered include those that threaten the project in a way that will prevent its' completion or hinder its' usage. Through research and data gathering the following risks have been assessed and the team is currently investigating the possibility of others.

Risk	Severity	Likelihood	Mitigation
Data and its complexity may require too much training power	High	60%	Design for multi-platform capability to allow users to transfer to more powerful setups easily
User may not be familiar with command line interface	High	15%	Create verbose documentation, walkthroughs, and/or tutorials
Apple update breaks API	Very High	1%	Observe and monitor potential changes.
Conversion to of TensorFlow to Apple Core ML breaks.	High	1%	Observe and monitor potential changes.

Table 1: Potential Risks of our project

5.1 Risk: Data Complexity

As per the requirements of the project, the type of data that will be used in training the machine learning kernels needs to incorporate additional details in order to properly guide the training of the images. These extra pieces of meta-data will be taken from the phone's sensors and paired with the image taken at the time. Our research on the usage of TensorFlow indicates that greater amounts of information result in longer wait times for the training of kernels. This is an inherent property of the machine learning process and as such, must depend on the user to supply adequate processing power in order to reduce these wait times. The likelihood that users will eventually need to move their work on to more powerful setups for training is estimated to

be most users (around 60%). Should users be unable to run the training in a reasonable time they will not invest in this product unless there is a way to avoid this problem. Since this issue could potentially eliminate possible users, including our client, this poses a high severity.

The best method for mitigating this risk is to allow for multi-platform capability. Since the system must rely on the processing power provided any user of this system, they will likely have the desire to transfer their work to a different setup. If the workbench for training the machine kernels is designed to be as multi-platform and portable as possible then this will mitigate the potential risk of losing users and investors who will require this capability.

5.2 Risk: User Accessibility

The current model for this product focuses on using a command line interface to allow the user to operate the workbench. While this provides speed and flexibility, a command line interface may prove difficult to learn. There is a low likelihood that users of this workbench will be completely unfamiliar with command line interfaces, as this product's main audience is intended for users with some degree of coding experience. This product requires some experience because the output of the workbench is only useful to users capable of coding a product around the kernel along with coding in the API packaged with this workbench. While the likelihood of this issue is low, the potential severity to the project is high. The reason for this severity is that while users may be familiar with general command-line, they will not be fully trained in the command line created for this product. Due to the potential complexity of the command line interface, users may be unable to utilize the product fully or become frustrated leading to disinterest.

The method chosen to best reduce the potential hazards of this risk is to include extensive documentation as well as clear instructions on the usage of this product. The documentation will

need to clearly and unambiguously cover each aspect of the workbench and its command line interface. This documentation should allow users to find the desired information needed to effectively use our product. The second part of this strategy is the creation of instructions for the products uses. These instructions could come in the form of simple manuals, video demonstration, and/or step by step guides. These various forms of instruction should cover common use cases of our product in a clear and concise manner, so it is easy to follow.

5.3 Risk: Updates to Apple IOS

This project features an environmental requirement which states that the product must be functional on Apple's mobile devices. The portion of this project that is used in mobile devices is the API. This API must collect data and images from the phone and store them for later export. This API must also be used in the product created by the user which uses the kernels generated by the workbench in order to maintain consistent function. Since Apple is a third party whom this project's team has no control over, it is possible that they might update their mobile device product's systems. This update could result in the API ceasing to function or may remove the viability of the API, if said update removes the ability to gather data and images. This poses a potential risk whose severity is very high. This event, should it occur late in development, would completely derail the project. Luckily this possibility has a very low chance of occurring and may instead result in minor changes to the API. The best form of prevention that this team can provide for this scenario is to closely monitor Apple and its' mobile device system updates so that any potential issues can be identified and fixed immediately.

