



# SORRY!

Project 1

A classic family board game, coded in C++

Uriel Salazar

Computer Science 5

43952

Uriel Salazar

Dr. Lehr

Computer Science 5

May 28 2015

## Project 1: Sorry! v1.13

### Introduction

For my first project in C++, I decided to go with the popular family game *Sorry!* It's a turn-based pursuit game where one – four players use both strategy and luck to get their pawn to the finish line. I decided to code this game not just because I loved it as a kid, but because I could clearly see the overall architecture of the game as an opportunity to use almost all of the program logic we have learned in the semester.

### Development Summary

The program is a C++ Application. It was developed using the NetBeans 7.4 IDE with the Cygwin compiler. The development of the project consists of many conditions for the user. Although the game is a very simple concept, I wanted to make sure the program would not crash under any conditions. Input validation was paramount! There is currently only one working function, which checks to see if the user's input is a positive integer. This function is called many times throughout the program. An average game takes a while to complete, which is why I stressed the importance of input validation. I did not want any crashes after the players invested a large amount of time into a game, nor did I want a confused player to break the rules and make illegal plays.

Lines of Code	<b>1,216</b>
Comment Lines	<b>123</b>
Blank Lines (White Space)	<b>93</b>
Lines used for game board	<b>2,698</b>
Total lines of source file	<b>4,130</b>
Project Size	<b>207 KB</b>

```

Please enter the number of players: -5
ERROR: Invalid Input
Please input a nonnegative integer: five
ERROR: Invalid Input
Please input a nonnegative integer: 5

This game can only be played with 1 - 4 players

Please enter the number of players: 1
AI version

Player 1          Player 2
-----          -----
*                #

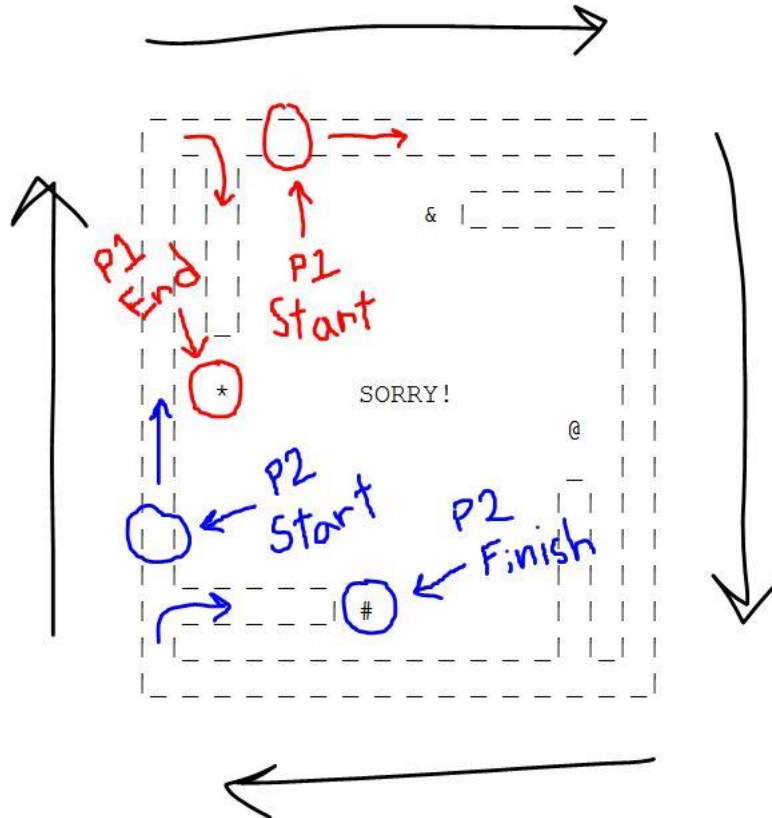
                PLAYER 1'S TURN:
Press the Enter key to draw a card! █

```

The heart of the program lies in a do-while loop, which contains two sets of nested if statements and switches. One set applies to the user, and the other applies to the AI. The first set is a bit more complex because it has much more conditions to prevent the user from inputting invalid choices. The game relies entirely on 11 possible, randomly generated cards, and each card has its own menu of possible actions and switch statements.

## Game Play and Rules

The rules for *Sorry!* are fairly simple: Get your pawn onto the game board (start), progress through the board, and exit through to the end point (finish).



Player 1 ( \* )

Player 2 ( # )

The Output on the left (The Game Board) is printed each time a player's pawn ( \* , # ) position has changed.

Note: Each player's pawn is displayed on a separate game board.

At the beginning of each turn, a player draws a card. There are 11 different types of cards:

“1”. – Start or 1 space forward

“2”. – Start or 2 spaces forward

“3”. – 3 spaces forward

“4”. – 4 spaces backward

“5”. – 5 spaces forward

“7”. – 7 spaces forward or move opponent 7 spaces backward

“8”. – 8 spaces forward

“10”. – 10 spaces forward or 1 step backward

“11”. – 11 spaces forward or a Switch places with an opposing pawn or skip turn

“12”. – 12 spaces forward

“Sorry!” – Remove the opponent's pawn off the game board

Note: There is no “6” and “9” cards.

## Game Play and Rules (cont.)

Cards “7”, “10”, and “11” offer choices to move in the opposite direction, as well as moving the opponent. This is where strategy comes into play

### Key Rules:

1. **One Card per Turn** – Each player can draw only one card per turn.
2. **One or Two to Start** – You must draw a “1” or a “2” in order to put your pawn on the game board and begin progressing through the board. If you draw anything else while your pawn is not on the board, you must skip your turn! (You must repeat this process if you are ejected from the game)
3. **Play the Card** – You must select an action when prompted. You can only skip a turn if no other legal move is available.

