**Game report**

Jan Kalinowski p17179637

1. **Overview**

For my C++ assignment I decided to make a mini golf type of game, so I could use Box2D’s and SFML’s maximum amount of features. In this report I would like to explain my design decisions and approach to testing.

1. **Used software**

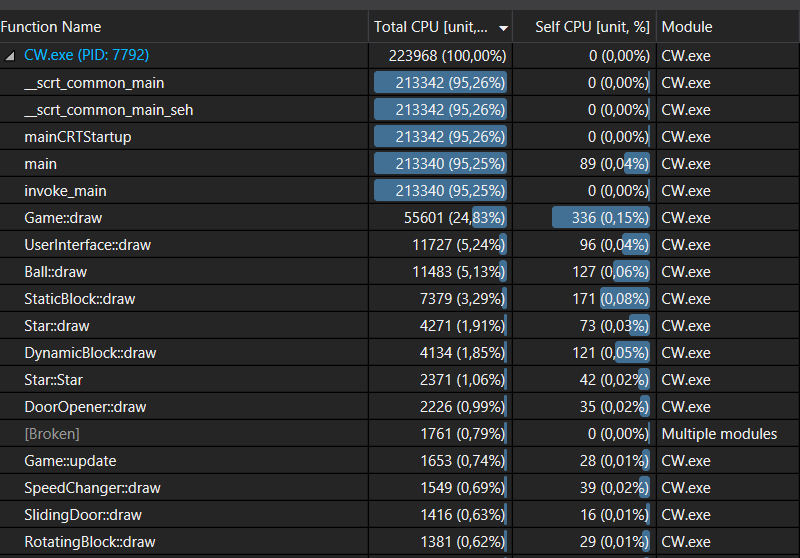
Game is written in C++. Box2D provides physics and SFML provides typical game features, such as graphics and sounds.

1. **Design**

In order to use both Box2D and SFML to their limits I have made a mini golf game with various obstacles, such as ramps, walls, hittable doors, doors opened with buttons, boosters, spinning objects, ice/sand fields etc. This allowed me to use maximum amount of features provided by SFMl and Box2D. Rather than trying to make game competitive I decided to make games relaxing, yet challenging, so I went for puzzle approach to golf game. Players have to solve simple riddles and collect stars along the way to the goal. When players reach the goal, the game loads next level. In case there’s no more levels, game ends.

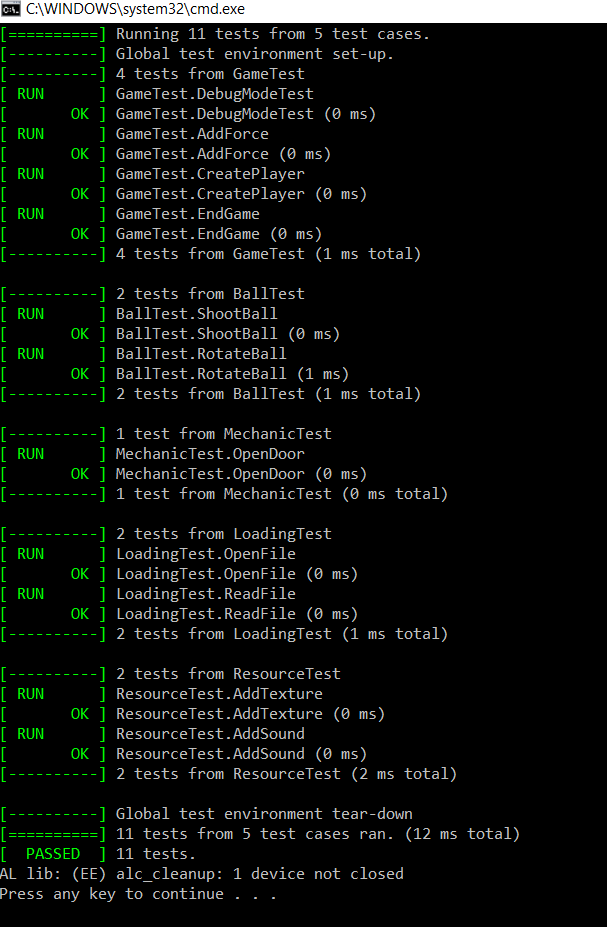
1. **Testing**

In order to make sure my software works correctly and runs smoothly, I have used testing tools to quickly check if code works and how effective it is. I have used Visual Studio performance profiler and I was pleased to see, that written code seems to be efficient.



Above is a report generated by the profiler. Those are the functions that used most of the CPU during whole ~5 minutes play through the levels. Most of the CPU has been taken by main() and drawing shapes, which sounds right, as they are being drawn and updated every single frame.

I have also used Google Unit Tests in order to check whether code runs correctly. I wrote 11 tests which were testing some of the most critical parts of my code, such as opening and reading a file or important game mechanics, for example ball rotating and shooting.



Above is the result produced by running those tests. All tests have passed successfully.

1. **Learning outcome**

Development of this game helped me gain better understanding of pointers, maths, design patterns and other important C++ features, such as RTTI. I found it fun to develop, yet still challenging, finding something new to create on each step of development. I am sure that experience gained from this project will pay off in the future, helping me to write better code and be a better developer.