**CPSC 224 Final Project**

**PROJECT PLAN**

**04/01/2024**

**Plinko**

**Good Noodles**

**A person wearing a suit and tie

Description automatically generated**

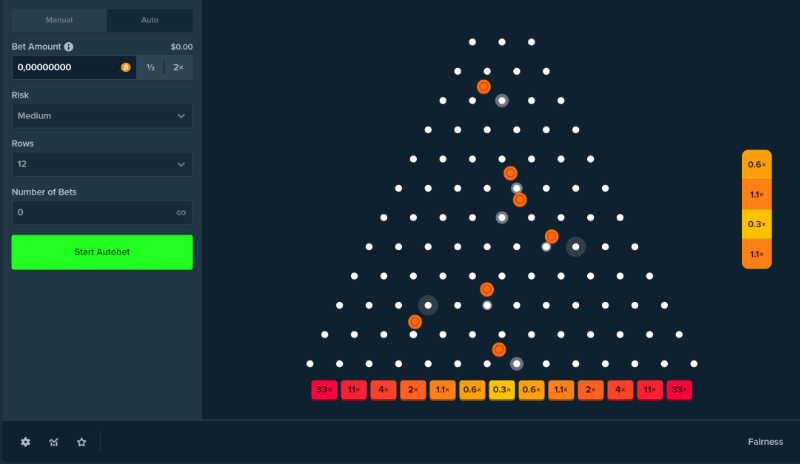
**Prepared by:**

Jack Kabil, Scott Riddle and Mark Sanghera

**1 Project Overview**

**1.1 Project Summary**

For our final project we are making Plinko. Plinko is gravity-based game where a player drops a ball into a system of pegs. The ball bounces off the pegs until it reaches the bottom where it lands in a zone that dictates how many points that zone is worth. In the case of our game each ball will have a monetary value and the zone below will return a multiplier against that value (i.e. zones are x0.5, x0.75, x1.25). The game will be primarily single player but support multiple plays (multiple dropped balls) at the same time. Our game will look similar to the image below.



**2 Project Requirements**

**2.1 Major Features**

Provide a description of the major features that must be implemented for a viable and useful product. Major features include broad feature areas, constraints that must be met, and other major items that must be completed for the project to be considered successful. You should have at least 4-5 major features.

**Table 1: Major Features**

|  |  |
| --- | --- |
| *Feature* | *Description* |
| *Physics Engine* | This is essential for the game to work in the intended manner. |
| *Assigning monetary value to dropped ball* | The dropped ball must have a monetary value assigned to it so that it can be multiplied by the zone it lands in. |
| *Player bank amount* | The player will need to have a banked amount of money. The bets placed by the user will come from this bank, and any money they win will be added to this banked amount. |
| *Slot rewards* | When the ball lands in a slot, a multiplier will apply to the monetary value of the ball that was dropped.  There will also be a jackpot slot that will appear periodically which will allow users to win the highest amount of money. |
| Extra Features | Statistics page (biggest winners, losers), watch an ad for more money, login with name and tracking player. |

**3 Project Game Design**

**3.1 Initial User Interface Design**

Provide a description of the general user interface layout, including a set of initial user interface design mock-ups. This can be done as a sketch if it's cleanly done by hand, or digitally using a drawing tool.