### Key Rules for version 2.0:

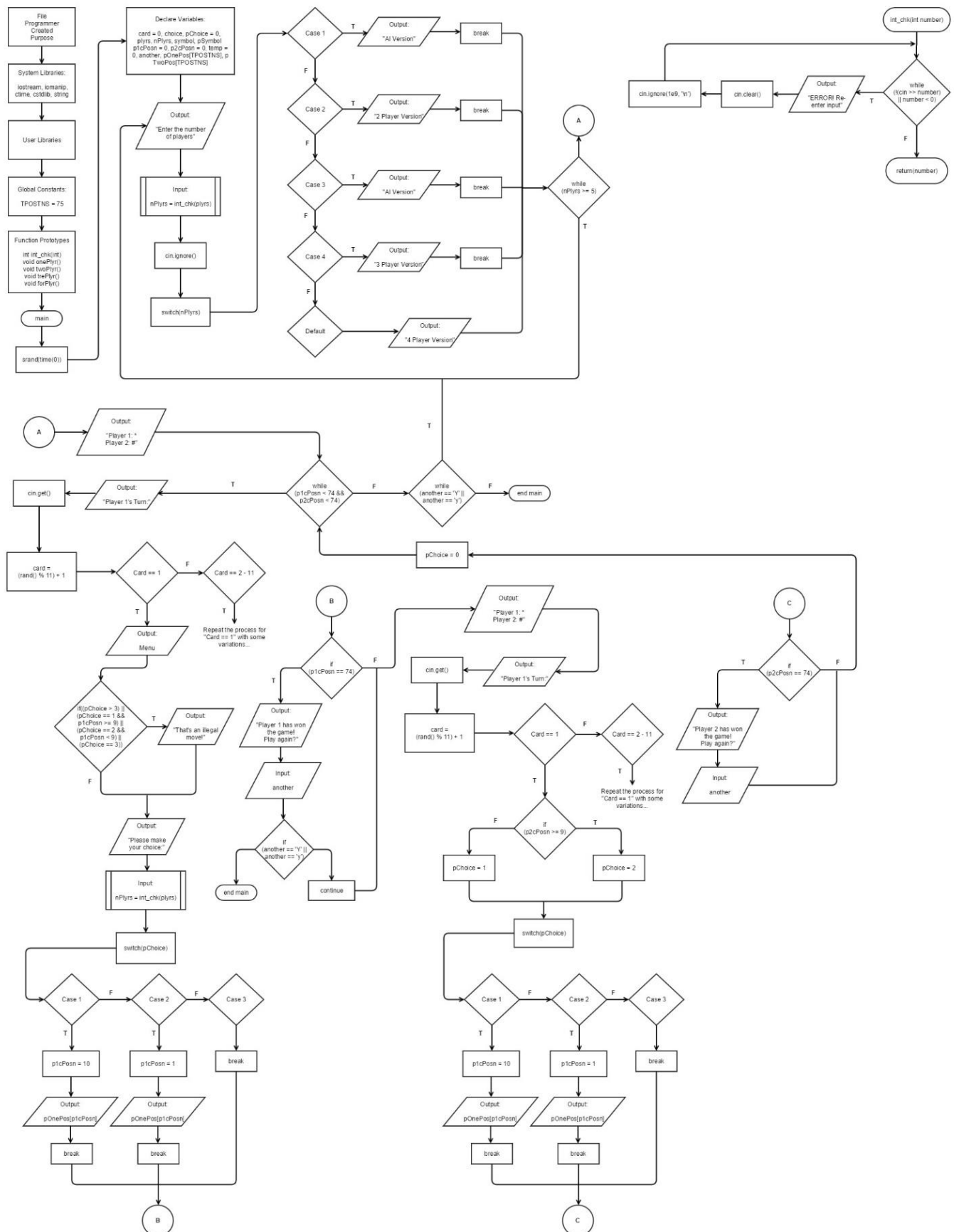
1. **Sorry! Bump** – When one player’s pawn lands on the same space as an opponent’s pawn, the opponent’s pawn gets removed from the game board (Optional: the player who overtakes the spot can tell the other player, “Sorry!”)
2. **Safe Zone** – When a player reaches their last stretch of spaces (space number 69 – 73, the outward rectangle that goes inside the board), they can no longer be removed from the game, under any circumstances.
3. **Exact Count** – Once in the safe zone, the player must draw an exact number of spaces for their pawn to exit the game board. Example. A pawn on space 71 must draw a 3! If they draw a 5, they will go back down the safe zone, to space 70. (Note: a pawn doesn’t leave the safe zone)

### Key Features for version 2.0:

1. **Four Pawns** – Each player will have four pawns instead of two.
2. **Multiplayer** – Add support for 2 – 4 players
3. **Slides** – Certain spaces on the board will allow a specific pawn to slide across a given number of spaces.
4. **Updated “7” and “Sorry!”** – An updated version of the “7” and “Sorry!” cards.
5. **One Game Board** – All pawns will be displayed on the same game board, instead of each pawn having a separate one.

## FlowChart

## Project 1 - Sorry!



## Pseudocode

Initialize

Do

Do

Print "Please enter the number of players"

Input number of players

Switch number of players

Case one

Print "AI Version"

Case two

Print "2 Player version"

Case three

"Print "3 Player version"

Case four

"Print "4 Player version"

Default

"This game can only be played with 1 – 4 players"

While number of players is greater than or equal to five

While player 1's current position is less than 74 and player 2's current position is less than 74

Set the card to a random number from one to eleven

If card has value one

Print menu

Do

If player choice is greater than three, or player choice is equal to one and player one's current position is greater than or equal to nine, or player choice is equal to two and player one's current position is less than nine, or player choice is equal to three

Print "that's an illegal move!"

