Deliverable II

Task	Difficulty	Importance	Owner
Main menu GUI (New game, Resume, Stats)	Medium	Essential	Joe
Board GUI	Medium	Essential	Chia-Chun
Instructions	Easy	Essential	Gavin
Deck	Easy	Essential	Peter
Player AI	Hard	Essential	Gavin
Game Pieces and optional movement locations	Hard	Essential	Joe
Stats	Easy	Essential	Peter
Saving game and resuming	Hard	Essential	Peter
Pawn movement and Card draw animations	Hard	Nice to have	Chia-Chun

System specifications:

Main menu

1. Start a new game
    a. Enter player's name
    b. Choose color
    c. Decide how nice and how smart each computer opponent is (mean/nice) based on top/left/right
    d. Start
2. Resume a game
3. Summary/reports

Start

1. Create the MAP
    a. GUI
        i. Display the map
    b. 4 PLAYERs
        i. Display the player's name
        ii. Rotate the map to match player to the chosen color
        iii. Map computers to different colors

iv. Create 4 PAWNs
1. Set the color the same as the player
2. Put every pawn at start position

v. Set the AI mode of every player

c. DRAW
i. Create a queue of 45 CARDs and shuffle()
ii. Put the pile of cards at draw position

d. DISCARD
i. Create an empty queue of CARDs
ii. Put the pile at discard position

2. Relaxation start
a. Put one pawn of each player at the position on the track below start
3. Pick a player to start first and run clockwise
4. The first player starts drawing a card

Resume

1. Load in saved data from MySQL
2. Create the MAP
a. GUI
i. Display the map

b. 4 PLAYERs
i. Display the player's name and current location and color of pawns
ii. Load in each computer player AI from last saved game

c. DRAW
i. Load deck of cards from draw pile of last saved game to draw area

d. DISCARD
i. Load deck of cards from discard pile of last saved game to discard area

3. Player whose turn it was at save of previous game is up
4. Player draws a card

Draw

1. If no pawn on track
a. Draw 1: move one pawn to the track
b. Draw 2: move one pawn to the track and draw another card
c. Draw Sorry!: Take one pawn from your START, place it on any space that is occupied by any opponent. If it's impossible, discard the card
d. Draw others: discard the card
2. Any pawn on track
a. Follow the instruction of the card
3. No cards in draw pile(end of deck array)
a. Clear cards in discard
b. Create and shuffle cards in draw

Move

1. If there are more than one options, choose one
2. If the pawn will land on a pawn in the same color, choose other options or discard the card
3. If the pawn will land on a pawn in different color, bump it to start
4. If the pawn will land on the triangle of slide in different color, bump all pawns in the area and slide to the end of slide
5. If the pawn will go over the position below safety zone in the same color, move to safety zone instead of on the track (the pawn cannot move backward into safety zone)
6. A pawn can only go to Home by exact count
7. If the pawn cannot move, discard the card
8. If there are 4 pawns in home, the player wins

Save/Resume

1. Save all information in MySQL (player, pawns, game, deck tables)
    a. Player position (up, down, left, right)
    b. Player pawn locations (locations 1-60, home, or start, for each pawn
    c. Player AI setting (smart, dumb, nice, mean)
    d. Pawn color for each player (blue, green, red, yellow)
    e. Current deck order, current card, store using serialize and unserialize

Help

1. Provide rudimentary help screens

CPU smart vs dumb

1. Dumb CPU - chooses all of its moves at random
2. Smart CPU - uses a scoring algorithm to calculate the most efficient move
    a. This includes moving a piece into the SAFE ZONE/HOME if possible
    b. Calculating a way to get a piece the closest to home as possible, by moving a set amount of spaces or using the abilities on the cards

CPU nice vs mean

1. Nice CPU - computer will always try and avoid taking (BUMPING) another piece, algorithm will check to see if there is a move possible without BUMPING a piece. If so the CPU will avoid BUMPING, unless there is no other option but to BUMP, in which case it will BUMP.
2. Mean CPU - computer will always try and take take (BUMP) another piece. Algorithm will check to see if there are any possible moves that BUMP another piece, if so it will take the piece, else it will move regularly.

Design Documentation

Descriptions of GUIs

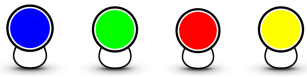
The main menu contains four options: new game, resume, stats, and exit.

1. When new game is clicked the GUI changes to a menu which allows the player to select their color by clicking on one of the four colored pawns. Next a menu opens that shows the player difficulty options, allowing them to choose between a dumb & nice, dumb & mean, smart & nice, and smart & mean computer player. After this the menu will disappear and the game will load.

