

Mini Project #1 Report

“23 Toothpicks”

Zac Porter

Section 1:

The objective of the mini-project code is to implement a simple game called the "23 Toothpick Game" where two players take turns removing 1 to 3 toothpicks from a pile of 23 toothpicks. The player who takes the last toothpick loses. The program includes a computer opponent with a strategy for picking toothpicks.

Section 2:

The program is designed to follow a turn-by-turn structure where the player and computer take alternate turns choosing the number of toothpicks to remove. Below is a description of the program's flow:

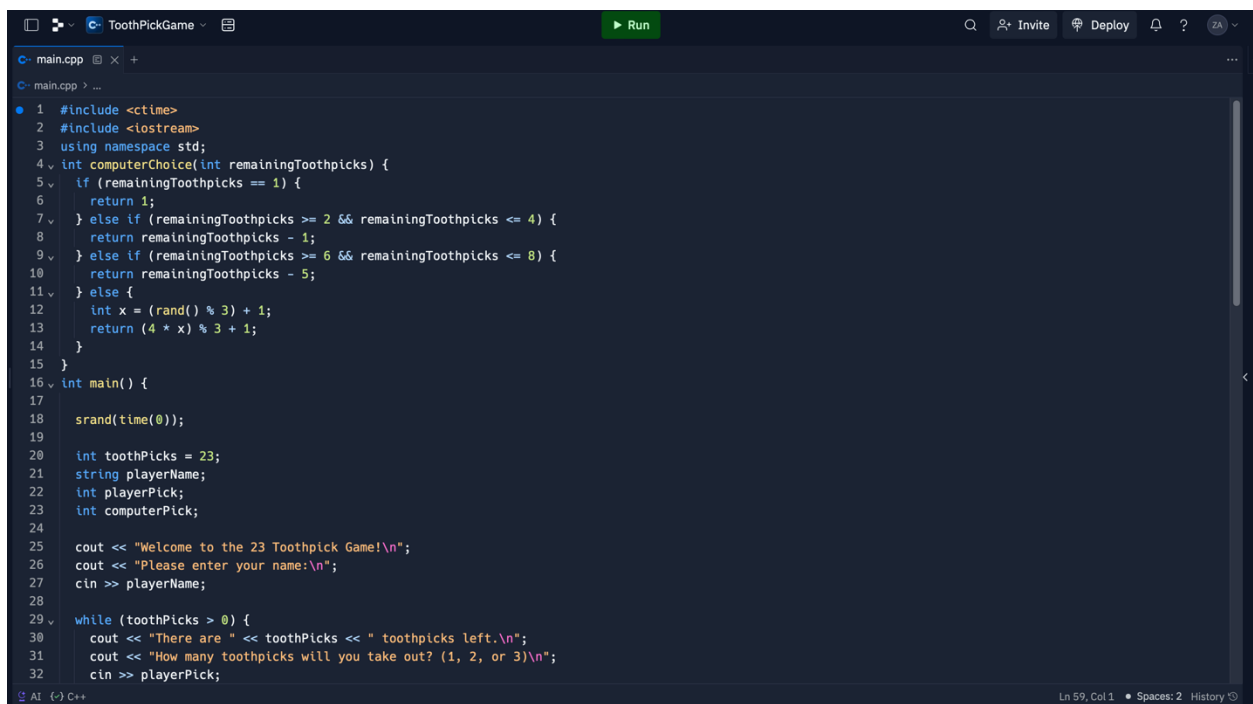
- 1) Initialization: Seed the random number generator for the computer's choices.
Initialize the total number of toothpicks to 23. Get the player's name as input.
- 2) Loop: While there are still toothpicks remaining: Display the current number of toothpicks. Prompt the player to input the number of toothpicks to remove.
Validate the player's input, ensuring it is a valid move. Update the number of remaining toothpicks after the player's move. Check if the player has won (removed the last toothpick). If the game is not over: Calculate the computer's move using the `computerChoice` function. Display the number of toothpicks the computer removed. Update the number of remaining toothpicks after the computer's move. Check if the computer has won (removed the last toothpick).

3) End Game: Display a message indicating whether the player or computer won.

