

# **Project 2**

**Chuck A Luck**

By: Justin Gutierrez

CIS-5

07/31/2022

## About the Game:

Game Title: Chuck a Luck

The game Chuck a Luck is a fairly popular game that is played in many casinos and has fairly simple logic on how to play it. In Chuck a Luck there are dices that are held by the dealer and usually there is a board or layout of the numbers one to six which are the numbers represented on each dice. The game can allow for many players and a variety of rules can be set by the dealer. So all participants of the game must bet money a number that lies between 1 and 6. The dealer will then roll the 3 dice and if any of the dice land on one of the numbers that was betted on, the person who betted on that number will receive a payout equivalent to their original bet along with their original bet. Though if two dice land on the same number that was betted on the payout is increased to 2 times the original bet along with their original bet. Lastly, if all three dice land on the same number that was betted on then the person who betted on that number will receive a payout equivalent to three times their original bet along with their original bet. If none of the dice land a number that was betted on then that participant loses and their money goes to the dealer.

## Summary:

My project contains about 425 lines of code. My game has become more refined and an overall better experience since project 1. I was able to produce a gameboard to go along with the game. This game board is used in the actual game itself and it was fun and efficient to produce using arrays and vectors. Furthermore, using functions I was able to build off the single, double, and triple payouts that my first project had. Now instead of just declaring if you won by single, double, or triple it will accurately calculate your payout based on your bet. Betting is also new to

my game as it was not included in project 1. You can now bet as much as you want on any number and the game will tell you whether you win - and by what method - or lose.

## **Development of Game - Four Versions**

I developed my game once again over four versions vastly improving from project one. The first thing I wanted to implement in my game was the betting function as it is arguably the most important component of the game. No doubt this was also the most difficult part of the game to form as well. Since winning has many scenarios that fall under a single, double, and a triple there were many obstacles to meet in order to get the accurate payout. I used two boolean functions in order to tell when the game was won by double, triple, or single. A triple was assigned a value of 3, a double 2, and a single 1. One function tested if the value of 3 was divisible by 2 if it was it would return false, if not it would return true. The false in this instance represented a single, and the true a triple. Another function tested if the value of 2 was divisible by 2 if it was it would now return true, if not it would return false. Then based on the return of those boolean functions, I used two functions to accurately calculate the winnings for a double or triple depending on how the player won. If both boolean functions returned false it could be determined that the player won via single. Lastly, if the player lost and losses incremented then my code would instruct the player to give the money to the dealer. In version 2, I created the game board using arrays and vectors. This took 4 functions. 2 functions to form the array and vector and 2 functions to output the array and vector. In version 3, I used more functions to create the payout for each scenario. In version 1 the winnings were calculated, though winnings are different from payout. The payout was the original bet plus the winnings. I used functions to add the bet back into the winnings in order to get the accurate payout depending on the scenario. Lastly, in version 4 I used a 2D Array in order to create a little scoreboard that would keep track of wins, losses, and games played. I

definitely had a lot of fun making this game and having to overcome many errors and challenges to make.

```
Game board
1      2      3
4      5      6

Please enter a number between 1-6
4

Please place the amount of money that you would like to bet on number 4
$20

Dice 1 = 4
Dice 2 = 3
Dice 3 = 2

You win
You won $20

You have won a payout of $40

You have lost 1 more games than you have won
You have played 3 games
You have won 1 games
You have lost 2 games

You have won 33% of games
You have lost 67% of games

Games Played Games Won Games Lost
3           1           2

Would you like to play again?
Enter Y for yes and any other letter to stop
```

**(Game won by a single^^)**

```

Enter Y for yes and any other letter to stop
y

      Game board
      1      2      3
      4      5      6

Please enter a number between 1-6
4

Please place the amount of money that you would like to bet on number 4
$20

Dice 1 = 4
Dice 2 = 5
Dice 3 = 4

You win by double! - double the payout
You won $40

You have won a payout of $60

You have lost 1 more games than you have won
You have played 13 games
You have won 6 games
You have lost 7 games

You have won 46% of games
You have lost 54% of games

      Games Played   Games Won   Games Lost
          13           6           7

Would you like to play again?
Enter Y for yes and any other letter to stop

