|  |
| --- |
| CEWP MOD4 **INTRODUCTION TO JAVA** FALL 2019  Deliverable 1  Yahtzee project  (Group B/Team E)  Lucas CHAMPSAUR / Lucas PECH / Arthur RAPP |

Table of Contents

1 - Reflection on working with the others towards a common goal

2 – Contribution on deliverable

1) Reflection on working with the others towards a common goal

Lucas CHAMPSAUR:

Although I had already work on a project as the member of a team several times, this time I had to do it with people that I don’t necessarily know that well. However, I think that we worked well together. We got along really quickly and have been able to work efficiently together. There were no disagreements on the project nor on the organization of the work. Though, we still have a lot of work until we finish this project completely.

Lucas PECH:

Working as a team is not something new for me, we are doing for 2 years now. But the fact is that I usually work with the same person, Lucas is one of these persons, while Arthur is not. At first, I was a bit afraid to be with somebody I never worked with. However, after speaking all three about the project we find a good way to work together by meeting each other after classes and using GitHub. Moreover, the time at the end of classes we had to spend on the project helped us helped us to know us better.

Finally, this first phase of the project went well, the cooperation helped us to go way faster and the fact of being 3 to think about the same project but in different way allowed us to find the best way to do it. I hope the other phase will be like this one.

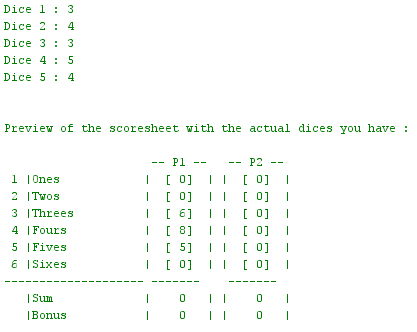
Arthur RAPP:

Working with both Lucas turn out better than expected. We start the project as soon as possible because we didn’t know how much time it would take to complete it. Each one had a task to complete and for each it worked out very well. I'm really pleased with my teammates and wouldn't mind to keep working with them on this project.

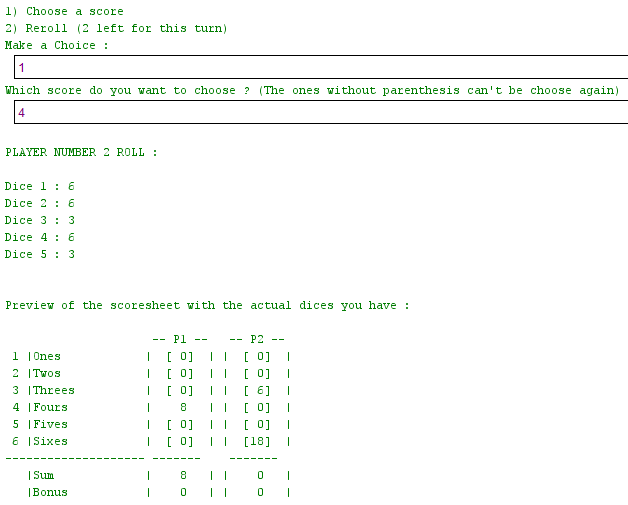
2) Contribution on deliverable

|  |  |
| --- | --- |
| **Group B / Team E** | **Contributions** |
| **Lucas CHAMPSAUR** | * Security of the inputs. * Optimization of the algorithm. * Implementation of the “play again” option. * Implementation of the exit of the algorithm. * Improvements on the display of the game. * Detection of the winner when playing multiplayer. * Correction of mistakes created by the implementation of the lower part of the scoreboard |
| **Lucas PECH** | * Implementation of the function that enters the score in the scoresheet and reset the other rows. * Implementation of the condition that forbid the user to choose the same score again. * Implementation of the end condition and of the calculation of the score. * Implementation of the limitation of 2 reroll per turn. * Implementation of multiplayer * Implementation of detection of all lower section rows |
| **Arthur RAPP** | * Implementation of all the function permitting to roll the dice. * Implementation of the function permitting to reroll. * Implementation of the menu permitting to choose if we want to reroll or block a score. * Implementation of the display of the scoresheet. * Separation of the code in different class |

**Tests:**

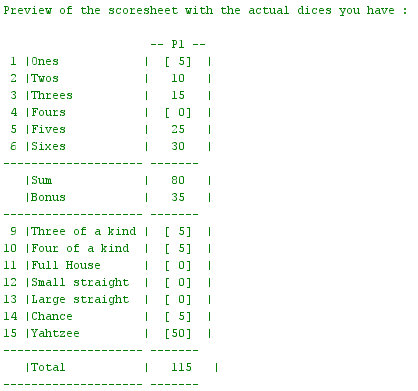
**Test of the upper section:**

Here we can see that depending of the values of the dices the upper section is filled well.