Print "which choice would you like to make?"

Input player choice

While player choice is greater than three, or player choice is equal to one and player one's current position is greater than or equal to nine, or player choice is equal to two and player one's current position is less than nine, or player choice is equal to three  
Switch player choice

Case one

Set player one's current position to ten

Print player one array with player one's position value

Case two

Increment player one's current position by one

Print player one array with player one's position value

Case three

Exit switch statement

Repeat for cards two through eleven

If player one's current position is equal to 74

Print "Player one has won the game!"

Print "Play again?"

Input yes or no

If yes

Exit the do-while loop

If no

**Pseudocode (cont.)**

*Exit the program*

*Print "Player two's turn"*

*Generate a random card from one to eleven*

*If card has value one*

*Print "player 2 drew a 1"*

*If player two's current position is greater than or equal to nine*

*Set player's choice equal to two*

*Else*

*Set player choice equal to one*

*Switch player choice*

*Case one*

*Print "player 2 chose to start"*

*Set player two's current position to ten*

*Print player two array with value of player two's current position*

*Case two*

*Print "player 2 chose to move one space forward"*

*Increment player two's current position by one*

*Print player two array with value of player two's current position*

*Repeat for cards two through eleven*

*If player two's current position is greater than 74*

*Print "Player two has won the game!"*

*Print "Play again?"*

*Input yes or no*

*Set player's choice to zero*

*While another equals yes*

*Print "Good-bye!"*

*Exit the program*

*Input validation function*

*Input number*

*While number is not an unsigned short or less than zero*

*Print "Error! Input must be a nonnegative integer"*

*Return number*

**C++ Concepts**

Chapter	Concept	Example	Line
2	Comments	//Print players and their symbols	2805
	The "cout" Object	cout << "\nPlease enter the number of players: ";	2780
	Variables & Literals	p1cPosn = 74;	3030
	Data Types	unsigned short card = 0;	37
	Arithmetic Operators	p2cPosn = 15 - (60 - temp);	3272
	Scope	int main()	31
		unsigned short plyrs, nPlyrs;	39
3	The "cin" Object	cin >> another;	4100
	Mathematical Experssions	p1cPosn = 60 + temp - 15;	3917
	Formatting Ouput	cout << setw(25) << "PLAYER 1'S TURN:";	2815
	Characters & String Objects	char another;	44
		string pTwoPos[TPOSTNS] =	1397
	Library Functions	cin.get();	2817
4	"if" Statements	if(card == 1)	2825
	"if-else" Statements	else if(card == 2)	2861
	Nested "if" Statements	if(card == 1)	3483
		if(p2cPosn >= 9)	3488
	"if-else-if" Statements	else if(card == 9)	3231
		if(p1cPosn > 74)	3255
	Logical Operators	if((pChoice > 2)    (pChoice == 1 && p1cPosn < 9))	3946
	Menus	cout << "1. Move 3 spaces forward" << endl;	2905
	Validating User Input	while(!(cin >> number)    number < 0)	4121
	"switch" Statements	switch(pChoice)	3814
Blocks	else if(card == 10) {	4018	
	}	4048	
5	Increment / Decrement Operators	p1cPosn += 11;	3254
	"while" Loops	while(p1cPosn < 74 && p2cPosn < 74)	2812
	"do while" loops	do	2777
		} while(nPlyrs >= 5);	2802
	Nested Loops	do	2774
		while(p1cPosn < 74 && p2cPosn < 74)	2812
6	Defining & Calling Functions	int int_chk(int number)	4118
	Function Protoypes	int int_chk(int);	24
	Sending Data	pChoice = int_chk(choice);	2841
	Passing Data	return(number);	4929
	"return" statements	return(number);	4929
7	Accessing Arrays	cout << pOnePos[p1cPosn];	3181
	Array Initialization	string pOnePos[TPOSTNS] =	45



## Major Variables

Type	Variable Name	Description	Line
const unsigned short	TPOSTNS	the total number of positions a pawn can have during a game	21
unsigned short	card	a random number from 1 to 11, simulates a deck of cards	37
unsigned short	choice	the users input after they are prompted with a cards menu	38
unsigned short	pChoice	the users input for a menu, after it has been validated to be a positive integer	38
unsigned short	plyrs	the user's input regarding the number of players	39
unsigned short	nPlyrs	the number of players, after it has been validated to be a positive integer	39
unsigned short	symbol	the user's choice of symbol in menu format	40
unsigned short	pSymbol	the user's choice of symbol in menu format, after it has been validated	40
unsigned short	p1cPosn	the value of player 1's current position	41
unsigned short	p2cPosn	the value of player 2's current position	42
unsigned short	temp	holds player 1 or player 2's temporary position value during a switch of positions	43
char	another	"Yes" or "No" input from the user, determines whether or not to loop the program	44
string	pOnePos	an array that holds all the possible game boards that contain player 1's symbol	45
string	pTwoPos	an array that holds all the possible game boards that contain player 2's symbol	1397
int	number	assigned to "pChoice" after it has been validated to be a nonnegative integer	4118

## References

Cover image - <http://www.sodahead.com/fun/favorite-card-game/question-3450103/?page=5>