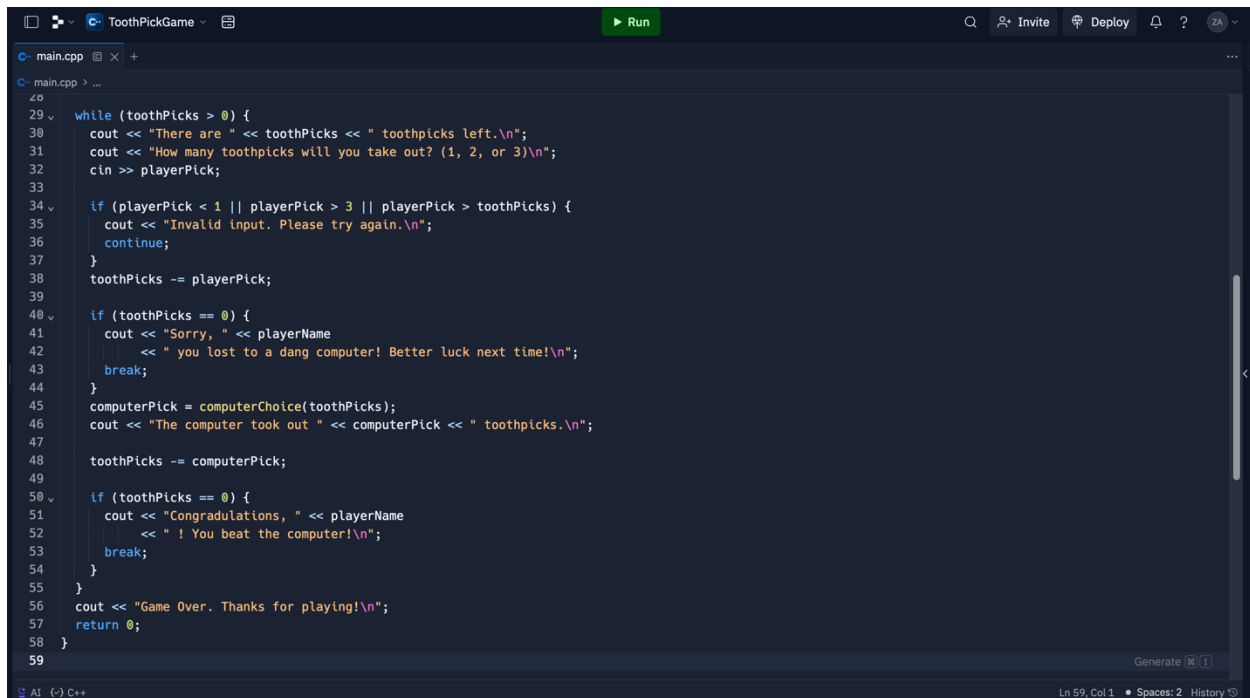
End the game.

Section 3:

<https://replit.com/join/dozjgxobec-zacporter179>



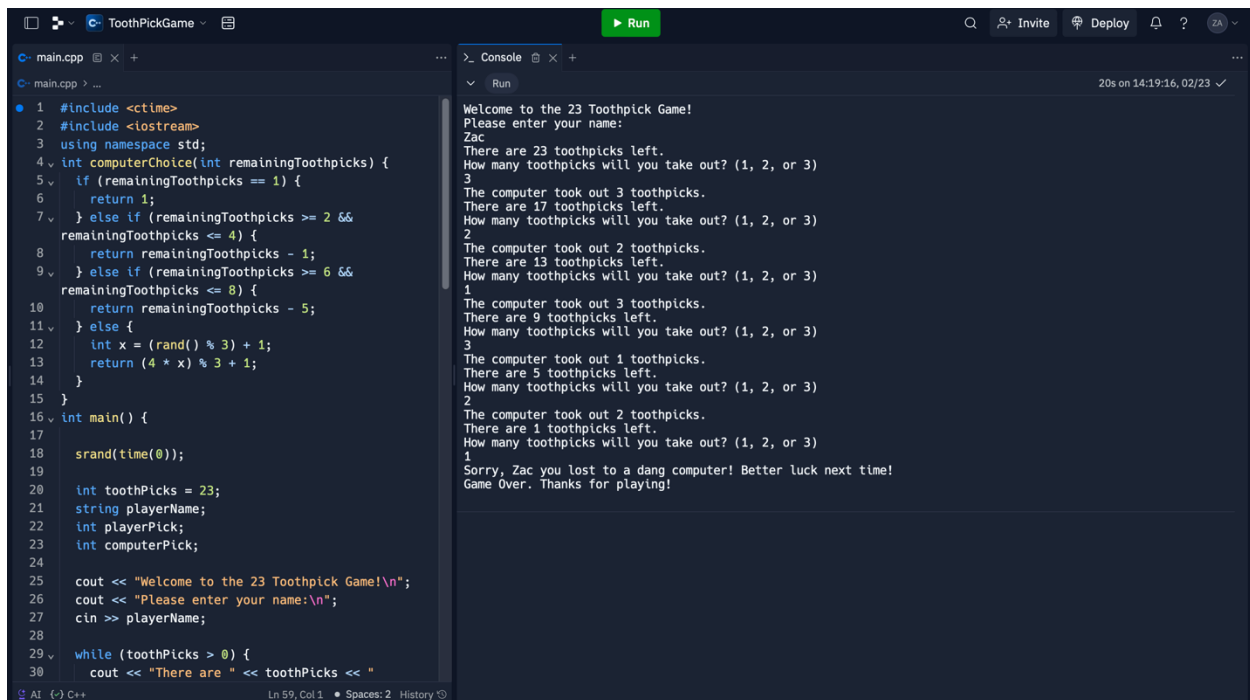
```
1 #include <ctime>
2 #include <iostream>
3 using namespace std;
4 int computerChoice(int remainingToothpicks) {
5     if (remainingToothpicks == 1) {
6         return 1;
7     } else if (remainingToothpicks >= 2 && remainingToothpicks <= 4) {
8         return remainingToothpicks - 1;
9     } else if (remainingToothpicks >= 6 && remainingToothpicks <= 8) {
10        return remainingToothpicks - 5;
11    } else {
12        int x = (rand() % 3) + 1;
13        return (4 * x) % 3 + 1;
14    }
15 }
16 int main() {
17     srand(time(0));
18
19     int toothPicks = 23;
20     string playerName;
21     int playerPick;
22     int computerPick;
23
24     cout << "Welcome to the 23 Toothpick Game!\n";
25     cout << "Please enter your name:\n";
26     cin >> playerName;
27
28     while (toothPicks > 0) {
29         cout << "There are " << toothPicks << " toothpicks left.\n";
30         cout << "How many toothpicks will you take out? (1, 2, or 3)\n";
31         cin >> playerPick;
```



```

29 while (toothPicks > 0) {
30     cout << "There are " << toothPicks << " toothpicks left.\n";
31     cout << "How many toothpicks will you take out? (1, 2, or 3)\n";
32     cin >> playerPick;
33
34     if (playerPick < 1 || playerPick > 3 || playerPick > toothPicks) {
35         cout << "Invalid input. Please try again.\n";
36         continue;
37     }
38     toothPicks -= playerPick;
39
40     if (toothPicks == 0) {
41         cout << "Sorry, " << playerName
42             << " you lost to a dang computer! Better luck next time!\n";
43         break;
44     }
45     computerPick = computerChoice(toothPicks);
46     cout << "The computer took out " << computerPick << " toothpicks.\n";
47
48     toothPicks -= computerPick;
49
50     if (toothPicks == 0) {
51         cout << "Congradulations, " << playerName
52             << " ! You beat the computer!\n";
53         break;
54     }
55 }
56 cout << "Game Over. Thanks for playing!\n";
57 return 0;
58 }
59