**Test choose a score:**

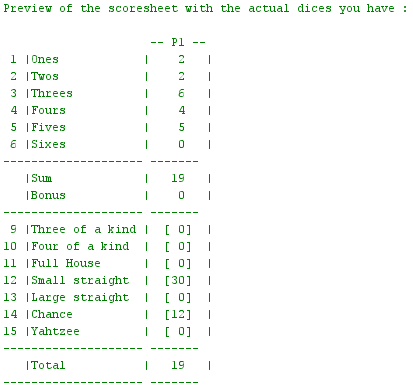
Here we can see that if the player 1 choose to fill the 4th row then the value if well blocked in its scoresheet.

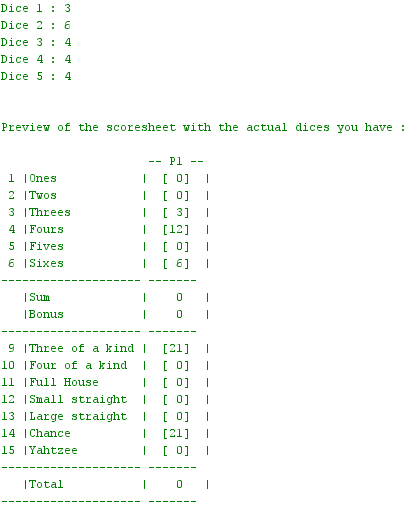
**Test bonus and total score:**

****

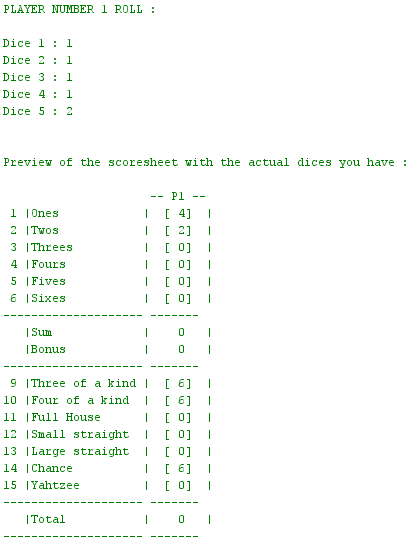
Here we can see that the sum of the upper section is working well. It does the addition of all the blocked rows. The bonus section also works: the bonus of 35 points appear because there is more than 63 point in the sum.

The total score also works, it does the addition of all the blocked rows.

We can see on this screen that if the sum of the upper section is lower than 63 then the bonus of 35 point is not given to the player.

**Test 3 of a kind:**

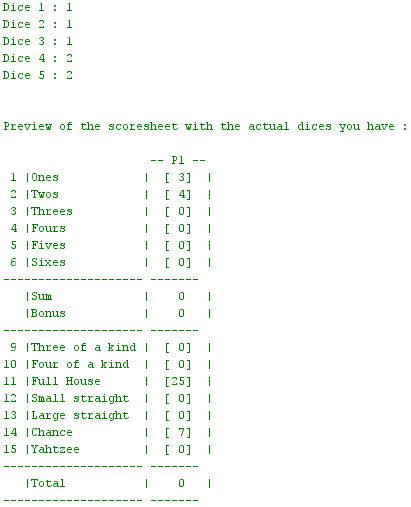
He we can see that there are 3 dice of the same kind (4,4,4) and that the row corresponding to this combination is well filled with the sum of all the dices.



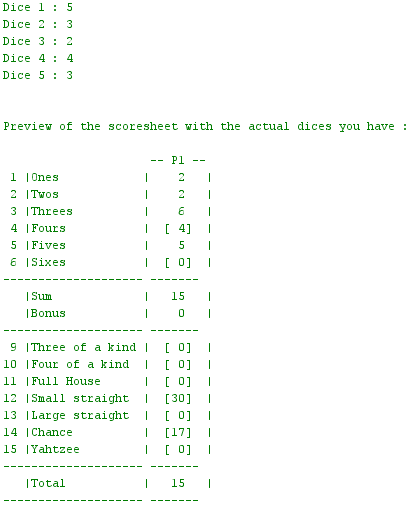
**Test 4 of a kind:**

He we can see that there are 4 dice of the same kind (1,1,1,1) and that the row corresponding to this combination is well filled with the sum of all the dices.

We also observe that the row three of a kind is filled with the sum of the dices which is logic.