```

**(Game won by a double^^)**

```

      Game board
      1      2      3
      4      5      6

Please enter a number between 1-6
2

Please place the amount of money that you would like to bet on number 2
$20

Dice 1 = 1
Dice 2 = 5
Dice 3 = 3
You Lose
Pay the dealer $20
You have lost 1 more games than you have won

You have played 1 games
You have won 0 games
You have lost 1 games

You have won 0% of games
You have lost 100% of games

Would you like to play again?
Enter Y for yes and any other letter to stop
k

Thanks for playing!

RUN FINISHED; exit value 0; real time: 5s; user: 0ms; system: 0ms
█
```

**(Game lost, end program^^)**

## **Pseudocode:**

**If** start is entered

Output Game board

**Do-loop**

Prompt input for a number to bet on 1-6

Accept a value 1-6

Output error if value is less than 1 or greater than 6

Prompt input for a bet that will be placed on the number selected

A random number is selected for die 1

Random Number is divided by 6 plus 1(value must be between 1-6)

**Switch**

If number is 1 it goes to case 1, number is 2 it goes to case 2, and so on

A random number is selected for die 2

Random Number is divided by 6 plus 1(value must be between 1-6)

**Switch (#2)**

If number is 1 it goes to case 1, number is 2 it goes to case 2, and so on

A random number is selected for die 3

Random Number is divided by 6 plus 1(value must be between 1-6)

**Switch (#3)**

If number is 1 it goes to case 1, number is 2 it goes to case 2, and so on

**If**

If case 1 was incremented in the switch for die 1 than the value is 1,

If case 2 was incremented in the switch for die 1 than the value is 2,

If case 3 was incremented in the switch for die 1 than the value is 3,

Repeats until case 6 which equals 6

**If**

If case 1 was incremented in the switch for die 2 than the value is 1,

If case 2 was incremented in the switch for die 2 than the value is 2,

If case 3 was incremented in the switch for die 2 than the value is 3,

Repeats until case 6 which equals 6

**If**

If case 1 was incremented in the switch for die 3 than the value is 1,

If case 2 was incremented in the switch for die 3 than the value is 2,

If case 3 was incremented in the switch for die 3 than the value is 3,  
Repeats until case 6 which equals 6

**If**

If the number that was inputted by the player equals the value of all 3 dice it is a triple  
Wins gets incremented by 1  
Bet is multiplied by 3 to give winnings of 3:1

**If**

If the number that was inputted by the player equals the value of 2 of the 3 dice it is a  
double  
Wins gets incremented by 1  
Bet is multiplied by 2 to give winnings of 2:1

**If**

If the number that was inputted by the player equals the value 1 of the dice it is a single  
Wins gets incremented by 1  
Bet is multiplied by 1 to give winnings of 1:1

Percentage of games won= games/wins  
Percentage of games lost= games/losses

**Output**

Winnings/Lost money  
\*Total payout (if won)  
You lose or you win  
Amount of games played  
Amount of wins  
Amount of losses

Scoreboard with wins and losses.

Prompt for input to play again  
While (Y or y is entered to play again)  
Any other letter ends game  
Output  
Thanks for playing

**If**

If exit was entered at main screen  
Game ends  
Output Thanks for playing



# Program

```
//System Libraries
#include <iostream>//Input-Output Library
#include <iomanip>//Format
#include <cmath>//Math Functions
#include <cstdlib>//Srand for the random number generator
#include <fstream>//File input and output
#include <string>//Strings
#include <ctime>//Time for rand
#include <vector>
using namespace std;

//User Libraries

//Global Constants
//Mathematical/Physics/Conversions, Higher dimensioned arrays
const float PERCENT=100.0f;//Used to convert values to percentage
const int COLS=4;
//Function Prototypes
void term();
void single(int);
bool isDbl(int);
bool isTpl(int);
void dpay(int &, int dbl=2);
void tpay(int &, int tpl=3);
void view(vector <int>);
void model (vector <int>);
void viewA(int []);
void modelA (int []);
void single(int);
//Execution Begins Here
int main(int argc, char** argv) {
    //Initialize the Random Number Seed
    srand(static_cast<unsigned int>(time(0)));//Random number generator
    //Declare Variables
    ifstream input;      //File input
    string Name;
    input.open("TitleProj.txt");    //Title of file being opened
    input>>Name;
    ofstream output;      //File output
    output.open("GameStats.txt");    //Title of file being outputted
```

```

unsigned char die1,die2,die3;          //Game requires 3 die
unsigned int c1s,c2s,c3s,c4s,c5s,c6s;  //All 6 possible values
unsigned int a1s,a2s,a3s,a4s,a5s,a6s;  //for all 3 dice
unsigned int b1s,b2s,b3s,b4s,b5s,b6s;

char again;                            //Used to decide to play again
string start;                           //starts the game/program
int num;                                //Allows player to select #
float games, wins, lose;                 //Determines games played, won, lost
float percwin,percL;                     //Percentage of games won & lost
bool test;                              //Used to increment wins or loss
bool WL;                                //Determines win vs loss
const int SIZE=5;
vector <int> firs (SIZE);
firs[0]=1;
firs[1]=2;
firs[2]=3;
int sec[SIZE];
sec[0]=4;
sec[1]=5;
sec[2]=6;
float lead[SIZE][COLS];
//Initialize Variables
test=true;                              //Automatically set to true
games=0;                                //0 games, 0 wins, 0 losses to start
wins=0;
lose=0;

cout<<endl;
cout<<setw(40)<<Name<<setw(40)<<endl;
cout<<endl;
cout<<setw(30)<<"Enter Start to begin the game"<<setw(40)
    <<"Enter Exit to close the game"<<endl;    //Allows to start or end game
cout<<endl;
cin>>start;

//Map inputs to outputs -> The Process
int bet;
int place;
if (start == "Start"|| start=="start"){    //Determines whether to start game
    cout<<endl;
    do{
        //Game reruns so long as player enters y after every game.
        cout<<endl;
        cout<<setw(20)<<"Game board"<<endl;