Gaddis – Starting Out with C++

Savitch – Problem Solving with C++

<http://www.cplusplus.com/>

<http://stackoverflow.com/>



```

    " | | | | & | - - - - | | \n"
    " | | | | | | | | | | \n"
    " | | | | | | | | | | \n"
    " | | | | | | | | | | \n"
    " | | | | | | | | | | \n"
    " | | * SORRY! | | | | \n"
    " | | | @ | | | | \n"
    " | | | | | | | | | | \n"
    " | | | | | | | | | | \n"
    " | | | | | | | | | | \n"
    " | - - - - | # | | | | \n"
    " | | | | | | | | | | \n"
    " | - - - - | # | | | | \n"
    " | - - - - | # | | | | \n"
};

//Output game board
cout << setw(24) << " CSC5 Project 1 " << endl;
cout << setw(24) << "by Uriel Salazar" << endl;
cout << setw(24) << "-----" << endl;
cout << "
| - - - - - - - - - - | \n
| | | | | | | | | | \n
| | | | | | | | | | \n
| | | | | | | | | | \n
| | | | | | | | | | \n
| | | | | | | | | | \n
| | * SORRY! | | | | \n
| | | @ | | | | \n
| | | | | | | | | | \n
| | | | | | | | | | \n
| | | | | | | | | | \n
| | - - - - | # | | | | \n
| | | | | | | | | | \n
| - - - - - - - - - - | \n";

//Loop the entire game
do
{
    //Prompt user for input
    do
    {
        //Prompt user for number of players
        cout << "\nPlease enter the number of players: ";
        nPlyrs = int_chk(plyrs);
        cin.ignore();

        //Determine number of players
        switch (nPlyrs)
        {
            case 1:
                cout << "AI version" << endl;
                break;
            case 2:
                cout << "2 player version" << endl;
                break;
            case 3:
                cout << "3 player version" << endl;
                break;
            case 4:
                cout << "4 player version" << endl;
                break;
            default:
                cout << "\nThis game can only be played with 1 - 4 players" << endl;
        }
    } while(nPlyrs >= 5); //End do-while loop
    cout << endl;
}

```

```

//Print players and their symbols
cout << "Player 1 \t Player 2" << endl;
cout << "----- \t -----" << endl;
cout << "      *      \t      #      " << endl;

//Begin the game
//Loop until a player exceeds 74
while(plcPosn < 74 && p2cPosn < 74)
{
    cout << endl;
    cout << setw(25) << "PLAYER 1'S TURN:";
    cout << "\nPress the Enter key to draw a card! ";
    cin.get();

    //generate a random card
    card = (rand() % 11) + 1;
    cout << endl;

    //Determine the card value and apply the rules
    //Player 1 draws a "1"
    if(card == 1)
    {
        //Print the card's specific menu
        cout << "You drew a 1!" << endl;
        cout << "1. Start" << endl;
        cout << "2. Move 1 space forward" << endl;
        cout << "3. Skip turn" << endl;

        //Check for legal move
        do
        {
            if((pChoice > 3) || (pChoice == 1 && plcPosn >= 9) || (pChoice == 2 &&
plcPosn < 9) || (pChoice == 3))
            {
                cout << "That's an illegal move!";
            }
            cout << "\nWhich choice would you like to make? ";
            pChoice = int_chk(choice);
        }while((pChoice > 3) || (pChoice == 1 && plcPosn >= 9) || (pChoice == 2 &&
plcPosn < 9) || (pChoice == 3));

        //Execute the desired action
        switch(pChoice)
        {
            case 1:
                plcPosn = 10;
                cout << pOnePos[plcPosn];    //Output modified game board
                break;
            case 2:
                plcPosn += 1;
                cout << pOnePos[plcPosn];    //Output modified game board
                break;
            case 3:
                break;
        }
    }

    //Player 1 draws a "2"
    else if(card == 2)
    {
        //Print the card's specific menu
        cout << "You drew a 2!" << endl;
        cout << "1. Start" << endl;
        cout << "2. Move 2 spaces forward" << endl;
        cout << "3. Skip turn" << endl;

        //Check for legal move
        do
        {
            if((pChoice > 3) || (pChoice == 1 && plcPosn >= 9) || (pChoice == 1 &&
plcPosn >= 9) || (pChoice == 3))

```

```

        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    }while((pChoice > 3) || (pChoice == 1 && plcPosn >= 9) || (pChoice == 1 &&
plcPosn >= 9) || (pChoice == 3));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn = 10;
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            plcPosn += 2;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 3:
            break;
    }
}

//Player 1 draws a "3"
else if(card == 3)
{
    //Print the card's specific menu
    cout << "You drew a 3!" << endl;
    cout << "1. Move 3 spaces forward" << endl;
    cout << "2. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
>= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    } while((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 3;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            break;
    }
}

//Player 1 draws a "4"
else if(card == 4)
{
    //Print the card's specific menu
    cout << "You drew a 4!" << endl;
    cout << "1. Move 4 spaces backward" << endl;

```

```

cout << "2. Skip turn" << endl;

//Check for legal move
do
{
    if((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
>= 9))
    {
        cout << "That's an illegal move!";
    }
    cout << "\nWhich choice would you like to make? ";
    pChoice = int_chk(choice);
} while((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
plcPosn >= 9));

//Execute the desired action
switch(pChoice)
{
    case 1:
        plcPosn -= 4;
        cout << pOnePos[plcPosn];    //Output modified game board

        //Determine if Player 1 has moved before their safe zone (P1-0 - P1-8)
        if(plcPosn == 8)
        {
            plcPosn = 68;
        }
        else if(plcPosn == 7)
        {
            plcPosn = 67;
        }
        else if(plcPosn == 6)
        {
            plcPosn = 66;
        }
        else if(plcPosn == 5)
        {
            plcPosn = 65;
        }
        else if(plcPosn == 4)
        {
            plcPosn = 64;
        }
        else if(plcPosn == 3)
        {
            plcPosn = 63;
        }
        else if(plcPosn == 2)
        {
            plcPosn = 62;
        }
        else if(plcPosn == 1)
        {
            plcPosn = 61;
        }
        else if(plcPosn == 0)
        {
            plcPosn = 60;
        }
        break;
    case 2:
        break;
}

//Player 1 draws a "5"
else if(card == 5)
{
    //Print the card's specific menu
    cout << "You drew a 5!" << endl;
    cout << "1. Move 5 spaces forward" << endl;