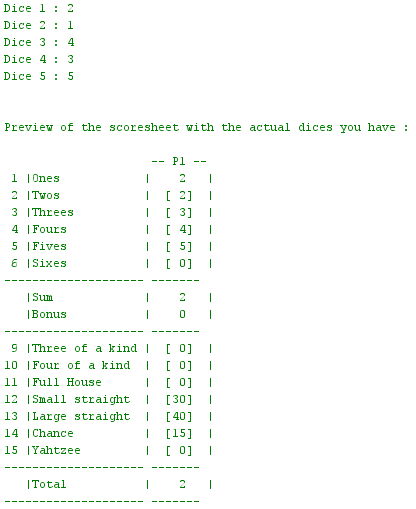
**Test full house:**

Here we can observe that the full house row is working well. In fact the dices are forming a full house (a pair of two and a brelan of ones) and the full house row is filled with 25 point.

****

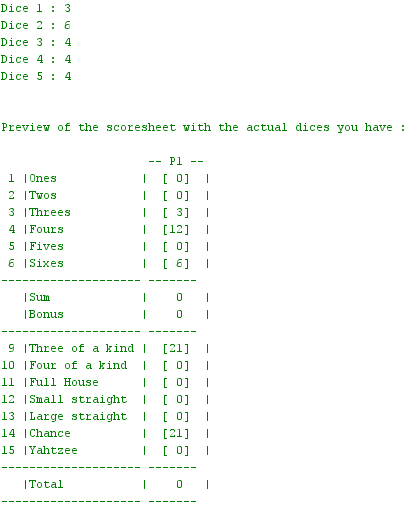
**Test small straight:**

Here we can see that there is a small straight in the roll (2,3,4,5) and that the small straight row is well filled with 30 point.

**Test large straight:**

Here we can see that there is a large straight in the roll (1,2,3,4,5) and that the large straight row is well filled with 40 point.

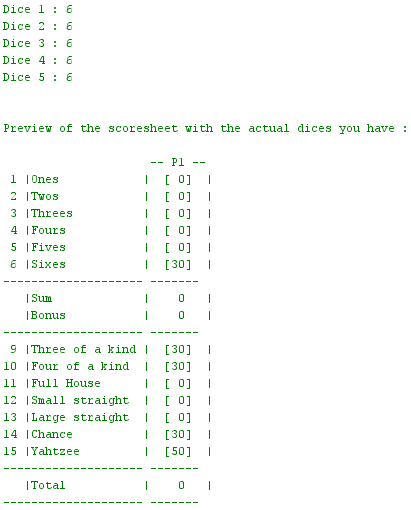
We also observe that the small straight row is filled too which is logic as a large straight is also a small one.

****

**Test chance:**

Here we observe that the chance row works well. This row is always filled with the total of all the dice of the roll.

Here 3+6+4+4+4 = 21 and the chance row is filled with 21.

