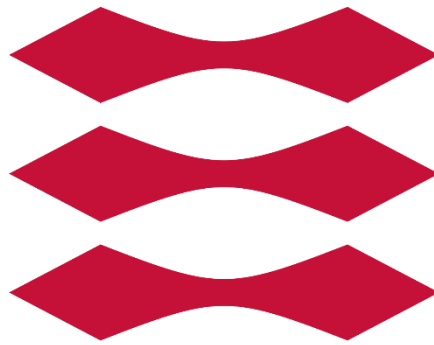


# DTU



02314 62531 62532

Introductory Programming, Development Methods for IT Systems and  
Version Control and Testing Methods

---

## CDIO 1

---

### Group 22

s235068 - Casper Kielstrup

s235108 - Emil K. Petersen-Dalum

s235124 - Mirza Zia Beg

s235079 - Mohammed Ismail

s235771 - Younes Hamadi

---

Delivery deadline:

Friday, 29 September 2023

---

# Contents

- Summary
- Hourly accounting
- Introduction
- Project planning
- Requirements/analysis
- Design
- Implementation
- Testing
- Conclusion
- Appendix
  - Literature
  - Code

## Summary

We have been tasked to make a dice game for IOOuterActive.

## Hourly accounting

Each person in the group has worked 6 hours per week on Campus, and at least 5 hours per week at home.

## Introduction

In this project for the courses Introductory programming, Development methods for IT systems, and Version Control and test methods, we have been tasked by the company IOOuterActive to make a dice game.

We have been divided into groups to work on this project to analyze, design, implement and test this project. Our Assistant teachers function as IOOuterActive's project leader and let us "talk" to the costumer.

## Project planning

The project was divided into 5 parts, 1 for each group member:

- The base game
- Pair of 1s
- Pair of 6s
- Win with 2 pairs of 6s.
- Need a pair to win.

The group members worked individually to get their part of the code running before the group met up Monday 25/9 – 23, where they could check for bugs, talk about problems that arose and to get help from the other group members.

## Requirements

IOOuterActive have the following mandatory requirements for the dice game:

- It should have 2 players.
- It should use 2 dice.
- The winner should be the first player to hit 40 points.
- It should be playable without the need to read an instruction manual.
- The rolled dice should be concluded as working correctly after 1000 throws.
- It should work on the Windows systems in the Data bars at DTU.

Besides these requirements, IOOuterActive have the following optional requirements:

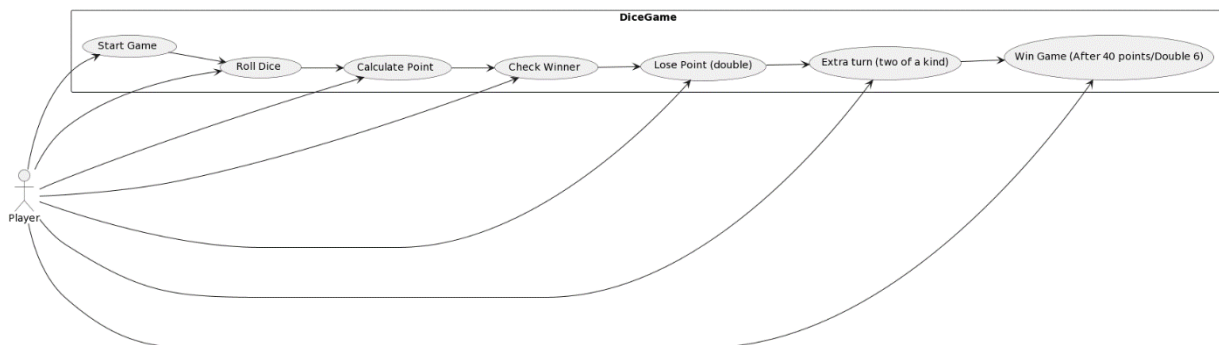
- The players lose all points if they roll a pair of 1s.
- The players gain an extra turn if they roll a pair of 6s.
- The player could win by rolling a pair of 6s if the same player rolled a pair of 6s on the previous roll, even if it was during an extra turn or the previous turn.
- The players would need to roll any pair to win after getting 40 points.

## Design

Actor – Players 1 and 2.

Use cases

- Start Game
- Roll Dice
- Calculate points
- Check Winner
- Lose points (Double 1)
- Extra turn. (Two of a kind)
- Win (with double 6)
- Win the Game (After 40 points/ or with double 6).



## Implementation

Since it is a requirement that the program should be able to be run on the computers in the data bars at DTU, we have made it so that it uses the terminal. The data bars at DTU all have java installed, so the program can be run by executing the compilation command, followed by the execute command. For example, if the .java files were in one folder, open a new terminal in that folder, and type “javac \*.java”, followed by “java App”.

## Testing

Sum of 2 Dice Chart						
	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

For a roll of 2 dice, the expected results are as follows:

2 in 1/36 of rolls

3 in 2/36 of rolls

4 in 3/36 of rolls

5 in 4/36 of rolls

6 in 5/36 of rolls

7 in 6/36 of rolls

8 in 5/36 of rolls

9 in 4/36 of rolls

10 in 3/36 of rolls

11 in 2/36 of rolls

12 in 1/36 of rolls

Furthermore, a pair should show up in 6/36 of rolls

For 1000 rolls, the expected outcome should be:

2 in 27/1000 of rolls

3 in 56/1000 of rolls

4 in 83/1000 of rolls

5 in 110/1000 of rolls

6 in 138/1000 of rolls

7 in 166/1000 of rolls

8 in 138/1000 of rolls

9 in 110/1000 of rolls

10 in 83/1000 of rolls

11 in 56/1000 of rolls

12 in 27/1000 of rolls

And we should expect a pair to show up 166/1000 rolls.

To test this, we created a program that runs the dice roll and outputs the result 1000 times.

For this test, the actual outcomes were:

2 in 27/1000 of rolls

3 in 46/1000 of rolls

4 in 92/1000 of rolls

5 in 121/1000 of rolls

6 in 155/1000 of rolls

7 in 161/1000 of rolls

8 in 128/1000 of rolls

9 in 115/1000 of rolls

10 in 82/1000 of rolls

11 in 53/1000 of rolls

12 in 20/1000 of rolls

Most of the actual outcomes were close to the expected outcomes, so it was a successful test

If you want to test it out yourself, the program is called “dietest”.

## Conclusion

From our testing, we can conclude that the program works correctly, since our observed outcomes closely match the expected outcomes, and after testing the program on people, who do not have any knowledge on the inner workings of the program, we can also conclude that the program is user friendly. We can further conclude that the program fulfils the requirements from IOOuterActive, including the optional functions.

## Appendix

- Literature
  - Sum of 2 dice diagram: <https://andymath.com/probability-with-dice/>
- Code:
  - [https://github.com/ZabZZZZZ/22\\_del1](https://github.com/ZabZZZZZ/22_del1)