```

```

    cout<<endl;
    view(firs);
    cout<<endl;
    cout<<endl;
    viewA(sec);
    cout<<endl;
    cout<<endl;
    cout<<"Please enter a number between 1-6"<<endl;
        cout<<endl;           //Can only bet on #s 1-6
    cin>>num;
    if(num<1|| num>6){
        cout<<"Value is out of range! Please enter a number between 1-6"<<endl;
    cin>>num; }           //Input validation
    cout<<endl;
    cout<<"Please place the amount of money "
        "that you would like to bet on number "<<num<<endl;
    cout<<"$"; cin>>bet;
    cout<<endl;
int ogbet=bet;
int roll=1;           //Roll =1 at start of every game
a1s=a2s=a3s=a4s=a5s=a6s=0;           //All dice values =0
b1s=b2s=b3s=b4s=b5s=b6s=0;
c1s=c2s=c3s=c4s=c5s=c6s=0;

while(roll<=1){           //Will roll dice 1 since every loop resets value of the roll
    die1=rand()%6+1;//[1,6]           //Used to make all possible random #s 1-6

    switch(die1){
        case 1:a1s++;break;           //Used for value of 1
        case 2:a2s++;break;           //Dice value of 2
        case 3:a3s++;break;           //Dice value of 3
        case 4:a4s++;break;           //Dice value of 4
        case 5:a5s++;break;           //Dice value of 5
        case 6:a6s++;break;           //Dice value of 6
        default:cout<<"Bad Dice"<<endl;           //If something messes up
    }
    roll++;
}for(int roll=1;roll<=1;roll++){
    die2=rand()%6+1;//[1,6]

    switch(die2){
        case 1:b1s++;break;           //Dice 2 values of 1-6
        case 2:b2s++;break;           //Same as dice 1 but for dice 2

```

```

        case 3:b3s++;break;
        case 4:b4s++;break;
        case 5:b5s++;break;
        case 6:b6s++;break;
        default:cout<<"Bad Dice"<<endl;
    }
}
for(int roll=1;roll<=1;roll++){
    die3=rand()%6+1;//[1,6]

    switch(die3){
        case 1:c1s++;break;          //Same as dice 1 and 2 but for dice 3
        case 2:c2s++;break;
        case 3:c3s++;break;
        case 4:c4s++;break;
        case 5:c5s++;break;
        case 6:c6s++;break;
        default:cout<<"Bad Dice"<<endl;
    }
    //Switch statement gives value of 1 to whatever case it lands on
    //independent if converts the 1 value to actual dice value
}if (a1s==1){
    a1s=1;          //Dice 1 is 1 if a1s =1
    cout<<"Dice 1 = "<<a1s<<endl;
}if (a2s==1){
    a2s=2;          //Dice 1 is 2 if a2s =1
    cout<<"Dice 1 = "<<a2s<<endl;
}if (a3s==1){
    a3s=3;          //Dice 3 is 1 if a3s =1
    cout<<"Dice 1 = "<<a3s<<endl;
}if (a4s==1){
    a4s=4;          //Dice 1 is 4 if a4s =1
    cout<<"Dice 1 = "<<a4s<<endl;
}if (a5s==1){
    a5s=5;          //Dice 1 is 5 if a5s =1
    cout<<"Dice 1 = "<<a5s<<endl;
}if (a6s==1){
    a6s=6;          //Dice 1 is 6 if a6s =1
    cout<<"Dice 1 = "<<a6s<<endl;
}

if (b1s==1){
    b1s=1;          //The same for dice 2 as dice 1 independent if's
    cout<<"Dice 2 = "<<b1s<<endl;
}

