Eli Wolfson & Ivy Zhang
Software Design (Python)
02/26/2020

Project Proposal

Section 1. The Big Idea

Our project is a blackjack game that will track the users' and casino's statistics and play over time. The game will also have built-in functionalities to show the best way to play every hand given any situation that the player encounters while playing. While we understand that there are other languages better suited for developing websites and games, we empathize heavily on building deep and useful analytics functions in our game to help users, especially beginning users, advance. Therefore we want to use python given its excellent data analytics capabilities.

This project will explore many different topics including:

- generating a database of users and tracking their statistics over time (we anticipate that this will be done using a login built on Django);
- storing and analyzing user statistics (such as win rates, session P&Ls, life-time winnings/losses);
- creating a functional website with an integrated backend database and a visualization of trends and data on play, highest dollar total achieved in session, etc.

The MVP will include a login page, the blackjack game with suggested play, and a dashboard to track play over time and other statistics.

Our stretch goal is to build in advanced statistics such as showing the probability of winning in any given situation with any given hand and as the hand progresses. We would also like to include a dashboard that shows the casino's win rate and bankroll across all users and all users winnings (P/L).

Section 2. Learning Goals

We both would like to hone our web development skills while building a fun-to-use product. Through this project, we hope to build data analytics skills that are transferable to different situations while adding value to players of a real-world casino game. Due to the fact that we are not professional software developers, the ability to leverage pre-existing frameworks and manage a database of users is a crucial learning objective for us. Ideally, the end product will be of a high enough quality that we can show it to potential employers as a tangible example of our coding skills.

Individual Goals:

Ivy would like to work on building an interactive software that tracks, stores, analyzes data streams on a continuous basis. She would also like to improve on researching and editing existing packages/libraries to achieve project objectives in a timely manner.

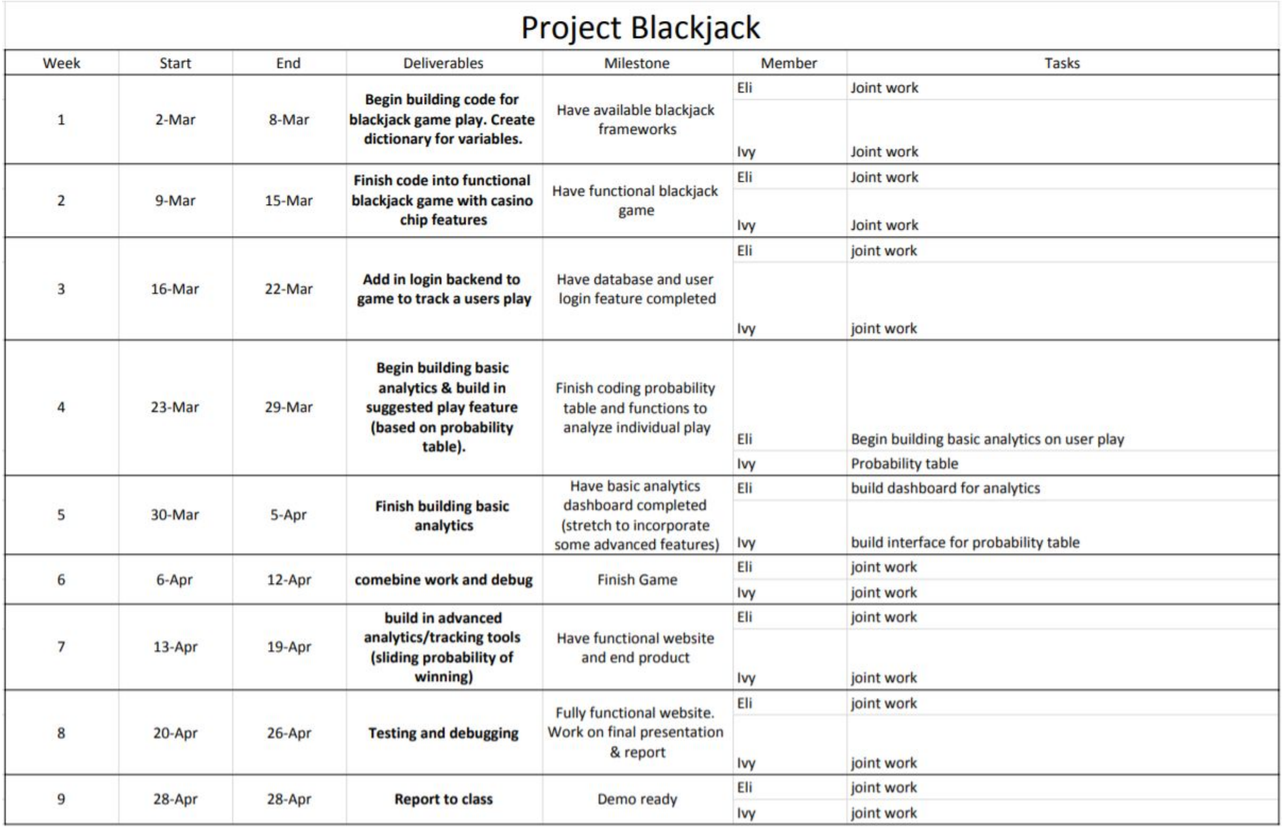
Eli would like to work on his python and analytics skills while learning how to launch a functional website and product. He would like to emphasize learning what it is like to work on a web development project on a team.