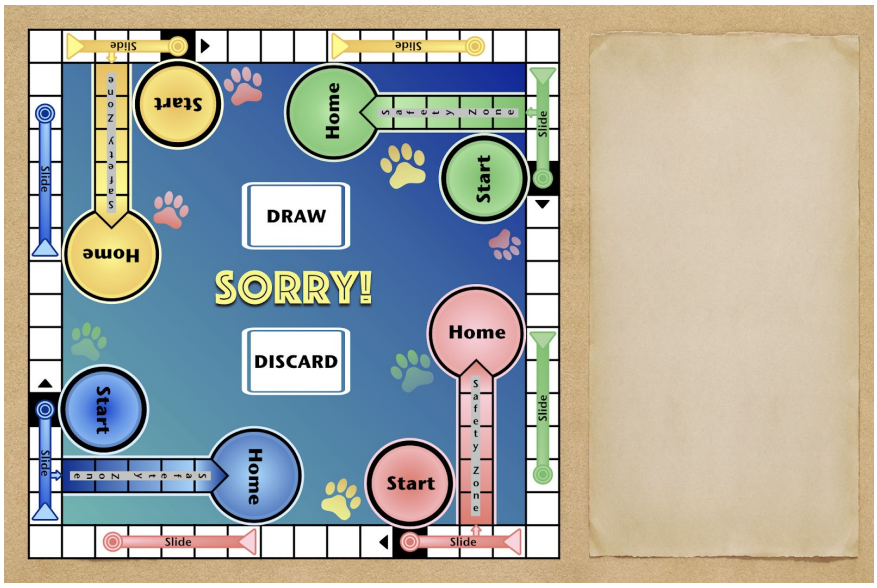
2. Clicking resume will skip the new game option selection and instead load saved variables from a file to restore the state of the saved game. The board will then be loaded and displayed to the player.

3. Clicking stats will simply show a screen that displays game statistics to the user.

4. Exit will quit the game.

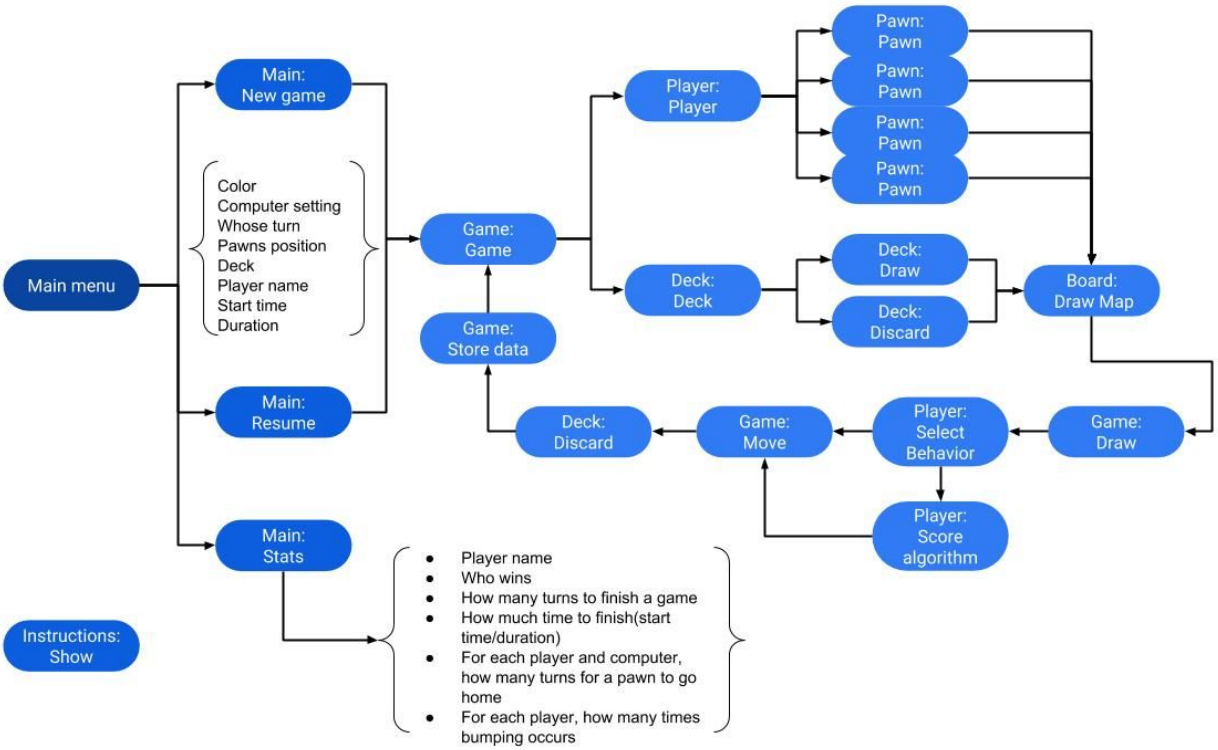


These are pawns in four colors, and there will be four in each color on the track.