```

```

}if (b2s==1){
    b2s=2;
    cout<<"Dice 2 = "<<b2s<<endl;
}if (b3s==1){
    b3s=3;
    cout<<"Dice 2 = "<<b3s<<endl;
}if (b4s==1){
    b4s=4;
    cout<<"Dice 2 = "<<b4s<<endl;
}if (b5s==1){
    b5s=5;
    cout<<"Dice 2 = "<<b5s<<endl;
}if (b6s==1){
    b6s=6;
    cout<<"Dice 2 = "<<b6s<<endl;
}

if (c1s==1){          //The same for dice 3 as dice 1 and dice 2 independent if's
    c1s=1;
    cout<<"Dice 3 = "<<c1s<<endl;
}if (c2s==1){
    c2s=2;
    cout<<"Dice 3 = "<<c2s<<endl;
}if (c3s==1){
    c3s=3;
    cout<<"Dice 3 = "<<c3s<<endl;
}if (c4s==1){
    c4s=4;
    cout<<"Dice 3 = "<<c4s<<endl;
}if (c5s==1){
    c5s=5;
    cout<<"Dice 3 = "<<c5s<<endl;
}if (c6s==1){
    c6s=6;
    cout<<"Dice 3 = "<<c6s<<endl;
}
cout<<endl;
//if the players number is the same as all 3 dice value its triple payout
if((num==a1s||num==a2s||num==a3s||num==a4s||num==a5s||num==a6s)&&
    (num==b1s||num==b2s||num==b3s||num==b4s||num==b5s||num==b6s)&&
    (num==c1s||num==c2s||num==c3s||num==c4s||num==c5s||num==c6s)){
    cout<<"You win by Triple! - Triple the payout";
    test=true;
    place=3;
}

```

```

isTpl(place);
if (isTpl(place)==true){
    tpay(bet);
    cout<<"You won $"<<bet<<endl;
    cout<<endl;
}
}
//if the players number is the same as Die 1 and 2 its double payout
else if ((num==a1s||num==a2s||num==a3s||num==a4s||num==a5s||num==a6s) &&
    (num==b1s||num==b2s||num==b3s||num==b4s||num==b5s||num==b6s)){
    cout<<"You win by double! - double the payout"<<endl;
    test=true;
    place=2;
    isDbI(place);
    if (isDbI(place)==true){
        dpay(bet);
        cout<<"You won $"<<bet<<endl;
        cout<<endl;
    }
}
//If the players number is the same as die 2 and 3 its double payout
else if ((num==c1s||num==c2s||num==c3s||num==c4s||num==c5s||num==c6s) &&
    (num==b1s||num==b2s||num==b3s||num==b4s||num==b5s||num==b6s)){
    cout<<"You win by double! - double the payout"<<endl;
    test=true;
    place=2;
    isDbI(place);
    if (isDbI(place)==true){
        dpay(bet);
        cout<<"You won $"<<bet<<endl;
        cout<<endl;
    }
}

//if the players number is the same as die 1 and 3 its double payout
} else if ((num==a1s||num==a2s||num==a3s||num==a4s||num==a5s||num==a6s)&&
    (num==c1s||num==c2s||num==c3s||num==c4s||num==c5s||num==c6s)){
    cout<<"You win by double! - double the payout"<<endl;
    test=true;
    place=2;
    isDbI(place);
    if (isDbI(place)==true){
        dpay(bet);
        cout<<"You won $"<<bet<<endl;
        cout<<endl;
    }
}

