Snack and Ladders Game

- Abhishek Maheshwari

Scope/Requirements -

- Standard Snack and Ladders game along with the following tweaks.
- Board size can be flexible.
- There is no limit on the number of players but we will end the game as soon as 3 players win. If the total number of players is less than 3, then the game will continue until one player is left.
- undo/redo is out of scope.
- We need to be able to replay the game.
- The game is offline in nature.
- There is only 1 die.

Features and Assumptions

- Default board size = 100.
- The default number of faces for dice = 6
- Human Players are assigned the id of the form “H1”, and “H2”.
- Computer Players are assigned id of the form “C1”, and “C2” and are auto-assigned names with “Computer” as prefixes.
- We are generating snakes and ladders in the Game service. We can do this using a random number generator (hardcoded for the scope of this exercise).
- Getting a six on the die does not let the user throw the die again.
- If a player is human, we prompt them to enter a key. This is to simulate the throw of the die. If a player is a computer, we throw the die without the prompt.
- We declare a winner only if the user reaches 100. If a die throw makes the user go beyond 100, we dismiss that throw and move on with the next player.

Notes

- We can also provide the option to users to provide custom snake and ladder positions. In this case, we will need to validate the input. For example - if the snake is at the 100th (last) position, then the game will never end.
- Do we need Singleton(?)
- For this exercise, all implementation code is in *.hpp. Ideally, the implementation should be in *.cpp and *.hpp should just have declarations.
- Some sample unit tests are also added. Using Xcode’s XCTest framework with a hack to make it work for CPP files. XCTest is otherwise available only for Objective C and Swift. More details about the hack are here - https://abskmhswri-er.medium.com/unit-test-in-c-with-xcode-9061df3b05e3
- All classes have descriptions on the top and a few test cases mentioned at the end in comments. Relevant comments are present in the code as required.