```
Author: Paul Ingram
Created On: May 2, 2022, 11PM
Purpose: Mancala v5
```

System Libraries
cstdlib
string
iostream
time.h
algorithm
list
map
stack
queue

User Libraries
Board.h
SimGame.h

Global Constants
none

Function Prototypes
none

(main)

**Set Random Number Seed
and
Declare Variables**
srand(time(NULL))
int gameMode

Prompt Input
"Main Menu..."
"Enter 1) 1v1"
"Enter 2) 1vBot"

gameMode==1

Declare Variables
string player[2]
int curPlr=rand()%2

Prompt Input
"Enter Player 1 Name: "
"Enter Player 2 Name: "

Board Constructor
board(player[0], player[1])

Board::playGame()

Declare Variables
string name

Prompt Input
"Enter Player 1 Name"

Simulate Constructor
sim(name);

return 0

Board::playGame()

↓

Board::setCurPlr(rand()%2)

↓

◇ Board::gameOver() → ⬭ Declare Variables / int choice → ◇ returnCurPlr()==0 → **Board::printBoard()**

Board::printBoard() ↓

Prompt Input
"player[0]'s turn choose
a cell 1-6 LR on top row

↓

Board::player1turn()
Board::setCurPlr(1)

Board::gameOver() ↓

Board::announceWin()

↓

(return)

returnCurPlr()==0 ↓

Board::printBoard()

↓

Prompt Input
"player[1]'s turn choose
a cell 1-6 LR on bottom row

↓

Board::player2turn()
Board::setCurPlr(0)

```
┌───────────────────────┐
│ │  Board::player1turn() │ │
└───────────────────────┘
            │
            ▼
┌───────────────────────────────────┐
│         Declare Variables          │
│         int capCount=0             │
│         bool capture=false         │
│ list<int>::iterator it=plr1pits.begin() │
│         int marbles=*it            │
└───────────────────────────────────┘
            │
            ▼
      it!=plr1.pits.end()  ──▶  marbles>0  ──▶  marbles--
                                                 *it+=1;
                                                
                                    marbles==0  ──▶  *it==0  ──▶  *it=0
                                                                  capture=true

                                                              capCount++
            │
            ▼
   marbles>0  ──▶  marbles--  ──▶  marbles==0  ──▶  goAgain=true

                          plrScore[player[0]]++

                                     capture  ──▶  Board::player1capture(capCount

                              it++              return
```

```
SimGame::SimGame(string s)
```

```
Board::setPlayerName(s,0)
Board::setPlayerName("Bot",1)
```

Declare Variables
list<int> botPlays(6)
set<int> cells;

Board::gameOver()

Declare Variables
int choice

returnCurPlr()==0

```
Board::printBoard()
```

Prompt Input
"player[0]'s turn choose
a cell 1-6 LR on top row

```
Board::player1turn()
Board::setCurPlr(1)
```

```
Board::announceWin()
```

```
Board::printBoard()
```

return

Prompt Input
"player[1]'s turn choose
a cell 1-6 LR on bottom row

Declare Variables
list<int>::iterator it
it=find(botPlays.begin(), botPlays.end(), rand()%6+1
choice=it

```
Board::botPlay(choice)
Board::setCurPlr(0)
```

Board
private: map<string, int> plrScore list<int> plr1pits list<int> plr2pits queue<int> gameHist stack<int> curPlr string player[2]
Board() Board(string, string) void setCurPlr(int) int returnCurPlr() void setPlayerName() void playGame() string getPlayerName(int) void printBoard() void updateGame(int) void player1turn(int) void player1capture(int) void player2turn(int) void player2capture(int) void botPlay(int) bool gameOver() void announceWin()

SimGame
private: Board board
SimGame(string)