CPS2004 — Object Oriented Programming

Assignment

Juan Scerri
B.Sc. (Hons)(Melit.) Computing Science and Mathematics (Second Year)

December 23, 2022

Contents

List of Figures

Listings

1 Plagiarism Declaration

Plagiarism is defined as *"the unacknowledged use, as one's own, of work of another person, whether or not such work has been published, and as may be further elaborated in Faculty or University guidelines"* (University Assessment Regulations, 2009, Regulation 39 (b)(i), University of Malta).

I, the undersigned, declare that the report submitted is my work, except where acknowledged and referenced. I understand that the penalties for committing a breach of the regulations include loss of marks; cancellation of examination results; enforced suspension of studies; or expulsion from the degree programme.

Work submitted without this signed declaration will not be corrected, and will be given zero marks.

Juan Scerri	CPS2004	December 23, 2022
Student's full name	**Study-unit code**	**Date of submission**

Title of submitted work: Object Oriented Programming Assignment

Student's signature

2 Village War Game

2.1 Language Choice

2.2 User Guide

2.3 Design

2.4 Technical Aspects

2.5 Testing

2.6 Limitations & Improvements

3 Minesweeper

3.1 Language Choice

C++ was chosen for Minesweeper because it has a fixed board size of 16×16. This means that it is possible to stack allocate every object removing the need for dynamic memory allocation. This is facilitated by `std::array` from the Standard Template Library (STL) which allows for the creation of fixed size arrays on the stack.

3.2 User Guide

3.2.1 Download, Compiling & Running

1. Clone the repository.

```
$ git clone https://github.com/JuanScerriE/minesweeper
```

2. Compile the tests and the game.

   Note: Make sure that `gtest` and `ncurses` are installed for the tests and the game, respectively.

```
$ cd minesweeper ; ./compile.sh
```

3. Run the tests.

```
minesweeper $ ./tests.sh
```

4. Run the game.

```
minesweeper $ ./run.sh
```

3.3 Design

3.4 Technical Aspects

3.5 Testing

3.6 Limitations & Improvements