```

```

    cout << "2. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
>= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    } while((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 5;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            break;
    }
}

//Player 1 draws a "7"
else if(card == 6)
{
    //Print the card's specific menu
    cout << "You drew a 7!" << endl;
    cout << "1. Move 7 spaces forward" << endl;
    cout << "2. Move the opponent 7 spaces backward" << endl;
    cout << "3. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 3) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && p2cPosn
< 9) || (pChoice == 3 && plcPosn >= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    } while((pChoice > 3) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
p2cPosn < 9) || (pChoice == 3 && plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 7;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            p2cPosn -= 7;
            cout << pTwoPos[p2cPosn];    //Output modified game board

            //Determine if Player 2 has moved before their safe zone (P1-0 - P1-8)
            if(p2cPosn == 8)
            {

```



```

        p2cPosn = 68;
    }
    else if(p2cPosn == 7)
    {
        p2cPosn = 67;
    }
    else if(p2cPosn == 6)
    {
        p2cPosn = 66;
    }
    else if(p2cPosn == 5)
    {
        p2cPosn = 65;
    }
    else if(p2cPosn == 4)
    {
        p2cPosn = 64;
    }
    else if(p2cPosn == 3)
    {
        p2cPosn = 63;
    }
    else if(p2cPosn == 2)
    {
        p2cPosn = 62;
    }
    else if(p2cPosn == 1)
    {
        p2cPosn = 61;
    }
    else if(p2cPosn == 0)
    {
        p2cPosn = 60;
    }
    break;
case 3:
    break;
}

}

//Player 1 draws an "8"
else if(card == 7)
{
    //Print the card's specific menu
    cout << "You drew an 8!" << endl;
    cout << "1. Move 8 spaces forward" << endl;
    cout << "2. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
>= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    } while((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 8;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board

```

```

        break;
    case 2:
        break;
    }
}

//Player 1 draws a "10"
else if(card == 8)
{
    //Print the card's specific menu
    cout << "You drew a 10!" << endl;
    cout << "1. Move 10 spaces forward" << endl;
    cout << "2. Move 1 space backward" << endl;
    cout << "3. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 3) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
< 9) || (pChoice == 3 && plcPosn >= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    }while((pChoice > 3) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
< 9) || (pChoice == 3 && plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 10;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            plcPosn -= 1;
            cout << pOnePos[plcPosn];    //Output modified game board

            //Determine if Player 1 has moved before their safe zone (P1-0 - P1-8)
            if(plcPosn == 8)
            {
                plcPosn = 68;
            }
            else if(plcPosn == 7)
            {
                plcPosn = 67;
            }
            else if(plcPosn == 6)
            {
                plcPosn = 66;
            }
            else if(plcPosn == 5)
            {
                plcPosn = 65;
            }
            else if(plcPosn == 4)
            {
                plcPosn = 64;
            }
            else if(plcPosn == 3)
            {
                plcPosn = 63;
            }
            else if(plcPosn == 2)
            {
                plcPosn = 62;
            }
        }
    }
}

```

```

    }
    else if(plcPosn == 1)
    {
        plcPosn = 61;
    }
    else if(plcPosn == 0)
    {
        plcPosn = 60;
    }
    break;
case 3:
    break;
}
}

//Player 1 draws an "11"
else if(card == 9)
{
    //Print the card's specific menu
    cout << "You drew an 11!" << endl;
    cout << "1. Move 11 spaces forward" << endl;
    cout << "2. Switch places with an opposing pawn" << endl;
    cout << "3. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 3) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
< 9) || (pChoice == 3 && plcPosn >= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    } while((pChoice > 3) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
plcPosn < 9) || (pChoice == 3 && plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 11;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            if(plcPosn >= 15 && p2cPosn >= 15)
            {
                temp = plcPosn;
                plcPosn = p2cPosn - 15;
                p2cPosn = temp + 15;
            }
            else if(plcPosn >= 15 && p2cPosn <= 15)
            {
                temp = plcPosn;
                plcPosn = 60 + p2cPosn - 15;
                p2cPosn = 15 - (60 - temp);
            }
            else if(plcPosn <= 15 && p2cPosn <= 15)
            {
                temp = plcPosn;
                plcPosn = 60 + p2cPosn - 15;
                p2cPosn = temp + 15;
            }
            else if(plcPosn <= 15 && p2cPosn >= 15)
            {
                plcPosn = p2cPosn - 15;
                p2cPosn = p2cPosn - (plcPosn - 10);
            }
        }
    }
}

