***SOFTWARE PROJECT PLAN***

1. **Introduction**

We will be creating a web-based black jack game where users can play against the house.

**1.1 Project scope**

The project aims to replicate an interactive blackjack game where users can play against the dealer. The game will follow all the rules of blackjack, even incorporating several decks into a shoe. The interface will be user-friendly, providing buttons and inputs that allow different bets and calls to be made.

**1.2 Major software functions**

The software will allow users to open a blackjack simulation. Functions of the software will replicate the key requirements of a traditional blackjack game, such as making bets, hits/stays, splits, double downs, and choosing the size of the shoe according to the number of decks.

**1.3 Performance/Behavior issues**

Performance: Response to user input during the game (bets, calls, etc.) should replicate a real life in that there is a short delay for the dealer to respond to a certain response.

Behavior: We expect the software to be reliable in that it won’t crash randomly and will run efficiently. We also wish to build the software in such a way that it is scalable so that additional features, if possible, can be added.

**1.4 Management and technical constraints**

. Time: Naturally, the project has a deadline that lines up with the class’s expectations for completion.

. Team Availability: As students, we have many things in academia and personal life going on while working on our project. This needs to be accounted for to ensure project completion.

**2.0 Risk Management**

This section discusses project risks and the approach to managing them.

**2.1 Project Risks**

. As our software is web based, there is the potential issue that our project won’t work across multiple browsers.

. Lag or delays in rendering cards and processing user actions could degrade users’ overall experience

**2.2 Risk Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Risk Name | Probability (Low/Med/High) | Impact (Low/Med/High) | RM3 Pointer |
| Browser compatibility issues | Med | High | Related to UI testing and cross-browser support |
| Performance Issues | Medium | High | Linked to optimization techniques in rendering |
| Project Delays | High | Medium | Related to time management and milestone tracking |
| Scope Creep | Medium | High | Connected to project scope management |

**3.0 Project Schedule**

This section presents an overview of project tasks and the output of a project scheduling tool.

**3.1 Project task set**

Phase 1: Planning

. Define project scope, objectives, and constraints

. Identify core functionalities

. Risk assessment

. Development project timeline and milestones

Phase 2: Modeling

. Design architecture

. Create UI wireframes and layout

. Define data structures

. Plan state management

Phase 3: Construction

1. Core game mechanics of Blackjack

2. User interface and interactivity

3. Game logic and refinement

Phase 4: testing and debugging

. Test functionality

. Works across different browsers

. Fix identified problems

Phase 5: Deployment and Feedback

. Test the game out on a web server and gather feedback

**3.2 Functional decomposition**

For any blackjack game, a deck of cards is required. The first step would be creating a standard 52 deck of cards, being able to scale off of it to later incorporate multiple decks into a single shoe. Then at the start of blackjack, cards are dealt with by players and dealers. From here we will need to work out a shuffling system for the cards and a dealing system that changes throughout the game (and that resets after each shoe). From there, we will have to make sure that each card is assigned a value that affects the player’s overall score as they try to get close to 21, such as how all face cards are worth 10 and that aces are worth both 1 and 11. Then calls such as betting, staying, etc. will be implemented. Other rules of the game will then be implemented to simulate a casino game, such as how the dealer stays on all 17’s. From there, testing and debugging will continue to ensure that the game works as intended.

**3.3 Task network**

A graph on a white surface

AI-generated content may be incorrect. **3.4 Timeline chart**

A diagram of a software development process

AI-generated content may be incorrect. **3.5 Schedule compliance**

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Planned Completion | Actual Completion | Notes |
| Deck and Card system | Week 1 | TBD | Includes card creation and value assignment |
| Shuffling and dealing | Week 2 | TBD | Ensures fair distributiuon of cards |
| Game logic and rules | Week 3 | TBD | Implements game flow |
| Betting system | Week 4 | TBD | Adds betting functionality and balance tracking |
| User interface | Week 4 | TBD | Develops front-end visuals and interactions |
| Testing and debugging | Week 5 | TBD | Identifies and fixes issues before deployment |

**4.0 Appendix**

**The Venetian Resort.** “How to Play Blackjack.” *The Venetian Resort Las Vegas*, [www.venetianlasvegas.com/resort/casino/table-games/how-to-play-blackjack.html](https://www.venetianlasvegas.com/resort/casino/table-games/how-to-play-blackjack.html). Accessed 22 Feb. 2025.