Project Name: Ashta Chamma

Overview:

Our Project is an Indian version of a Ludo game, which is very common and one of the oldest games back in India. This game is played on a board.In most Indian homes you can see a board or drawing of this game. This board typically consists of 25 squares which are divided into 5 Rows X 5 Columns. Each block is uniquely defined by the type of the block. A block can be a safe place or it can be an unsafe place. Each player has a turn in sequence and is asked to flip a die. Each player has a couple of pawns and are moved according to the dice value. Each player has a fixed path in which the player's pawns move, which is in an counter-clockwise direction. For every sequential turn of dice if a value of 4 or 8 reflected on dice resulting in the entrance of a single pawn (if value 4) and 2 pawns (if value 8) [Normally we use shells to play this game but instead, we are using dice with numbers: 1,2,3,4,5,8 on it].

A Pawn moving on the board is completely based on dice reading and the user's choice of choosing between the 2 pawns that the user was provided.

A Player should knock out at least one pawn of the opposite player to get access to the inner gateway2 (Inner squares: the squares that are touching the home in Fig-1,2). Each player's piece must completely traverse the outer squares before moving into the inner squares. User continues to move their pawns logically to knock out other players' pawns, To escape from other players' pawns and to enter into a safe home (which is in the center of board and final destination). The first player whose 2 pawns enters safe home "The Player is the winner".

Initially In our development we have set two different modes of the game which are 1.A Player against a computer 2. A player against a Player. In case of Player against a player the option of moving the specific pawn is completely the player's choice. In the case of computer players, execution of the probability case is made in a way that it makes the best possible choice.

Functional Requirements:

FR1: Once the Level is selected , the game starts.

Level 1: A Human player trying to knock out and win against another Human player.

Level 2: A Human player trying to knock out and win against another Human player with less probability of dice rolling 4 or 8 in the game

FR2: Each player has 2 pawns. After selecting mode between Level1 and Level 2, the level panel will be replaced with a game control panel which consists of dice, players and game board.

FR3: Users rolls the dice randomly and a value which is a number will be generated randomly

Dice possible values are assumed as :1,2,3,4,5,8 according to our game.

Initially. when there are no pawns of a player on the board:

A: If the value is 4 ,One pawn enters the board.

B:If the value is 8,Two pawns enter the board.

C:Any other value will be ignored

FR4:Users rolls the dice randomly and a value which is a number will be generated randomly

Dice possible values are assumed as :1,2,3,4,5,8 according to our game.

When a player has only one of his pawns on the board:

A: If the value is 4, Player has 2 options ->

i) Player's second pawn can enter the board.

ii) Player can move the existing pawn on the board by 4 squares.

B:If the value is 8,Player has 2 options ->

i) Player's second pawn can enter the board and move 4 squares.

ii) Player can move the existing pawn on the board by 8 squares.

C: If any other value i.e 1,2,3,5.The existing pawn on the board will be moved by that value(if dice value is x then the pawn moves by x squares).

When a player has two of his pawns on the board:

A: Player has an option to move any one of his pawns.

FR5: The square blocks of the board are divided into 2 types which are safe squares and unsafe squares. Safe squares are represented by green color(ref : prototype board) and unsafe squares are white in color.If a player enters an square occupied by an opponent's pawn:

1. Unsafe square : The opponent's pawn knocked out. One player can knock out another player in an unsafe square
2. Safe square : One player cannot knock out another player in a safe square.Both players' paws can share a place in safe square.

FR6: The opponent player's pawn which is knocked out will be out of the board and will start the play again from the first, it can enter the board again if and only if the opponent player rolls dice to 4/8 as mentioned in FR4,FR5.

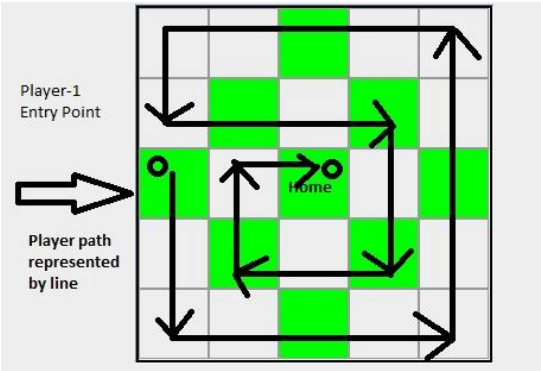
FR7: Knock out criteria is that if one player is "Player-A" in Unsafe square.Next player turns to enter the same unsafe square which results in knock out of player-A.