```

```

}
cout << pOnePos[p1cPosn] << endl;
cout << endl;
cout << pTwoPos[p2cPosn] << endl;

//Determine if Player 1 has moved before their safe zone (P1-0 - P1-8)
if(p1cPosn == 8)
{
    p1cPosn = 68;
}
else if(p1cPosn == 7)
{
    p1cPosn = 67;
}
else if(p1cPosn == 6)
{
    p1cPosn = 66;
}
else if(p1cPosn == 5)
{
    p1cPosn = 65;
}
else if(p1cPosn == 4)
{
    p1cPosn = 64;
}
else if(p1cPosn == 3)
{
    p1cPosn = 63;
}
else if(p1cPosn == 2)
{
    p1cPosn = 62;
}
else if(p1cPosn == 1)
{
    p1cPosn = 61;
}
else if(p1cPosn == 0)
{
    p1cPosn = 60;
}

//Determine if Player 2 has moved before their safe zone (P2-0 - P2-8)
if(p2cPosn == 8)
{
    p2cPosn = 68;
}
else if(p2cPosn == 7)
{
    p2cPosn = 67;
}
else if(p2cPosn == 6)
{
    p2cPosn = 66;
}
else if(p2cPosn == 5)
{
    p2cPosn = 65;
}
else if(p2cPosn == 4)
{
    p2cPosn = 64;
}
else if(p2cPosn == 3)
{
    p2cPosn = 63;
}
else if(p2cPosn == 2)
{
    p2cPosn = 62;
}

```

```

    }
    else if(p2cPosn == 1)
    {
        p2cPosn = 61;
    }
    else if(p2cPosn == 0)
    {
        p2cPosn = 60;
    }
    break;
case 3:
    break;
}
}

//Player 1 draws a "12"
else if(card == 10)
{
    //Print the card's specific menu
    cout << "You drew a 12!" << endl;
    cout << "1. Move 12 spaces forward" << endl;
    cout << "2. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 && plcPosn
>= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    } while((pChoice > 2) || (pChoice == 1 && plcPosn < 9) || (pChoice == 2 &&
plcPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            plcPosn += 12;
            if(plcPosn > 74)
            {
                plcPosn = 74;
            }
            cout << pOnePos[plcPosn];    //Output modified game board
            break;
        case 2:
            break;
    }
}

//Player 1 draws a "Sorry!" card
else
{
    //Print the card's specific menu
    cout << "You drew a \"Sorry!\" Card!" << endl;
    cout << "1. Move an opponent's pawn off the game board!" << endl;
    cout << "2. Skip turn" << endl;

    //Check for legal move
    do
    {
        if((pChoice > 2) || (pChoice == 1 && p2cPosn < 9) || (pChoice == 2 && p2cPosn
>= 9))
        {
            cout << "That's an illegal move!";
        }
        cout << "\nWhich choice would you like to make? ";
        pChoice = int_chk(choice);
    }
}

```

```

    } while((pChoice > 2) || (pChoice == 1 && p2cPosn < 9) || (pChoice == 2 &&
p2cPosn >= 9));

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            p2cPosn = 0;
            cout << "Player 2 has been moved off the game board!" << endl;
            cout << "
                _ _ _ _ _ _ _ _ _ _ \n"
                "| | | | | | | | | | \n"
                "| | | | |      & | | | | | \n"
                "| | | | |      | | | | | \n"
                "| | | | |      | | | | | \n"
                "| | | | |      | | | | | \n"
                "| | *          SORRY!       | | \n"
                "| |              @           | | \n"
                "| |              _           | | \n"
                "| |              | | | | | \n"
                "| | _ _ _ _ _ | #           | | \n"
                "| | _ _ _ _ _ | | | | | \n"
                "_ _ _ _ _ _ _ _ _ _ \n";

            break;
        case 2:
            break;
    }
}

//Determine whether Player 1 has won the game
if(plcPosn == 74)
{
    cout << "\nPlayer 1 has won the game!" << endl;
    cout << "\nWould you like to play another game? (Y/N) ";
    cin >> another;
    if(another == 'Y' || another == 'y')
    {
        continue;
    }
    else
    {
        cout << "\nGood-bye!" << endl;
        return 0;
    }
}

//Simulate the AI (Player 2)
cout << endl;
cout << setw(25) << "PLAYER 2'S TURN:";
cout << "\nPress the Enter key to continue... ";
cin.ignore();
cin.get();

//generate a random card
card = (rand() % 11) + 1;
cout << endl;

//Determine the card value and apply the rules
//Player 2 draws a "1"
if(card == 1)
{
    cout << "Player 2 drew a 1!" << endl;

    //Check for legal move
    if(p2cPosn >= 9)
    {
        pChoice = 2;
    }
}

```

```

else
{
    pChoice = 1;
}

//Execute the desired action
switch(pChoice)
{
    case 1:
        cout << "Player 2 chose to start." << endl;
        p2cPosn = 10;
        cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
        break;
    case 2:
        cout << "Player 2 had to move 1 space forward." << endl;
        p2cPosn += 1;
        cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
        break;
}
}

//Player 2 draws a "2"
else if(card == 2)
{
    cout << "Player 2 drew a 2!" << endl;

    //Check for legal move
    if(p2cPosn >= 9)
    {
        pChoice = 2;
    }
    else
    {
        pChoice = 1;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to start." << endl;
            p2cPosn = 10;
            cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
            break;
        case 2:
            cout << "Player 2 had to move 2 spaces forward." << endl;
            p2cPosn += 2;
            if(p2cPosn > 74)
            {
                p2cPosn = 74;
            }
            cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
            break;
    }
}

//Player 2 draws a "3"
else if(card == 3)
{
    cout << "Player 2 drew a 3!" << endl;

    //Check for legal move
    if(p2cPosn >= 9)
    {
        pChoice = 1;
    }
    else
    {
        pChoice = 2;
    }
}

