

ASHTA CHAMMA

Mirukula Pravalika

Şandeep Kumar Şutharapu

Şardarni Harjoth Kaur



CONTENT:

Intro to “Ashta Chamma”

Rules of the Game

Initial Prototype

Functional Requirements

UML Diagram

Testing

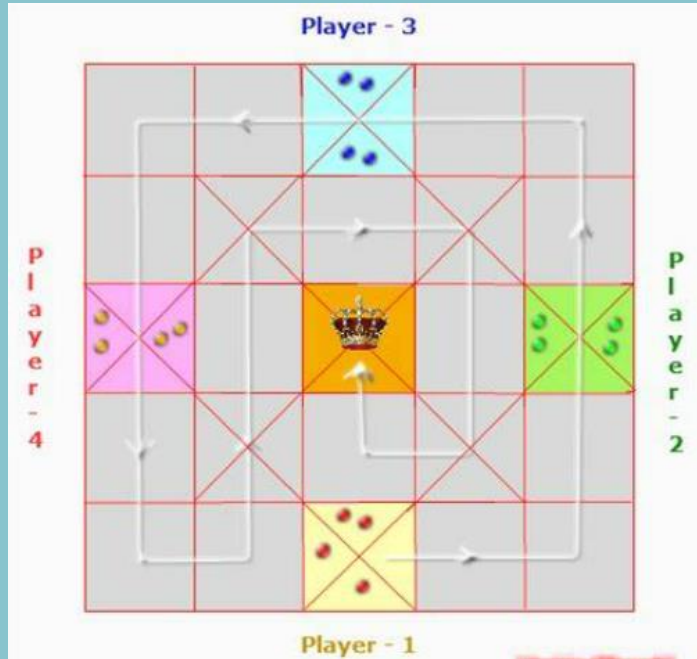
Demo

Future Extensions

Reflection



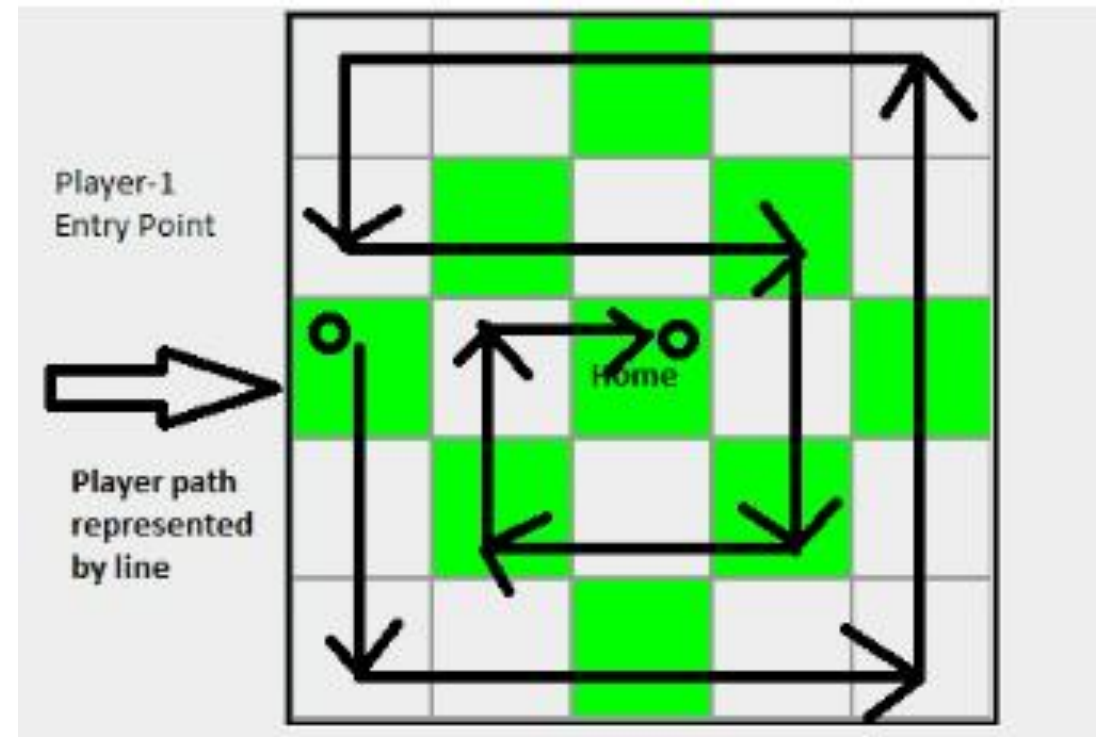
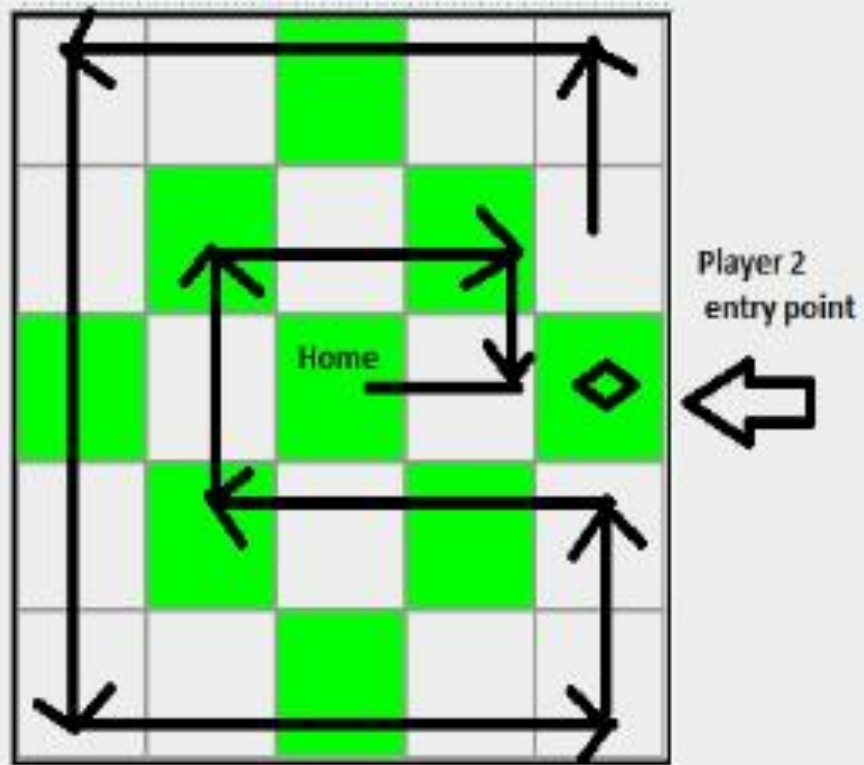
Intro to Ashta Chamma:



