

### Abstract:

This paper describes the design methodology, implementation, challenges faced, and current status of the game 'Snakes and Ladders' developed by the author for the course CS510

### Objective:

'Snakes and Ladders' is a well known board game played with multiple players. Players compete with each other to reach the *end* mark while the board is replete with snakes that take you back several spots and with ladders that land you several spots ahead.

The author designed and implemented this game with Rust Programming Language.

The objective of choosing this design problem was to learn about the following mechanisms that Rust offers to developers with several of its libraries/crates:

Design concepts (Hash/Linked lists implementation), User interaction (I/O: Command line, Graphical, Audio-Visual), GUI, and Program scalability.

### Design approach & Implementation:

The author divided the problem into the following categories:

#### a) Designing the board layout

This included designing the initial board layout, and using different ways to track and update position of every player on the board. The board layout had to also include the various positions of 'snakes and ladder's on it. The author initially started with using a Piston crate for 2D graphics to design the board, and linked lists to handle snake and ladder behavior within the board. But soon found it to be too cumbersome to debug. While 2D graphics worked for a simple board layout, for a complex changing board with multiple updates, the author found it too laborious a code for maintenance.

So, shifted the board layout to terminal based design with 2D arrays, and also shifted to use HashMaps for Snake and Ladder positions. Handling 2D arrays and Hashmaps in Rust provide to be very convenient, and helped in code maintainability and readability.

#### b) Multi player handling

With the 2D array approach, the author could handle as many players but restricted it to a maximum of 4 for the scope of the project. The author implemented a Player struct (akin to a player object) to contain player related information – including avatar, position, etc., and had associated functions defined to handle all player related data. Using this approach helped in encapsulating all Player related functionality at a single place. Also, to incorporate the roll of dice, used the random number generator crate – which worked as expected.

#### c) Display & Input

The author tried several approaches for 'displaying' the game to end user. The first few trials were on Piston 2D graphics library with grid. While the display part itself worked cleanly, updating and changing of positions and handling the hashmaps for snakes and ladders seemed a bit cumbersome (in the given time frame). The author then shifted to a terminal based approach, and used the 'termion' crate to handle all I/O related activity (including inputs to the game), and found it to be very convenient in handling all terminal activity, along with the flexibility to handle other aspects of code, like Hashmaps.

With Termion, the author was also able to incorporate color based terminal activity aiding end user experience (which was incorporated through the feedback provided by fellow-code reviewer)

Throughout the project, the author has used the Rust way of programming sections into several 'struct and implementing functions', providing ease of code handling, and scalability – while also being safe.

#### Testing:

Wrote small unit tests to check each of the implemented functions to ensure modular level testing. And, integrated the whole code and tested as a final product on the terminal.

#### Current Status:

Snake and ladders works for 2 to 4 players where players get a chance to roll dice which is randomly generated. After rolling their dice, they move according to the number with ladders taking them higher and snakes putting them down in the board. A colorful display of the board with the players' position on it, aided with some text inputs for players, guide the players through the game.

#### Challenges faced & Future work:

Some of the challenges faced during the project were: Like mentioned above, using the Piston graphics crate to handle board design and layout. But, that is something that can be handled for 'Future work' to provide gamers a fancier GUI experience. Also using animation and audio crate would be fun. Some additional testing infrastructure and mechanism can also be added to make the design more robust.