```

```

//Execute the desired action
switch(pChoice)
{
    case 1:
        cout << "Player 2 chose to move 3 spaces forward." << endl;
        p2cPosn += 3;
        if(p2cPosn > 74)
        {
            p2cPosn = 74;
        }
        cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
        break;
    case 2:
        cout << "Player 2 had to skip their turn." << endl;
        break;
}
}

//Player 2 draws a "4"
else if(card == 4)
{
    cout << "Player 2 drew a 4!" << endl;

    //Check for legal move
    if(p2cPosn >= 9)
    {
        pChoice = 1;
    }
    else
    {
        pChoice = 2;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to move 4 spaces backward." << endl;
            p2cPosn -= 4;
            cout << pTwoPos[p2cPosn] << endl;    //Output modified game board

            //Determine if Player 2 has moved before their safe zone (P2-0 - P2-8)
            if(p2cPosn == 8)
            {
                p2cPosn = 68;
            }
            else if(p2cPosn == 7)
            {
                p2cPosn = 67;
            }
            else if(p2cPosn == 6)
            {
                p2cPosn = 66;
            }
            else if(p2cPosn == 5)
            {
                p2cPosn = 65;
            }
            else if(p2cPosn == 4)
            {
                p2cPosn = 64;
            }
            else if(p2cPosn == 3)
            {
                p2cPosn = 63;
            }
            else if(p2cPosn == 2)
            {
                p2cPosn = 62;
            }
            else if(p2cPosn == 1)

```



```

        {
            p2cPosn = 61;
        }
        else if(p2cPosn == 0)
        {
            p2cPosn = 60;
        }
        break;
    case 2:
        cout << "Player 2 had to skip their turn." << endl;
        break;
    }
}

//Player 2 draws a "5"
else if(card == 5)
{
    cout << "Player 2 drew a 5!" << endl;

    //Check for legal move
    if(p2cPosn >= 9)
    {
        pChoice = 1;
    }
    else
    {
        pChoice = 2;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to move 5 spaces forward." << endl;
            p2cPosn += 5;
            if(p2cPosn > 74)
            {
                p2cPosn = 74;
            }
            cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
            break;
        case 2:
            cout << "Player 2 had to skip their turn." << endl;
            break;
    }
}

//Player 2 draws a "7"
else if(card == 6)
{
    cout << "Player 2 drew a 7!" << endl;

    //Check for legal move
    if(p1cPosn < (p2cPosn - 15))
    {
        pChoice = 1;
    }
    else if(p1cPosn > (p2cPosn - 15) && p1cPosn < 9)
    {
        pChoice = 2;
    }
    else
    {
        pChoice = 3;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to move 7 spaces forward." << endl;

```

```

        p2cPosn += 7;
        if(p2cPosn > 74)
        {
            p2cPosn = 74;
        }
        cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
        break;
    case 2:
        cout << "Player 2 chose to move their opponent 7 spaces backward." << endl;

        p1cPosn -= 7;
        cout << pOnePos[p1cPosn] << endl;    //Output modified game board

        //Determine if Player 2 has moved before their safe zone (P2-0 - P2-8)
        if(p2cPosn == 8)
        {
            p2cPosn = 68;
        }
        else if(p2cPosn == 7)
        {
            p2cPosn = 67;
        }
        else if(p2cPosn == 6)
        {
            p2cPosn = 66;
        }
        else if(p2cPosn == 5)
        {
            p2cPosn = 65;
        }
        else if(p2cPosn == 4)
        {
            p2cPosn = 64;
        }
        else if(p2cPosn == 3)
        {
            p2cPosn = 63;
        }
        else if(p2cPosn == 2)
        {
            p2cPosn = 62;
        }
        else if(p2cPosn == 1)
        {
            p2cPosn = 61;
        }
        else if(p2cPosn == 0)
        {
            p2cPosn = 60;
        }
        break;
    case 3:
        cout << "Player 2 had to skip their turn." << endl;
        break;
    }
}

//Player 2 draws an "8"
else if(card == 7)
{
    cout << "Player 2 drew an 8!" << endl;

    //Check for legal move
    if(p2cPosn >= 9)
    {
        pChoice = 1;
    }
    else
    {
        pChoice = 2;
    }
}

```

```

//Execute the desired action
switch(pChoice)
{
    case 1:
        cout << "Player 2 chose to move 8 spaces forward." << endl;
        p2cPosn += 8;
        if(p2cPosn > 74)
        {
            p2cPosn = 74;
        }
        cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
        break;
    case 2:
        cout << "Player 2 had to skip their turn." << endl;
        break;
}
}

//Player 2 draws a "10"
else if(card == 8)
{
    cout << "Player 2 drew a 10!" << endl;

    //Check for legal move
    if(p2cPosn < 9)
    {
        pChoice = 3;
    }
    else if(p2cPosn == 9 || p2cPosn == 10)
    {
        pChoice = 2;
    }
    else
    {
        pChoice = 3;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to move 10 spaces forward." << endl;
            p2cPosn += 10;
            if(p2cPosn > 74)
            {
                p2cPosn = 74;
            }
            cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
            break;
        case 2:
            cout << "Player 2 chose to move 1 space backwards." << endl;
            p2cPosn -= 1;
            cout << pTwoPos[p2cPosn];    //Output modified game board

            //Determine if Player 2 has moved before their safe zone (P2-0 - P2-8)
            if(p2cPosn == 8)
            {
                p2cPosn = 68;
            }
            else if(p2cPosn == 7)
            {
                p2cPosn = 67;
            }
            else if(p2cPosn == 6)
            {
                p2cPosn = 66;
            }
            else if(p2cPosn == 5)
            {
                p2cPosn = 65;
            }
        }
    }
}