5.4 Risk: Updates to TensorFlow and Apple Core ML

Our project uses TensorFlow as it is one of the few machine learning systems which can be converted into Apple Core ML. This creation of Apple Core ML is a requirement imposed by our client. It is possible that this requirement may end up becoming impossible to fulfill if Apple changes its conversion system so that it no longer supports the TensorFlow used in our workbench. Because this could result in a failure to meet our contract between the team and our client, this risk is a high-level threat. This risk is not very likely to occur as Apple has given no indication of making alterations to the converter and it may also be possible to continue using the old converter without any major issues. The only course of action is to ensure that our client is aware of this risk in order to avoid litigation and to monitor the situation in order to provide a quick response that will hopefully satisfy the client's needs.

6. Project Plan

The project plan for our team focuses on the 2019 fall semester timeframe. The major events that are the most relevant at this time are the tech demo presentation and our final requirements paper (which must be finalized so that our client will sign off on it). The tech demo presentation is the current focus of our team and thus has been planned out into its finer details.

The current project plan, as seen in Figure 3, is as follows. The team has already completed the milestone of fully researching the technical feasibility of this project. This was a major milestone that allowed the team to further its understanding of the project and the project's viability moving forward. Shown by the "now" line in Figure 3, at this current point in time the team has finished the API /TensorFlow classifier demonstration as well as fully update the current team website. Aside from these milestones the team is currently in the process of

finalizing the requirements acquisition in order to have the client be approved and signed by the client.

Figure 3: Current Projected plan though 2019

Since the beginning of this project the team has been working towards the major milestone of completing the requirements acquisition that will create an initial contract for our client to sign by December 13th (Figure 3). This task is currently on track to be finalized and approved by the client. The current focus of the team is to ensure this document meets the expectation of our client and to continue to prepare for the development phase next semester.

7. Conclusion

Currently online TTRPG players are given a choice on how to roll dice. They can roll their own physical set then convey the results to their group (e.g. picture, video, live feed, vocal), resulting in the potential of the group accusing them of cheating. Or they can use an online dice generator and be forced to let a computer determine the fate of their character; meanwhile their physical set they have come to love, collects dust.

Our sponsor Mr. Robinson wants to create a product that solves this problem by creating an application that allows online TTRPG players to roll their own set of dice, while also

decreasing their ability to cheat. To achieve this goal, Mr. Robinson wants to create a feature of the application that detects polyhedral dice and their result using machine learning.

Our project will help Mr. Robinson with his application by creating a workbench and API that will assist in the creation of a machine learning function that uses data collected from a mobile device. The data collected from the mobile device will use the API that we will create specifically to work alongside the workbench to unify the data and remove inconsistencies from use after training. This will allow users to easily create and validate various machine learning models to be used in their own projects such as an app that reads physical polyhedral dice. Projects such as this would allow online players to be able to play the game with their own dice, giving them a sense of control over their character when playing the game.

In this document we have shown the requirements for what our project should accomplish. As we progress with this project, we will be constantly communicating with the client as well as referring to this document, to ensure the product we create meets the client's needs. In addition, our client will review and sign this document to show his approval of the requirements for this assignment.

We believe we have a strong approach towards solving the issue at hand. We are on track to complete our product by May 2020. We have already completed our technical feasibility and created a demo that displays the potential of our project. Our demo shows that TensorFlow can identify the value of a regular pip dice, as well as detect a given polyhedral dice. In addition, our demo shows that we can take an image from an iPhone and some of its accelerometer data and display the data. We will continue to develop our project in the upcoming months to produce a final workbench and API that our sponsor desires. Throughout the course of our project we will observe the potential updates and changes to Apple Core ML, Apple IOS, and TensorFlow. We

will also conduct further research into future potential issues such as training model time frames and multi-platform development.

Our team believes that our pipeline will be an instrumental tool for creating not just dice classifiers, but for furthering the world of machine learning by reducing the complexity of entry. The implications of our project are beyond dice and even imagery. Our project will not only help our client create his application to help the TTRPG community but has the potential to help more projects assist other communities. Our project could be used in countless future projects and help usher a wave of developers to create classifiers to solve future problems and improve the world through machine learning. We are on track with our project and ready to take on the challenges ahead.