Project 5 PDF

Project Summary:

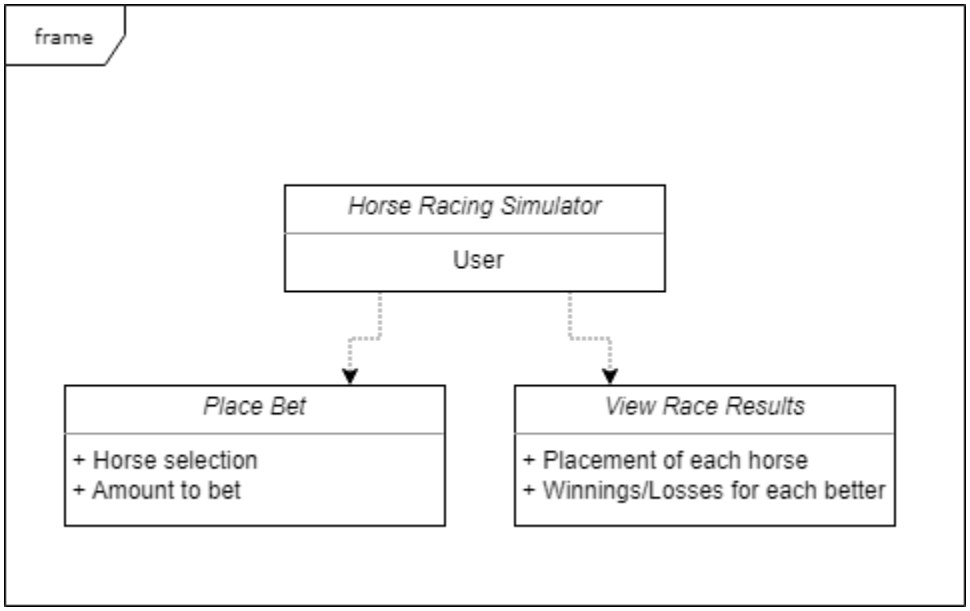
- Project Title: Horse Racing Simulator
- Team Members: Tommy Hoang, Christopher Parker, Brehn Heil
- High-level overview: Our team is going to be implementing the FNCD racing event, but to a more polished and extreme level. We will take inspiration from the FNCD racing event and convert that idea into a horse racing simulation that will consist of horses of randomly generated traits and skills. Each horse generated will also keep track of its Win/Loss ratio as well as other information if needed. During the race, we will determine the odds for victory for each horse and allow betters to make bets on the horses present in the race based on the odds. Winnings will be determined by the position of the horse at the end of the race with first place giving the most amount of money. Our system should generate a single or series or horse races and print out the horse placements as well as the betters and what horse they bet on, and how much they earned/lost from the race. The user will also be playing as a better and will have an initial balance of $500. The user may choose to not bet as well. If the user wins the bet, the user earns money based on the position of the horse, else the user loses the bet.

Project Requirements:

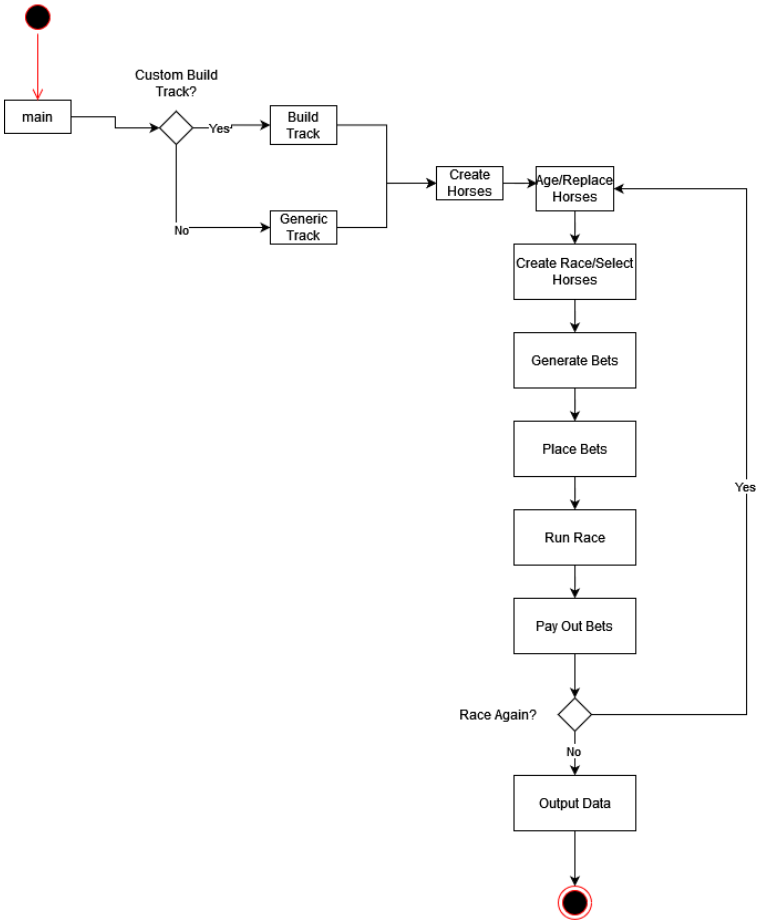
- We must require having a horse class with randomized stats. We must also require a way of recording the results similar to the closing event for the FNCD project. We also must allow the user to put inputs into the simulation to choose whether to bet or skip and how much to bet.

Users and Tasks, Use Cases:

There are two types of users which are the betters generated randomly and the user itself. The user will be able to place bets (or not) and view the race results.



UML Activity or State Diagram:



Architecture Diagram:

Data Storage: Our data will be simply maintained in memory, but we will print out the results and information on a text file after a successful horse race. The tool Tommy will be using the Eclipse IDE for Java 17.0.5 where he will maintain his memory.

UI Mockups/Sketch:

UML Class Diagrams & Pattern Use:

User Interactions/UML Sequence Diagrams:

+ The user will have a stored balance with the system and the user can interact with it by betting.
+ The user will be able to interact with the application to choose whether or not to bet on the horse race at the moment.
+ The user will be able to select how much they're willing to bet on the horse race with limitations based on their current balance.