```

```

        else if(p2cPosn == 4)
        {
            p2cPosn = 64;
        }
        else if(p2cPosn == 3)
        {
            p2cPosn = 63;
        }
        else if(p2cPosn == 2)
        {
            p2cPosn = 62;
        }
        else if(p2cPosn == 1)
        {
            p2cPosn = 61;
        }
        else if(p2cPosn == 0)
        {
            p2cPosn = 60;
        }
        break;
    case 3:
        cout << "Player 2 had to skip their turn." << endl;
        break;
    }
}

//Player 2 draws an "11"
else if(card == 9)
{
    cout << "Player 2 drew an 11!" << endl;

    //Check for legal move
    if(p2cPosn >= 1 && p2cPosn < plcPosn)
    {
        pChoice = 2;
    }
    else if(p2cPosn >= 10)
    {
        pChoice = 1;
    }
    else
    {
        pChoice = 3;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to move 11 spaces forwards." << endl;
            p2cPosn += 11;
            if(p2cPosn > 74)
            {
                p2cPosn = 74;
            }
            cout << pTwoPos[p2cPosn];    //Output modified game board
            break;
        case 2:
            cout << "Player 2 chose to switch places with Player 1." << endl;
            if(p2cPosn >= 15 && plcPosn >= 15)
            {
                temp = p2cPosn;
                p2cPosn = plcPosn + 15;
                plcPosn = temp - 15;
            }
            else if(plcPosn >= 15 && p2cPosn <= 15)
            {
                temp = p2cPosn;
                p2cPosn = 15 - (60 - plcPosn);
                plcPosn = 60 + temp - 15;
            }
        }
    }
}

```

```

}
else if(plcPosn <= 15 && p2cPosn <= 15)
{
    temp = p2cPosn;
    plcPosn = plcPosn + 15;
    p2cPosn = 60 - (15 - p2cPosn);
}
else if(plcPosn <= 15 && p2cPosn >= 15)
{
    temp = p2cPosn;
    p2cPosn = plcPosn + 15;
    plcPosn = 60 + (temp - 15);
}
cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
cout << endl;
cout << pOnePos[plcPosn] << endl;    //Output modified game board

//Determine if Player 1 has moved before their safe zone (P1-0 - P1-8)
if(plcPosn == 8)
{
    plcPosn = 68;
}
else if(plcPosn == 7)
{
    plcPosn = 67;
}
else if(plcPosn == 6)
{
    plcPosn = 66;
}
else if(plcPosn == 5)
{
    plcPosn = 65;
}
else if(plcPosn == 4)
{
    plcPosn = 64;
}
else if(plcPosn == 3)
{
    plcPosn = 63;
}
else if(plcPosn == 2)
{
    plcPosn = 62;
}
else if(plcPosn == 1)
{
    plcPosn = 61;
}
else if(plcPosn == 0)
{
    plcPosn = 60;
}

//Determine if Player 2 has moved before their safe zone (P2-0 - P2-8)
if(p2cPosn == 8)
{
    p2cPosn = 68;
}
else if(p2cPosn == 7)
{
    p2cPosn = 67;
}
else if(p2cPosn == 6)
{
    p2cPosn = 66;
}
else if(p2cPosn == 5)
{
    p2cPosn = 65;
}

```

```

    }
    else if(p2cPosn == 4)
    {
        p2cPosn = 64;
    }
    else if(p2cPosn == 3)
    {
        p2cPosn = 63;
    }
    else if(p2cPosn == 2)
    {
        p2cPosn = 62;
    }
    else if(p2cPosn == 1)
    {
        p2cPosn = 61;
    }
    else if(p2cPosn == 0)
    {
        p2cPosn = 60;
    }
    break;
case 3:
    cout << "Player 2 had to skip their turn." << endl;
    break;
}

//Player 2 draws a "12"
else if(card == 10)
{
    cout << "Player 2 drew a 12!" << endl;

    //Check for legal move
    if(p2cPosn >= 10)
    {
        pChoice = 1;
    }
    else
    {
        pChoice = 2;
    }

    //Execute the desired action
    switch(pChoice)
    {
        case 1:
            cout << "Player 2 chose to move 12 spaces forward." << endl;
            p2cPosn += 12;
            if(p2cPosn > 74)
            {
                p2cPosn = 74;
            }
            cout << pTwoPos[p2cPosn] << endl;    //Output modified game board
            break;
        case 2:
            cout << "Player 2 had to skip their turn." << endl;
            break;
    }
}

//Player 2 draws a "Sorry!" card
else
{
    cout << "Player 2 drew a \"Sorry!\" card!" << endl;

    //Check for legal move
    if(p1cPosn >= 1)
    {
        pChoice = 1;
    }
}

```

```

else
{
    pChoice = 2;
}

//Execute the desired action
switch(pChoice)
{
    case 1:
        plcPosn = 0;
        cout << "Player 1 has been moved off the game board!" << endl;
        cout << "
            _ _ _ _ _ _ _ _ \n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_&|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_*_|_|_|SORRY!_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n"
            "|_|_|_|_|_|_|_|_|_|_|_\n";

        break;
    case 2:
        cout << "Player 2 had to skip their turn." << endl;
        break;
}

//Determine whether player 2 has won the game
if(p2cPosn >= 74)
{
    cout << "\nPlayer 2 has won the game!" << endl;
    cout << "\nWould you like to play another game? (Y/N) ";
    cin >> another;
}

//Reset the player's choice
pChoice = 0;

//Exit while loop
}

//Exit or repeat the do-while loop
} while(another == 'Y' || another == 'y');

//Exit the program
cout << "\nGood-bye!" << endl;
return 0;
}

//Input validation function
int int_chk(int number)
{
    //Check for input
    while(!(cin >> number) || number < 0)
    {
        cout << "ERROR: Invalid Input\n"
                "Please input a nonnegative integer: ";
        cin.clear();
        cin.ignore(1e9, '\n');
    }
    //Return choice after it has been validated
    return(number);
}

```