

DV1445 Coursework 3 : Project Descriptions

Deadline: Flexible

1 Overview

The final assessment in the course is a programming project. This task is expected to be of extended duration and is used to evaluate that:

- Your level of knowledge about C++ (syntax, data-types, control-structure etc) is advanced enough that you can tackle a medium-sized software project.
- Your understanding of the language, and related terminology, is good enough to switch from the teacher-guided tuition of the lab-sessions to self-guided exploration.
- Your technique for problem-solving is sophisticated enough that you can design a piece of software of reasonable complexity, and then implement the design.

You may choose to work individually, or in pairs for this assignment. Several project descriptions are included below as samples of how complex the software that you write should be. You may choose one of the provided project descriptions, or you can write your own. If you choose to write your own then you must email me a proposal (maximum 500 words) that covers the same criteria as below.

The process for the software project is as follows:

1. Choose one of the provided descriptions or write your own.
2. Email `awm@bth.se` with your choice. Attach a zip-file containing all of your lab-exercises from the course. Each file should be named `exA-B.cpp` where `A` is the week number and `B` is the exercise number. ONLY include the `.cpp` source files, not any other files.
3. I will check your proposal and email you with either approval, or changes you must make.
4. You must then implement the project (in previous years this has taken students 1 – 6 months depending on ability and how much effort they put into it).
5. When your project is complete, email me and I will arrange a viva for grading.

2 Description I

Breakout is a simple action game by Atari in the early 80s. The player moves a bat at the bottom of the screen from left to right. A ball bounces between the bat and a number of bricks at the top of the screen. Every time the ball touch a brick the brick is removed and the player gains points. A more detailed description can be found online, as well as screenshots and videos.

In this project you will design a simple version of this game and implement it using SDL. You will need a form of user input to move the bat (keyboard or mouse can be appropriate). You will need to maintain the state of the game (e.g ball position, bat position, brick locations etc) and implement a state-update function to handle animation of the objects and handle events (such as collision detection) within the game.

This project tests your ability to interface to a large external library, to design a working representation of the game world, to implement the necessary functionality as callbacks and to successfully implement a FSM of the game that interfaces with those callbacks.

3 Description II

Lunar Lander is a simple action game. The player controls a spaceship and attempts to land successfully. The control allow the ship to rotate and to fire a thruster. The game simulates gravity and the player must use the thruster to manoeuvre succesfully onto the ground, landly softly enough not to crash.

In this project you will design a simple version of this game and implement it using SDL. You will need a form of user input to move the bat (keyboard or mouse can be appropriate). You will need to maintain the state of the game (e.g lander position, velocity and rotation) and implement a state-update function to handle animation of the objects, simulation of gravity, and handle events (such as collision detection) within the game.

This project tests your ability to interface to a large external library, to design a working representation of the game world, to implement the necessary functionality as callbacks and to successfully implement a FSM of the game that interfaces with those callbacks.

4 Description III

Backgammon is a classic boardgame. Each player rolls dice and then uses the numbers shown to move a set of counters around a board. The rules of the game determine which moves are valid and which moves are illegal. This project ignores the problem of implementing a computer player — you will only need to implement a version that two human players can use by taking turns.

In this project you will design a simple version of this game and implement it using SDL. You will need a form of user input (keyboard or mouse can be appropriate). You will need to maintain the state of the game (e.g counter positions, current dice values). You will need to check the validity of moves that the players try to make against the rules of the game. Only legal moves should be allowed, and illegal moves should be rejected.

This project tests your ability to interface to a large external library, to design a working representation of the game world, to implement the necessary functionality as callbacks and to successfully implement a validation of game moves from an externally defined set of rules.