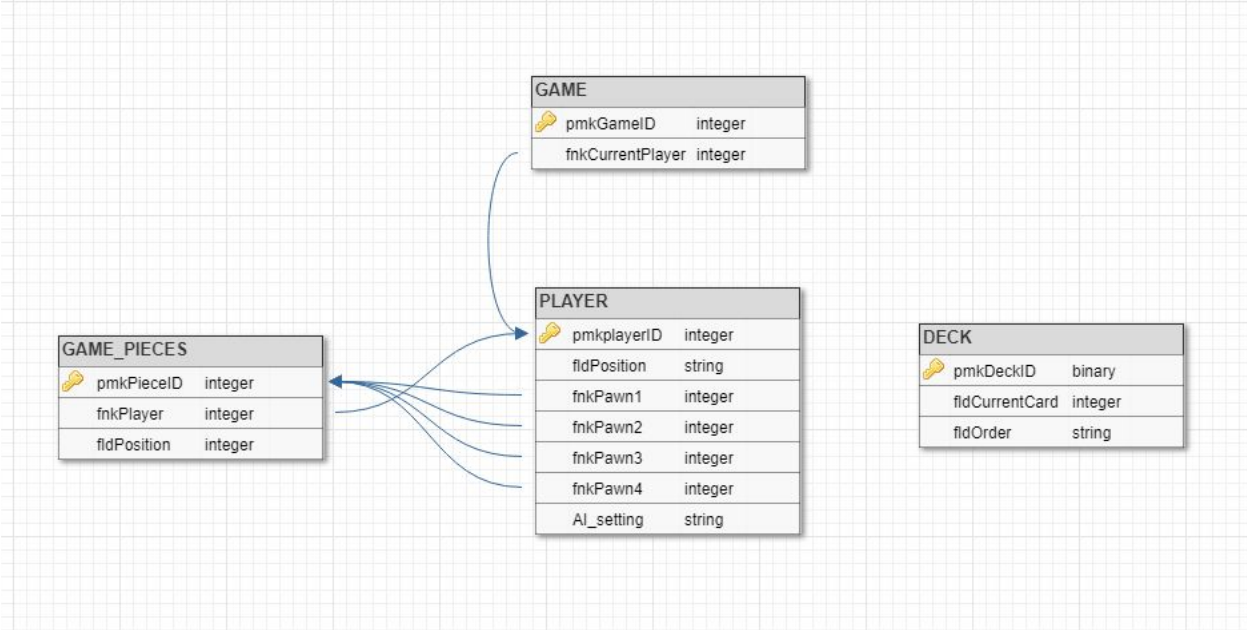


The square part on the left side is the map of the game, and there will be pawns on the track. Cards will be piled on the draw and discard sections. Some buttons will be put at the top of the paper on the right side. Game messages will also show in the paper.

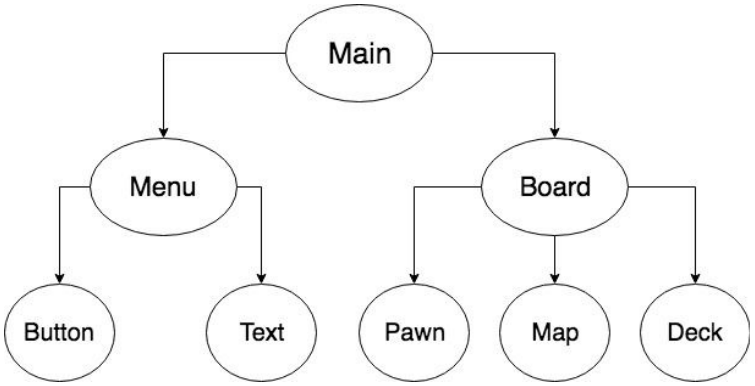
Diagrams of major components



Database schema



Class hierarchy



Implement "dumb" vs. "smart" computer opponents

CPU smart vs dumb

1. Dumb CPU - chooses all of its moves at random
2. Smart CPU - uses a scoring algorithm to calculate the most efficient move
    c. This includes moving a piece into the SAFE ZONE/HOME if possible
    d. Calculating a way to get a piece the closest to home as possible, by moving a set amount of spaces or using the abilities on the cards

CPU nice vs mean

1. Nice CPU - computer will always try and avoid taking (BUMPING) another piece, algorithm will check to see if there is a move possible without BUMPING a piece. If so the CPU will avoid BUMPING, unless there is no other option but to BUMP, in which case it will BUMP.
2. Mean CPU - computer will always try and take take (BUMP) another piece. Algorithm will check to see if there are any possible moves that BUMP another piece, if so it will take the piece, else it will move regularly.

Implement the suspend game/resume game functions

Save/Resume

2. Save all information in MySQL (player, pawns, game, deck tables)
    f. Player position (up, down, left, right)
    g. Player pawn locations (locations 1-60, home, or start, for each pawn
    h. Player AI setting (smart, dumb, nice, mean)
    i. Pawn color for each player (blue, green, red, yellow)
    j. Current deck order, current card, store using serialize and unserialize

Picture of our SORRY! Game