FR8:Once a player gets a minimum of 1 knock out point against another player will be eligible to move forward in the specific path towards home with the reference of Fig-1.Fig-2 below. If the player is not knocked out after completing the outer loop once, it will go on moving on the outer loop until it knocks out any of the players and knocks out the opponent player pawns at least once.

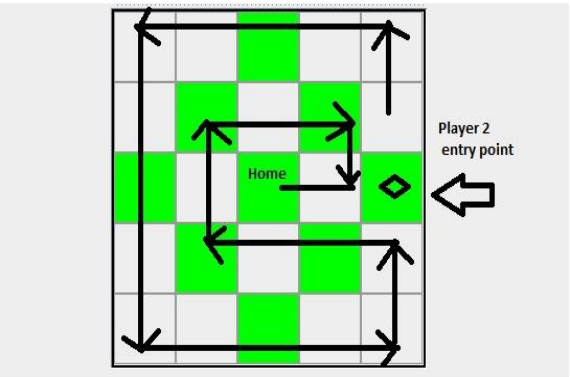
FR9: User continues to move their pawns logically to knock out other players' pawns , To escape from other players' pawns and to enter into home (which is in the center of board and final destination).

FR10:The first player whose 2 pawns enters safe home “The Player is the winner”.

FR11: A player winning with a minimal number of dice moves will be asked to enter his name and his score will be stored as the highest score.

