Esha Mahabir

**Methods**

Public:

SnakesandLadders();-class constructor that assigns reference variable and makes an output window to Console.

splashScreen () - displays an animation and description of code

mainMenu() - allows the user to choose how they want to continue

difficultyLevel() - allows the user to choose the difficulty of the game

players() - gets the names of the players

goodbye() - prints goodbye statement

instructions() - provides instructions of the game to the user

playHardGame() - allows the user to play the hard version of the game

playEasyGame() - allows the user to play the easy version of the game

highScores() - keeps track of the top ten players and stores them in a file

animation() - draws Animation from animation class

Private:

drawTitle()

pauseProgram()

Black Box return:

May need :

spacesMove(int die1Num, die2Num); - adds the two dice numbers and determines how many times the player has to move.

**Variables**

Global:

static Console c; -The output console

static final int TOTAL = 20;

int choice; - Saves the user input for main

String player1Name; - saves the first player's username

String player2Name; - saves the second player's userName

int difficulty; - saves the user input for difficulty level

String numstr; - used for errortraping

int clearFile; - saves the user input for wheater they want to save the file

int player1Moves; - saves the amount of moves player one makes

int player2Moves;- saves the amount of moves player two makes

int die1Num; - saves the first die number

int die2Num; - saves the second die number

int dieTotal; - saves the number of both dies added together

Boolean winCheck1; - saves whether player 1 won or not

Boolean winCheck2; - saves whether player 2 won or not

**Arrays**

String[] top10Names;

int[] top10Scores;

String[] top10Levels;

**Main Method**

public static void main (String[] args)

{

SnakesandLadders s = new SnakesandLadders ();

s.splashScreen ();

while (true)

{

s.mainMenu ();

if (s.choice == 2)

{

s.instructions ();

}

else if (s.choice == 3)

{

s.highScore ();

}

else if (s.choice == 4)

{

break;

}

else if (s.choice == 1)

{

s.players ();

s.difficultyLevel ();

if (s.difficulty == 1)

{

s.playEasyGame ();

}

else if (s.difficulty == 2)

{

s.playHardGame ();

}

}

}

s.goodBye ();

}

} // main method