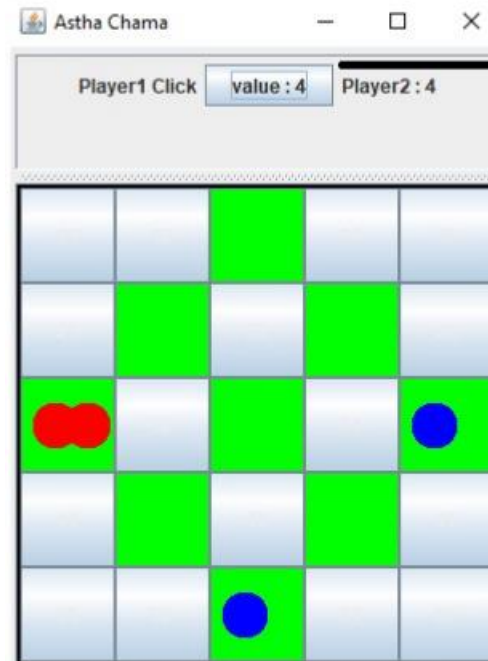
- Indian version of a Ludo game, which is very common and one of the oldest games back in India.
- Consists of 25 squares which are divided into 5 Rows X 5 Columns.
- Each player takes a turn to roll the Dice.
- In telugu, If Dice rolls 4 it means Ashta and If Dice rolls 8 its chamma
- A block can be 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..

Rules of The Game

- Each player has a fixed path in which the player's pawns move, which is in counter-clockwise direction.
- The possible values of the Dice are 1, 2, 3, 4 and 8. Depending on the number rolled, the player can move one of their pawns that many number of squares on the board
- Dice value is n than moves by n squares.
- Killing spirit turns true for a player then player's pawns can move into inner loop.
- For a player to win, he/she must move all their pawns to the center square.



Initial Prototype



Dice button representing values on click

Panel representing
players turn

Board representing players pawns
represented by different colours

Requirements:

- Functional Requirements:

Dice: Level-1 : Normal Play B/w 2 players
Level-2: Less possibility of Dice rolling 4 and 8.

Player :