```

```

    }
}
//If the players number is the same as any of die 1,2,or 3
else if(num==a1s||num==a2s||num==a3s||num==a4s||num==a5s||num==a6s||
    num==b1s||num==b2s||num==b3s||num==b4s||num==b5s||num==b6s||
    num==c1s||num==c2s||num==c3s||num==c4s||num==c5s||num==c6s){
    cout<<"You win"<<endl;
    test=true;
    place=1;

    if ((isTpl(place)==true)&&(isDbI(place)==false)){

        cout<<"You won $"<<bet<<endl;
        cout<<endl;
    }
}
else{    //If players number doesn't match any die they lose
    cout<<"You Lose"<<endl;
    test=false;
    cout<<"Pay the dealer $"<<ogbet<<endl;
    cout<<endl;
}

if (test==true){    //Test is true when a win occurs
    int payout=ogbet+bet;
    cout<<"You have won a payout of $"<<payout<<endl;
    cout<<endl;
    wins++;    //Wins increments
}
else{    //Test is false when a loss occurs
    lose++;    //Loss increments
}

WL=wins>=lose?true:false;
if(WL==true){
    cout<<"You have won "<<wins-lose<<" more games than you have lost"<<endl;
}
else{
    cout<<"You have lost "<<lose-wins<<" more games than you have won"<<endl;
}

games++;    //Games increment after every game
lead[0][0]=games;
lead[0][1]=wins;

```

```

lead[0][2]=lose;
percwin=(wins/games)*PERCENT;    //Percentage of games won conversion
percL=(lose/games)*PERCENT;    //Percentage of games lost conversion
cout<<"You have played "<<games<<" games"<<endl;    //Output # of games
cout<<"You have won "<<wins<<" games"<<endl;    //Output # of games won
cout<<"You have lost "<<lose<<" games"<<endl;    //Output # of games lost
cout<<endl;
cout<<"You have won "<<round(percwin)<<" % of games"<<endl; //% of wins rounded to nearest
percent
cout<<"You have lost "<<round(percL)<<" % of games"<<endl; //% of losses rounded to near
percent
cout<<endl;

cout<<setw(20)<<"Games Played "<<setw(12)<<"Games Won "<<setw(12)<<" Games Lost"<<endl;
cout<<setw(15)<<lead[0][0]<<setw(12)<<lead[0][1]<<setw(12)<<lead[0][2]<<endl;

cout<<endl;
cout<<"Would you like to play again?"<<endl;    //Prompts if player would like to play again
cout<<"Enter Y for yes and any other letter to stop"<<endl;
cin>>again;    //Input y to play again or any other letter to stop

} while (again=='Y' || again=='y');    //Game continues so long as input is y
term();
return 0;
}

if (start=="Exit" || start=="exit"){    //If exit is typed at home screen game ends
cout<<endl;
cout<<"Thanks for Playing!";
return 0;

}
//Display Results
//Outputs games to file
output<<"You have played "<<games<<" games"<<endl;
output<<"You have won "<<wins<<" games"<<endl;
output<<"You have lost "<<lose<<" games"<<endl;
output<<endl;
output<<setprecision(2)<<"You have won "<<round(percwin)<<" % of games"<<endl;
output<<setprecision(2)<<"You have lost "<<round(percL)<<" % of games"<<endl;

output<<setw(20)<<"Games Played "<<setw(12)<<"Games Won "<<setw(12)<<" Games Lost"<<endl;
output<<setw(15)<<lead[0][0]<<setw(12)<<lead[0][1]<<setw(12)<<lead[0][2]<<endl;

```



```
//Exit Stage right
//Closes input and output
input.close();
output.close();
return 0;
}
```

```
void term(){
    cout<<endl;
    cout<<"Thanks for playing!"<<endl;
    exit(0);
}
```

```
void single(int sing){
    sing=1;
}
```

```
bool isTpl(int place){
    bool status;
    if(place%2==0){
        status=false;
    }
    else{
        status=true;
    }
    return status;
}
```

```
bool isDbl(int place){
    bool status;
    if(place%2==0){
        status=true;
    }
    else{
        status=false;
    }
    return status;
}
```

```
void dpay(int &bet, int dbl){
    bet=bet*dbl;
}
```

```
void tpay(int &bet,int tpl){  
    bet=bet*tpl;
```

```
}
```

```
void view(vector<int> num){  
    cout<<setw(8)<<num[0]<<setw(8)<<num[1]<<setw(8)<<num[2];  
}
```

```
void model (vector<int> num){  
    num[0]=1;  
    num[1]=2;  
    num[2]=3;  
}
```

```
void modelA(int numA[]){  
    numA[0]=1;  
    numA[1]=2;  
    numA[2]=3;  
}
```

```
void viewA(int numA[]){  
    cout<<setw(8)<<numA[0]<<setw(8)<<numA[1]<<setw(8)<<numA[2];  
}
```