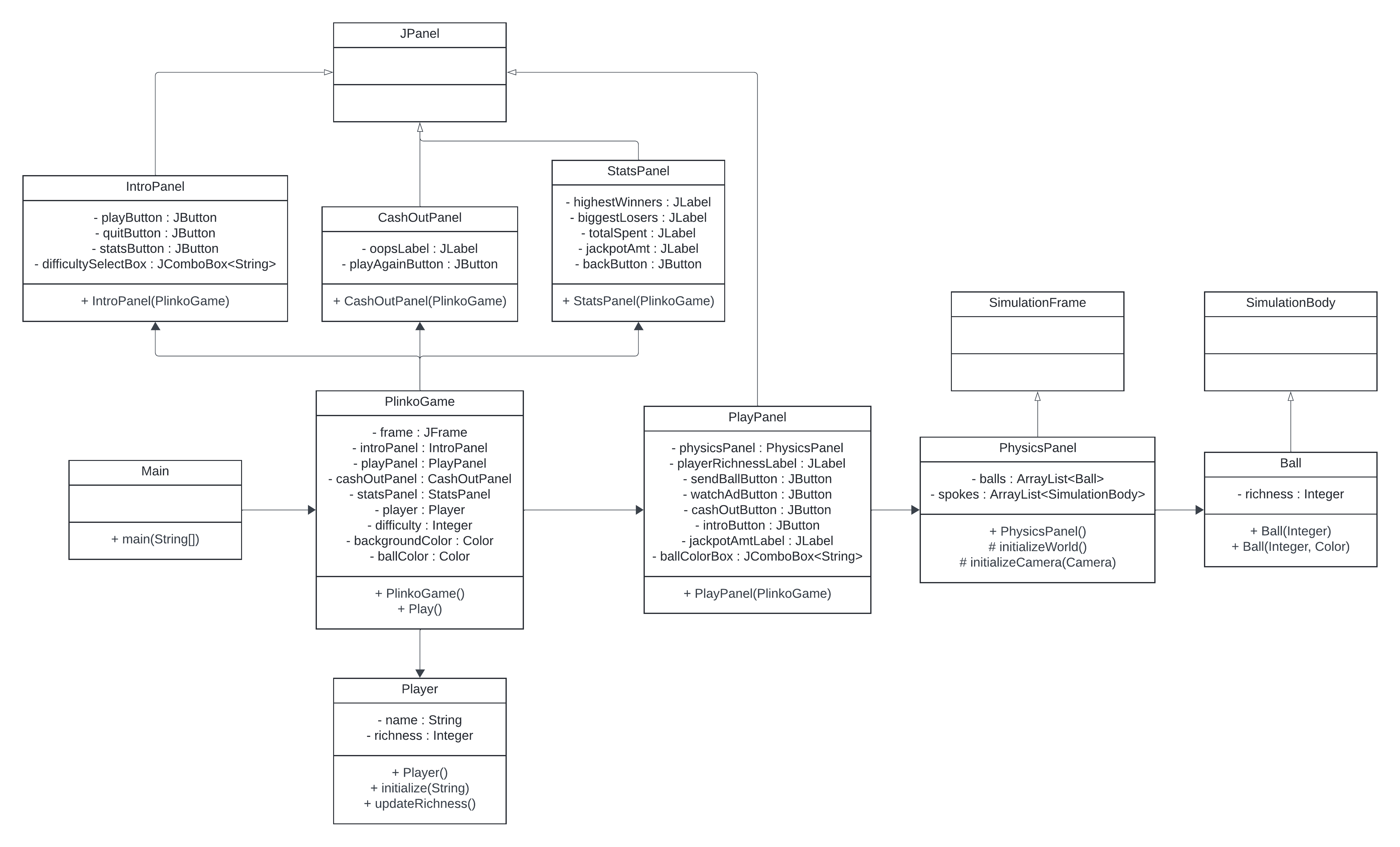
A white paper with writing on it

Description automatically generated

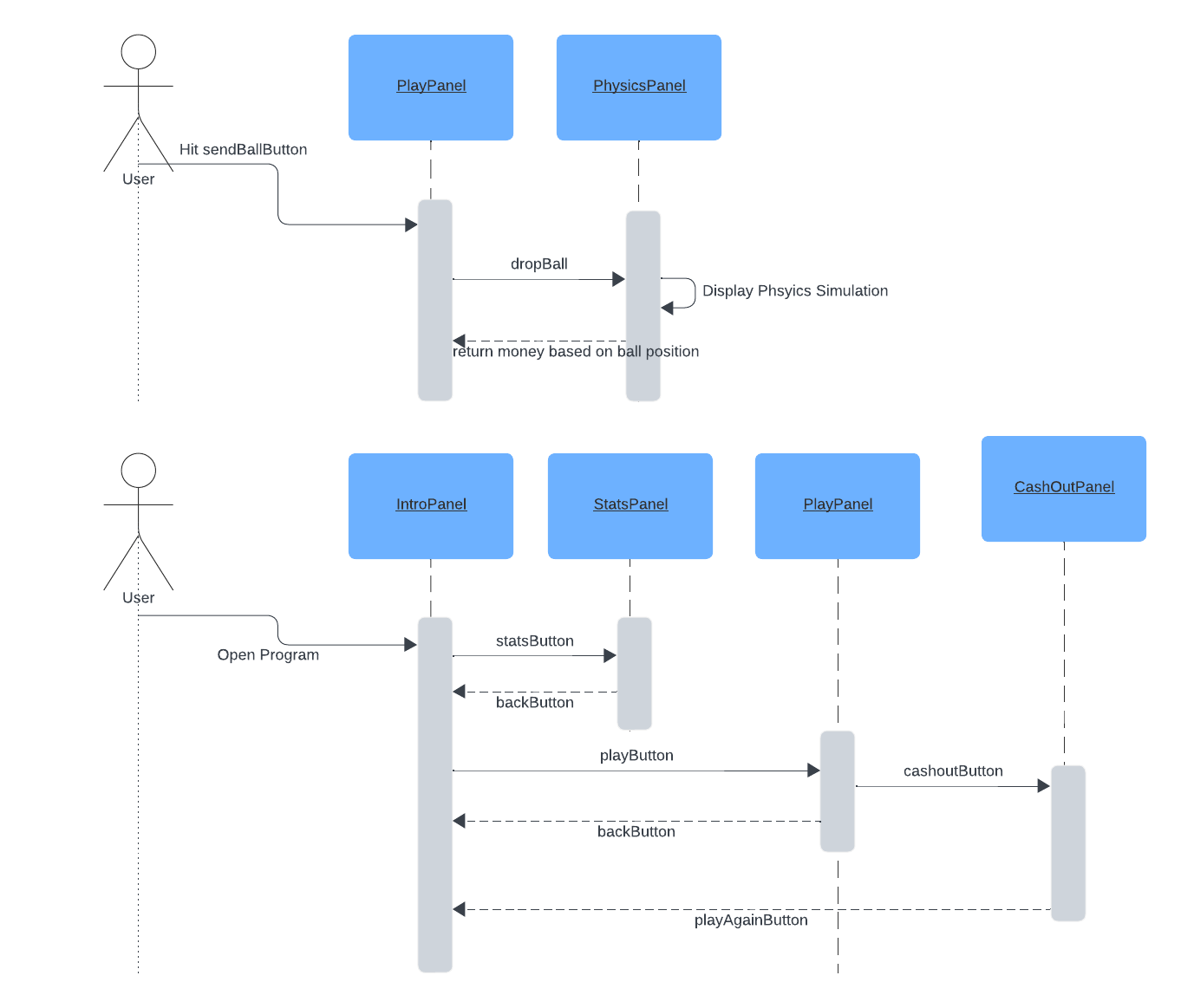
**3.2 Initial Software Architecture**

Provide a description of the initial architecture of your application, focusing on the major components of your system and how they will interact. This should include a UML class diagram and 2-3 sequence diagrams of the various modules in the system interacting.

UML Class Diagram



UML Sequence Diagrams



**4 Project Schedule**

Provide a description of the major scheduling dates of your project. For each schedule milestone dates, clearly describe the milestone (e.g., what features will be implemented) and when the milestone must occur by. Include the project plan, code complete, presentation, and final report dates.

**Table 3: Major Scheduling Milestones**

|  |  |  |
| --- | --- | --- |
| *Milestone* | *Description* | *Target Completion Date* |
| Complete project plan | This document; it describes the summary, requirements, design, and schedule of the project. Sections such as the major features, GUI design, and the UML diagrams will be useful to refer back to later in the process. | First week of April |
| Display physics simulation in JPanel | As of now, we have created a rough physics simulation for our project that opens in a JFrame. We will need to be able to open this simulation in a JPanel in order to insert it in a container in the GUI. This is a major step in our game’s development process and is necessary for our game to utilize a physics simulation. | Second week of April |
| Finalize and build non-functional GUI | This is the final GUI complete with all pages, containers, and elements needed for the game to fully function. This gives us a completed blueprint to work off of when implementing backend support. | Third week of April |
| Add backend support for GUI | This adds functionality to the previously unfunctional GUI. At this point, a user will be able to play and interact with the game, but bugs may exist in this release. | Fourth week of April |
| Complete code | Here, all bugs introduced to the backend in the previous release will be fixed. This release will be our fully functional, bug-free version of the game that is ready to ship. | Fourth week of April |
| Presentation | After copious preparation and rehearsal, we will present our project to the class in the week before finals. | First week of May |
| Final report | Due at the time of the final, we will write a final report about our project as a whole. | Second week of May |

**Appendix**

Provide additional supplemental information in an appendix as necessary.