**Final Project**

**Ding Chen**

**Tic Tac Toe**

**ANALYSIS PHASE**

# **Functional Specification**

This TicTacToe is designed for human player plays with human player and human player play with AI in two level in real life. A move is guaranteed to be valid and is placed on an empty block. Once a winning condition is reached, no more moves is allowed. A player who succeeds in placing n of their of their in a horizontal, valtical, or diagonal row wins the game.

# **User Manual**

The highlight playname is the current player, click the button on the grid to play the game rules follow the TicTacToe game, if X is winning there will be a box showing winning message.User can reset the board/reset the game by clicking “reset” button. User can go back to main menu by click “Exit” button and close “Yes” when the confirm box show up.

Xs and Os is for two players or one player with AI computer. X and O, who take turns making the spaces in a 3\*3 grid. The player who succeeds in placing three of their marks in horizontal,vertical, or diagonal row is the winner.

There are two level ( easy and normal ) if human play with computer AI, the computer goes first. Who is moving that name is highlighted ( Yellow )

Check for Win: if the player has two in a row, they can place a third to get three in a row.

Block:if the opponent has two in a row, the player must play the third themselves to block the opponent.

Blocking an opponent’s fork: if there is only one possible fork for the opponent, the player should block it. Otherwise, the player should block all forks in any way.

# 

# **Use Cases**

1. Use Case for run the game

1.show the game Frame

Variation 1

Not to running

2.Use case for close\_Button

1.Program end

Variation 1

No response

3.Use case for menu option

3.1 human VS human

Show “ input dialog to get 2 players name

Playboard Frame showup

Player1 = human1

Player2 = human2

3.2 Human VS Computer(easy)

1.get 1 player name

2. Show playboardFrame

Player1 = AI(easy)

Player2 = User

3.3 computer (Medium) VS Human

1.get 1 player name

2. Show playboardFrame

Player1 = AI(medium)

Player2 = User

Variation #`1

Click exit button

gameover

4. Use Case for Click Exit button

4.1.Ask question to confirm if user want to quit the board and back to main menu

4.2 if “yes” back to main menu

4.3.if “No” back to board

Variation #1

Not back to main menu when click “ yes”

No response when clicking

5.Use Case for into the playboardFrame

5.1.each playboard game button clicked(“9” grid game)

Show the mark ‘X’ or ‘O’ in the red

The button showing the mark can not be clicked again

The mark will rotate ‘X’ or ‘O’ with each turning going current player’s name is highlighted

**DESIGN PHASE**

**Identify Classes**

PlayBoard

Player

Game

HumanPlayer

ComputerPlayer

**Class Responsibility**

PlayBoard: store each of point so that have 3 types of 3\*3 grid ; visit each of point in the grid. -1 and 1 mean that it has x in the board 0 means nothing.

Player: it creats a name to output the result. For example, player VS player he may calls player1 or player2, but player VS AI he may calls “you”. I add Get\_name() to output result easily. Movement which records currently where you are. Mark records that how many players we have. Hand records who is gonna be first.

Game: let player entry into the game and count that who wins.

PlayerHuman: read\_human read in the input from players.

ComputerPlayer: designed weak and strong algorithm to play with human player.

**Class Relationship**

There are two relationships in TicTacToe which is inherit and composition

1. inherit:

Board is inherit of Frame

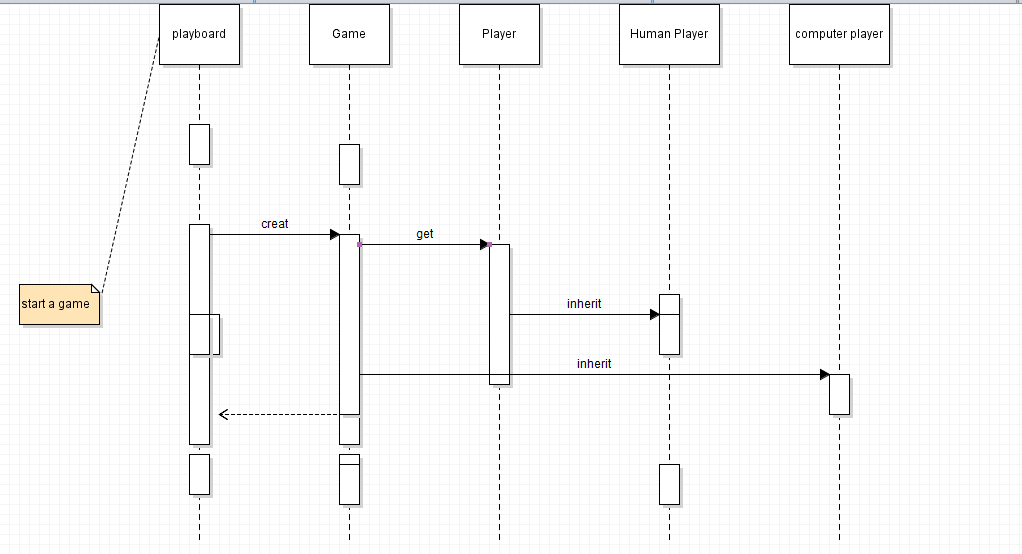
playerHuman is inherit of player

PlayerComputer is inherit of player

1. composition:

Player and playboard are compositions of Game.

**Sequence Diagram**

****

# **State Diagram**