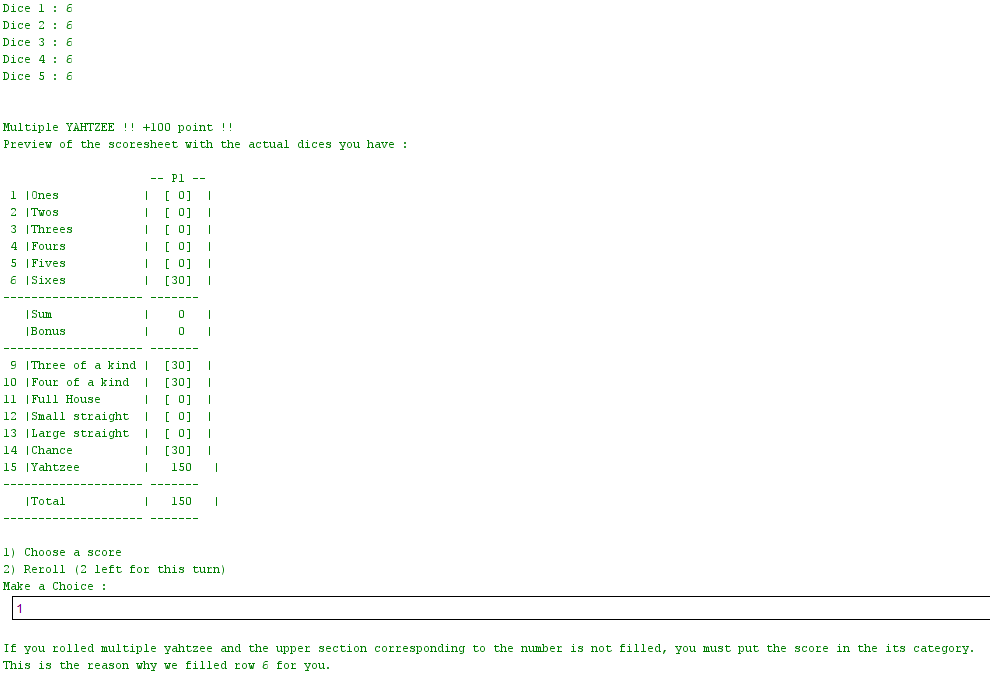
**Test yathzee:**

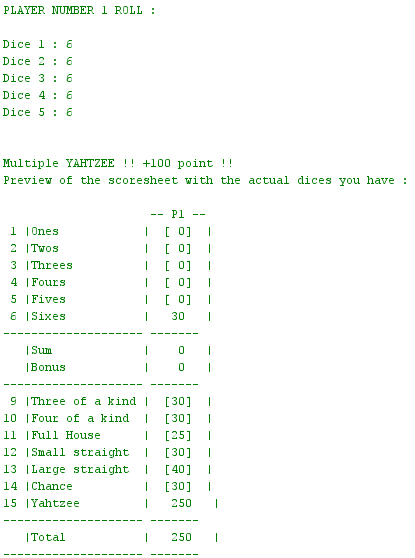
Here we have 5 same dices and the Yahtzee column if well filled with 50 points.

The three/four of a kind columns are filled as well which means they work well.

**Test multiple yahtzee:**

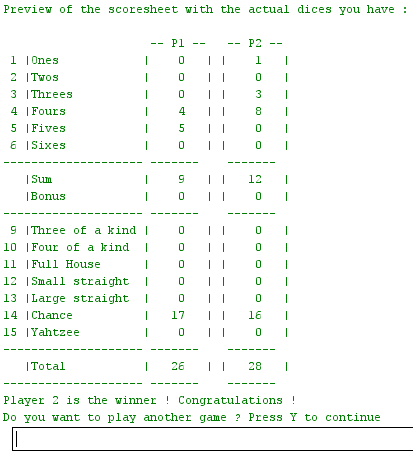
Here we are testing multiple Yahtzee. The rules say that when a player is having a Yahtzee and he has already filled the Yahtzee row, then he get 150 bonus points and if the row corresponding to the number he got 5 times is not filled in the upper case then he MUST block the case of this number in the upper section. So here if the player chooses to not reroll, we do not let him the choice and we block automatically the corresponding row in the upper section.



**Test multiple Yahtzee:**

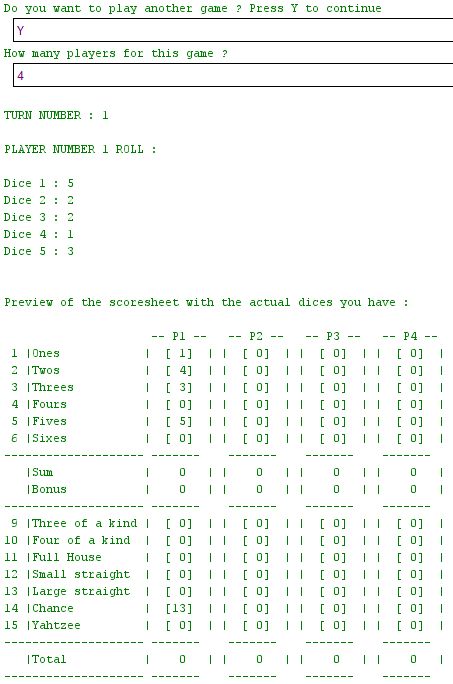
Here we can observe what happen when the player gets multiple Yahtzee and the upper section is already filled.

The rules say that he can choose the row he wants in the lower section scoring the normal score. We the program is doing exactly this.

**Test end of a game:**

Here we can see that the scoresheet has been fully filled by the 2 player so the game end and the winner is showed.

We can see that it asks the user if he wants to play again or not.

**Test new game and choice of the number of players:**

Here we can see that if the user press Y another game is started.

We also observe that the choice of the number of players is working well. In fact the user select 4 player and so the scoresheet got 4 columns, one for each of the player.

**Ideas for enhancements:**

There are two types of enhancements we could make to improve our actual Yahtzee.

First, we could make enhancements that offers the player a more enjoyable experience while playing. For example, we could add a graphical interface to the game. This would improve a lot the clearness and fun of the game. Also, we could find a new way of keeping dices during a reroll. The user could enter directly all the dices he wants to keep rather than saying yes/no five times.

Then, we could make enhancements that offers more options to the player. For example, we could implement a high score option in our game.