Section 3. Implementation Plan

Given that the project places emphasis on data analytics in blackjack, we will build all data tracking and analytics features from scratch. Also due to the nature of blackjack being a game, we will be building a website with presentable user interfaces. To do so, we anticipate incorporating other languages such as HTML for the UI and UX front end during the process.

On the other hand, we will utilize as much existing code as possible when building the login function and the blackjack game itself. For the backend database and login functionality, we plan to consult external advisors and use the Django framework. In terms of the game itself, we have found on github several existing versions of the blackjack game in python. We plan on leveraging these resources yet the main challenge is to modify the code and incorporate all desired functionalities which may imply heavy revisions on existing codes.

Section 4. Project Schedule



Project Blackjack

Week	Start	End	Deliverables	Milestone	Member	Tasks
1	2-Mar	8-Mar	**Begin building code for blackjack game play. Create dictionary for variables.**	Have available blackjack frameworks	Eli	Joint work
					Ivy	Joint work
2	9-Mar	15-Mar	**Finish code into functional blackjack game with casino chip features**	Have functional blackjack game	Eli	Joint work
					Ivy	Joint work
3	16-Mar	22-Mar	**Add in login backend to game to track a users play**	Have database and user login feature completed	Eli	joint work
					Ivy	joint work
4	23-Mar	29-Mar	**Begin building basic analytics & build in suggested play feature (based on probability table).**	Finish coding probability table and functions to analyze individual play	Eli	Begin building basic analytics on user play
					Ivy	Probability table
5	30-Mar	5-Apr	**Finish building basic analytics**	Have basic analytics dashboard completed (stretch to incorporate some advanced features)	Eli	build dashboard for analytics
					Ivy	build interface for probability table
6	6-Apr	12-Apr	**comebine work and debug**	Finish Game	Eli	joint work
					Ivy	joint work
7	13-Apr	19-Apr	**build in advanced analytics/tracking tools (sliding probability of winning)**	Have functional website and end product	Eli	joint work
					Ivy	joint work
8	20-Apr	26-Apr	**Testing and debugging**	Fully functional website. Work on final presentation & report	Eli	joint work
					Ivy	joint work
9	28-Apr	28-Apr	**Report to class**	Demo ready	Eli	joint work
					Ivy	joint work

* Task distributions will be revised as date approaches.

Section 5. Collaboration Plan

Overview

We plan on holding each other accountable to complete work in a timely and accurate fashion. We will collaborate on most elements of the project, especially at the beginning, as our first critical objective is to build a functional MVP. Tackling this portion together is very essential before accomplishing other goals. We then plan on adding advanced analytical features and capabilities by splitting up works individually, with each feature being divided up into separate modules.

Communication

We recognize the importance of maintaining unimpeded conversation. We plan on communicating over Slack and through weekly in-person meetings (Sunday 6pm) where we will review code, go over our weekly progress, and plan for any deliverables in the coming week. This plan will roughly follow the agile development framework with weekly goals (sprints). Aside from finishing assigned work before weekly in-person meetings, we also plan on attending office hours together as needed when we encounter unresolvable issues.

Documentation

All documents related to the project should be stored in shared google drive or shared repository on github. Shared google drive is used for any class deliverables, writeups, presentations, and other additional documentation. A shared repository will also be created upon the start of the project with dictionaries, modules for the game, the analytical features, and etc.

Section 6. Risks

While we are excited to achieve our project goal, we do recognize that it is an ambitious and complicated project that we are taking on as novice web developers. As such if we do not hold ourselves accountable to stick to the schedule, we will likely be unable to accomplish our lofty goals. Because there are several components to this project, the biggest risk is that we will not be able to have a functional blackjack game by our designated deadline. Having a functional game in a timely manner is critical because this component is the stepping stone for all future components. To mitigate this risk, we have started searching for applicable packages and frameworks, and we've agreed to give up certain advanced functions and interface designs (just for the game itself) if we find ourselves behind schedule.

It is also a risk that we have to incorporate several languages and a framework(Django) that we are not familiar with. We will be frontloading all other tasks to make sure that the majority of our work is finished before moving onto building the webpage and using Django. We will also make sure to communicate with professors frequently to avoid any undetected or last-minute problems.

Section 7. Additional Course Content

Because our project requires us to write new python codes, integrate pre-written codes, develop website interface, develop user logins and database, and so on, we would love to see (if possible) the following contents covered in class.

- Building Logins and database management
- Launching a website with a usable UI front end.
- Methodologies for integrating existing code to work with your project seamlessly.
- Tips for more effectively debugging code.