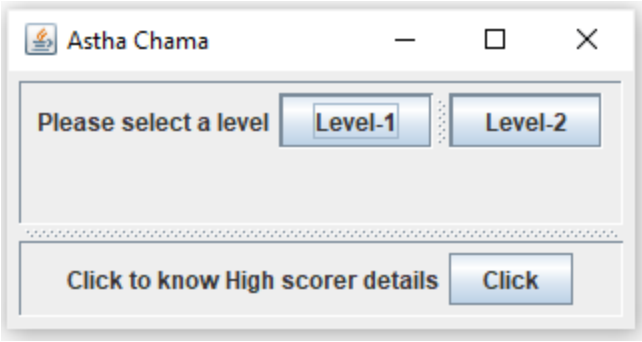


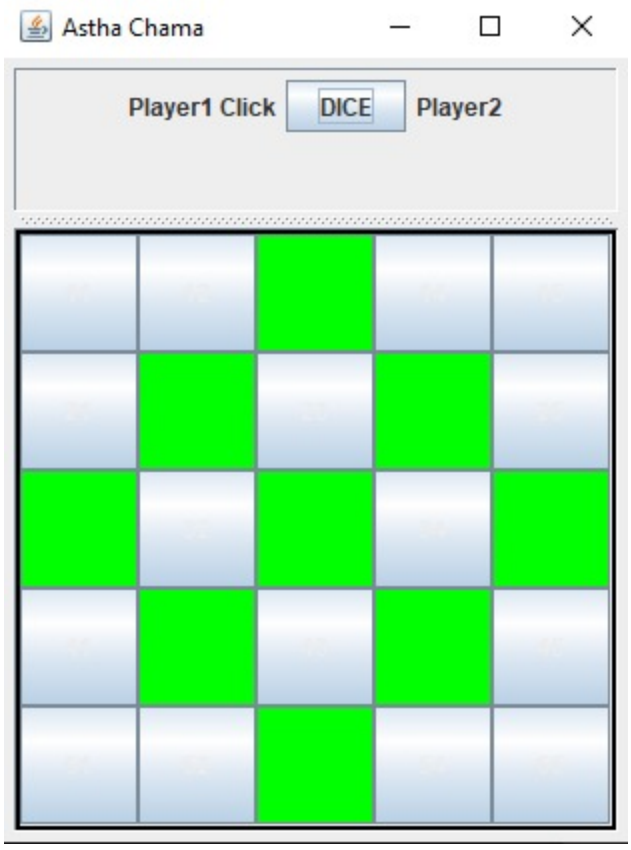
(i)Fig-1

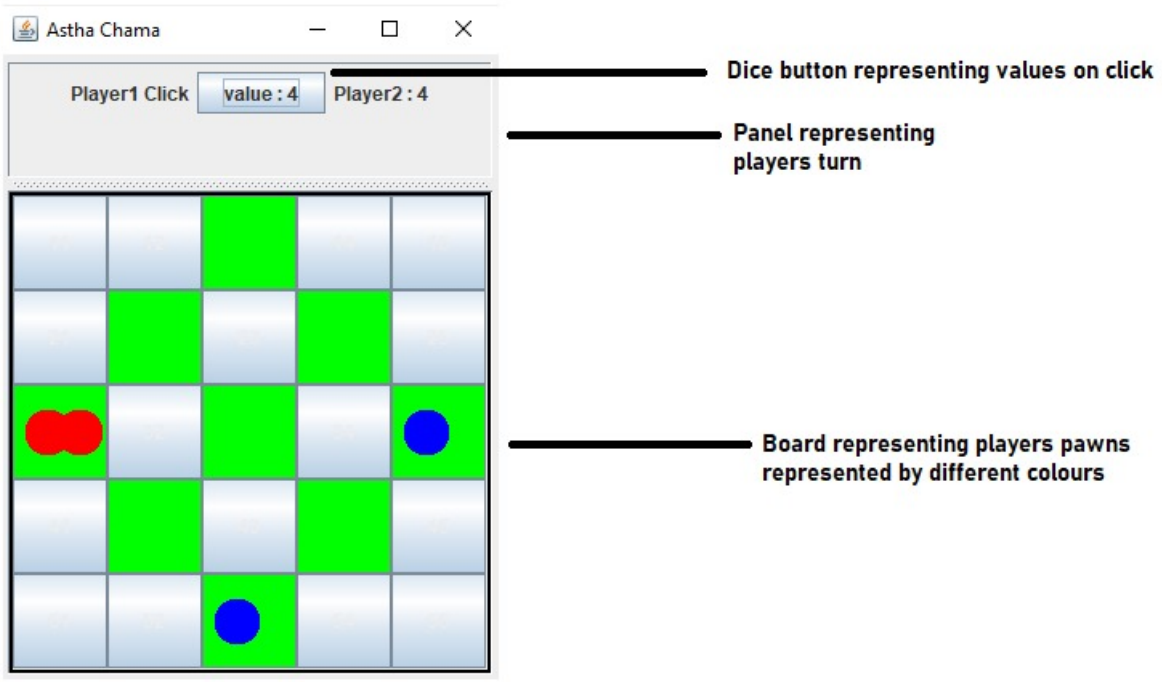


(ii) Fig-2

User Interface Prototype:
(Include a screenshot of the user interface prototype, developed using Java SWING A PI)




Astha Chama
Player1 Click
DICE
Player2



User Interface Requirements:

UIR1: Once the game opens a player needs to make a choice between Level-1 and Level-2.The screen initially displays levels.
UIR2: A Board will be divided into 2 types of square blocks identified by color. Example: A green color block is a safe square whereas white color block is an unsafe square
UIR3: After selecting mode between Level1 and Level 2, the mode panel will be replaced with a game control panel which consists of dice and players.
UIR4: Once you click on the Dice FR4 will be executed.(dice value will be displayed on screen and pawns move according FR4)
UIR5: When 2 players start playing the game and rolling the dice, FR5 will be executed.
UIR6: If FR6 is executed (means a pawn gets knocked out) The pawns get out of board into the initial box.
UIR7: After FR10 is executed the winning message displayed on screen.
UIR8: Highest score recorded using a text enter box.

Future feature additions:

1. We have designed our game with respect to design patterns. There are many features that can be added to our game. One of the major features will be adding different types of players to the game.

a) In our game the human player is implementing the interface player.
b) In case there is any requirement to add a different kind of player such as a typical computer player/ human player with more probability of winning.
c) We can increase the number of pawns assigned to a player.

2. Improving the difficulty level of the game.
3. We have maintained our user interface in different classes by differentiating the panels and adding it to the main panel. In future if a developer who needs to add a feature to UI can just add a panel and follow by adding it to the main panel.

Appendix

List any sources you used to gather ideas for your application.

Project Plan:

Requirement	Deliverable 1	Deliverable 2	Deliverable 3
FR1	*		
FR2	*		
FR3		*	
FR4	*		
FR5		*	
FR6		*	
FR7			*
FR8			*
FR9			*
FR10			*
FR11			*
UIR1	*		
UIR2	*		
UIR3	*		

UIR4		*	
UIR5		*	
UIR6		*	
UIR7			*
UIR8			*