- Tracking No of pawns on board
- Tracking killing spirit

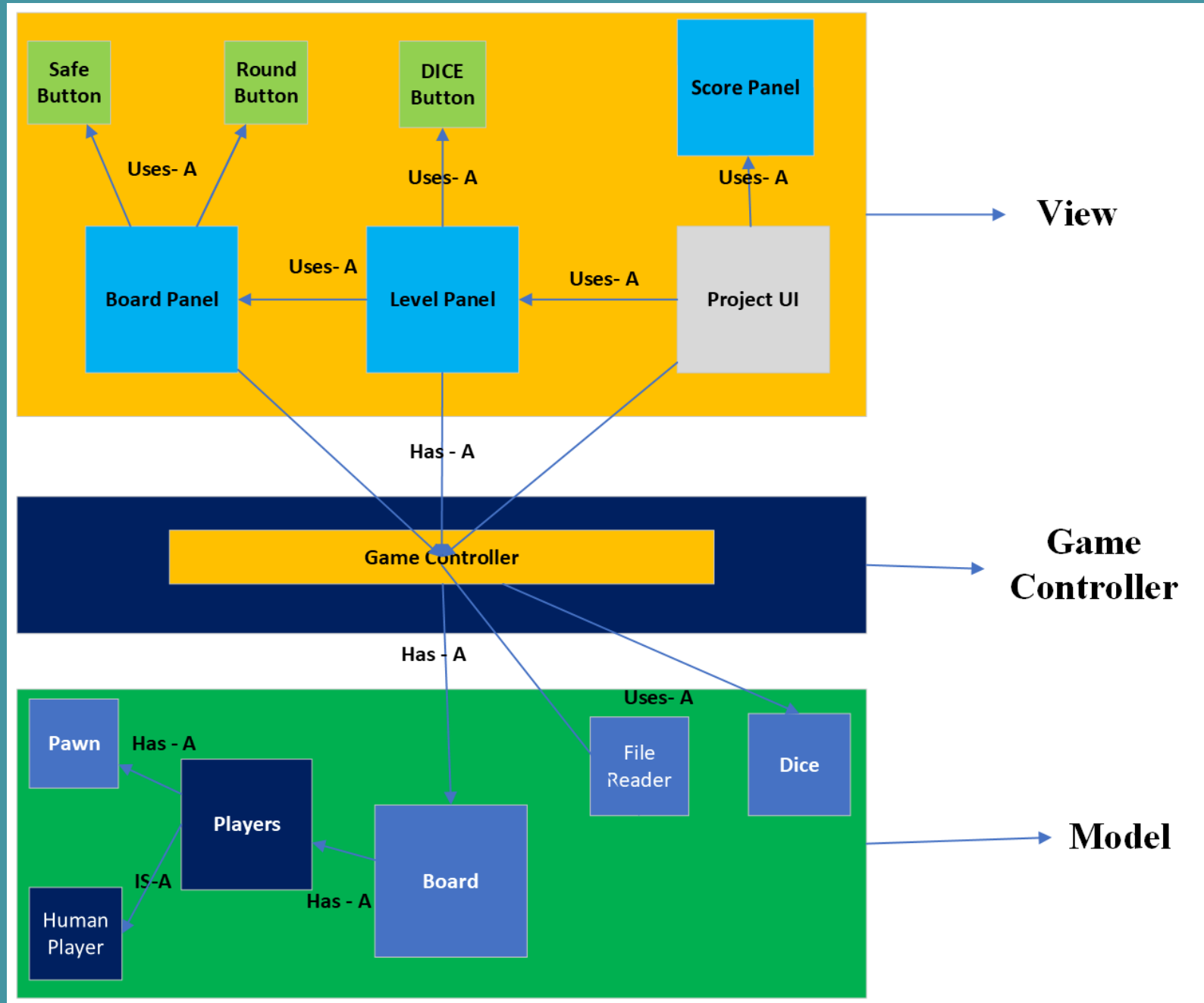
Board :

- Placing pawns on board
- Updating pawn positions
- Knockout Checking
- Storing pawn positions
- Checking if the game is complete

- User Interface requirements:

- Board with 2 Different color squares. Panel with level 1 and 2 creation.
- Pawns of 2 different colors of 2 players movement according to dice value on their turn.
- On winning the game new panel with highest score as well as their name

UML Diagram:



Testing:



- Classes of Model:
 - Board
 - Pawn
 - Player
 - Dice
- No issues while testing.



Future extensions:

- 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.
- In our game the human player is implementing the interface player.
- 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.

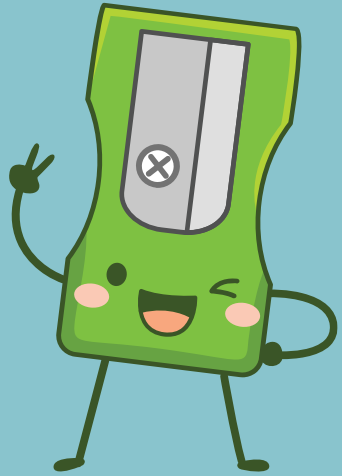


- We can increase the number of pawns assigned to a player.
- Improving the difficulty level of the game.
- We have maintained our user interface in different classes by differentiating the panels and adding it to the main panel. In future if a developer who needs to add a feature to UI can just add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



Reflection

- At every part of our development, we have faced no of issues in contrast the few major problems we faced are to give a clear-cut picture of user interface on every condition of the game which led us to create a tree of if-else conditions.
- Issue while taking input from button listener and moving the pawn on the interface.
- Implementing serializer.



THANK YOU