```



```

1 #include <ctime>
2 #include <iostream>
3 using namespace std;
4 int computerChoice(int remainingToothpicks) {
5     if (remainingToothpicks == 1) {
6         return 1;
7     } else if (remainingToothpicks >= 2 &&
8         remainingToothpicks <= 4) {
9         return remainingToothpicks - 1;
10    } else if (remainingToothpicks >= 6 &&
11        remainingToothpicks <= 8) {
12        return remainingToothpicks - 5;
13    } else {
14        int x = (rand() % 3) + 1;
15        return (4 * x) % 3 + 1;
16    }
17 }
18 int main() {
19     srand(time(0));
20
21     int toothPicks = 23;
22     string playerName;
23     int playerPick;
24     int computerPick;
25
26     cout << "Welcome to the 23 Toothpick Game!\n";
27     cout << "Please enter your name:\n";
28     cin >> playerName;
29
30     while (toothPicks > 0) {
31         cout << "There are " << toothPicks << "

```

Console Output:

```

Welcome to the 23 Toothpick Game!
Please enter your name:
Zac
There are 23 toothpicks left.
How many toothpicks will you take out? (1, 2, or 3)
3
The computer took out 3 toothpicks.
There are 17 toothpicks left.
How many toothpicks will you take out? (1, 2, or 3)
2
The computer took out 2 toothpicks.
There are 13 toothpicks left.
How many toothpicks will you take out? (1, 2, or 3)
1
The computer took out 3 toothpicks.
There are 9 toothpicks left.
How many toothpicks will you take out? (1, 2, or 3)
3
The computer took out 1 toothpicks.
There are 5 toothpicks left.
How many toothpicks will you take out? (1, 2, or 3)
2
The computer took out 2 toothpicks.
There are 1 toothpicks left.
How many toothpicks will you take out? (1, 2, or 3)
1
Sorry, Zac you lost to a dang computer! Better luck next time!
Game Over. Thanks for playing!

```

Section 5:

The mini project successfully implemented the 23 Toothpick Game with a computer opponent.

Challenges included designing an effective strategy for the computer's choices and ensuring input validation for the player's moves. Successes include creating a functional and entertaining

game with a clear flow of turns. The program encourages strategic thinking and decision-making. Future improvements could involve refining the computer's strategy or adding additional features to enhance the gaming experience. Overall, the project provided valuable experience in designing and implementing a simple turn-based game.

Project Flowchart:

