EC327

Prof. Densmore

Team Name: Emotional Damage

Team Member: Chengze Zheng, Yuhan Li, and Yuke Li

Project Name : Peg Solitaire

EC 327 Final Project: Peg Solitaire

**Game Description**

Peg Solitaire is a single-player game played on a board with holes in it. The board is typically a grid with 7 rows and 7 columns, for a total of 49 holes. The game starts with a board that is mostly empty, except for a few pegs placed in specific holes. The player's goal is to jump pegs over one another, removing the peg that was jumped, until only one peg remains on the board. To play the game, the player selects a peg that is adjacent to an empty hole and jumps it over another peg into the empty hole. This removes the peg that was jumped from the board. The player can only jump pegs horizontally or vertically, not diagonally. The player continues to make jumps until no more moves are possible, at which point the game ends.

**Implementation Logic:**

The game logic for a program implementing Peg Solitaire might involve several different components. First, the program would need to represent the game board and its current state, including the locations of all the pegs and empty holes. The program would also need to provide a user interface for the player to make moves and view the current state of the game. Next, the program would need to implement the rules of the game, including the rules for making valid moves and the conditions for ending the game. This would involve checking the locations of pegs and empty holes on the board and determining whether a given move is valid, as well as detecting when the game is over because no more moves are possible. Finally, the program might include additional features, such as the ability to solve the game automatically using a computer algorithm, or to save and load game states. These features would require additional game logic to implement.

**Working Log:**

11/13/2022:

Project Outline:

Presented: All the members are presented

Time Duration: 2 hr

Minute: Discussed ideas with each other, proposed different ideas and brainstormed on what to do, and decided to build a peg solitaire game from Yuhan’s idea. Planned

11/20/200

Project Description: Peg Solitaire

Presented: All the members are presented

Time Duration: 2 hrs

Minute: First, we looked at the game logic of the peg solitaire, afterwards, we searched existing github repo and webpage that have this game and see how they built it, however, none of them are using cpp to write a complete full working game. In the meanwhile, Yuhan was amused by EasyX that can be used to build interfaces while other parts of the programming game logic remains in cpp, EasyX is basically an open source library that we can run on Visual Studio. After we decided upon on what language we want to use, we submitted the proposal and wait for approval

11/27/2022

Game Logic Work out in Unix Command Window

Project Description: Peg Solitaire

Presented: All the members are presented

Time Duration: 4 hrs

Minute: This meeting was set up for discussing the breakdown of the work and assigning the works to individual members, before doing anything to the interface EasyX. We decided to divide it into three part, one person will initialize the game board, assign the location coordinate for the peg, another person will write the class of peg and member functions of how the peg is updated and moved, last person will write the main script which is the workflow of our game.

Assigned as following：

Yuke -> Model class

Yuhan -> Initialization

Chengze -> Main

11/27 - 12/01/2022

Game Logic Wrap Up

Presented: Via Zoom and Individual Work Session

Time Duration: 4~6 hrs

Minute: Chengze finished his part and passed the work to Yuke and Yuhan in group chat, Yuke finished her part next and Yuhan finished his part and helped wrap up everything to have everything work. Yuke and Yuhan upload the changes to Github.

12/05/2022

EasyX building Interface

Presented: All member presented

Time Duration: 2 hrs

Minute: After individual learning on the tutorial Yuhan sent out about EasyX, we started hands on building the interface, Yuhan finished the initialization of the board during the meeting session and passed on to Yuke and Chengze to finish the game logics and run the game. Also, during the whole process, Yuhan provided all the hand-drawn image source for our game

12/07/2022

EasyX Wrap Up

Presented: Yuke, Chengze in person, Yuhan via Zoom

Time Duration: 2 hrs

Minute: Yuke and Chengze wrapped everything up and tried to add the feature for users to cancel clicks if they have invalid first move, but did not turn out too well, therefore, removed this feature later. Yuhan suggested adding animation when user defined first clicked.

12/11/2022

Comment and Video Wrap Up

Presented: All member via Zoom

Time Duration: 2 hr

Minute: All group members worked together to finish the documentations

**Final Workflow of the Game**

1. After enter the game, it will display a command window for user to prompt what style of board they want
2. After enter in the number, it will pop a window that have all the pegs initialized with one empty spot
3. User first click on the starting peg that will have an orange highlight, and selected destination point to land, if it is invalid destination, command window will indicate it is a invalid move, and user have to repeat number 3 again
4. Once no valid moves exist, the pop up window will close, the command window will either indicate that Users have won the game or there are how many pegs left on the board.

**Design Sketch (Pseudo Code for Early Stage)**

* source.cpp
  + Prompt and error checking for input (While loop until no valid move possible)
    - Select different boards to play
    - Select two Pegs to move
* Model.cpp
  + Initialization (Board setup) and visualization for three type of boards, Update (check for valid move), Check for Game Over (check if there is valid move left
* Peg.CPP
  + int x,y (coordinate)
  + char status: empty, occupied, Selected