Practical 9 (due 2021-05-28 @ 09:00)

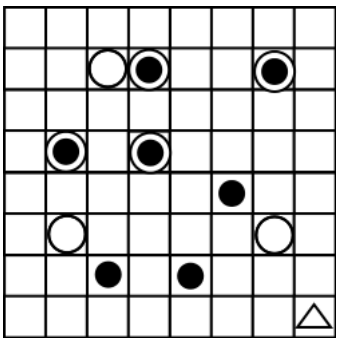
This practical extends an olive branch to students that are writing Computer Science 1A ST2 and\or Informatics 1A ST2. The aim of this practical is to assist students to practice either a Computer Science 1A past paper or and Information 1A past paper. To achieve this aim you have two options. You must select which option you would like to complete:

1.) Complete and upload any **Informatics 1A** past semester test 2 **OR**
2.) Complete and upload the solution to the **Computer Science 1A** past paper described below.

Since the submissions are due the last day of the semester, a participation only mark is awarded. The marksheet attached to this document is for self-evaluation only.

Shelter In Place

The Event has required that Utopian Citizens shelter in place. The Utopian Command Council has requested that you develop training software for the training of compliance officers. Every utopian has a fixed home location that they are associated with. Every now and again they will leave that location and begin to wander randomly through the environment. The role of the compliance officer is to return the utopians to their home location before two utopians move to occupy the same square.



Home (white circle), Utopian (dark circle), Utopian at home (double circle), and Compliance Officer (triangle)

Your system will model the behaviour of both non-playable utopians and the playable compliance officer in a two-dimensional environment. Your logic must be placed in the `ShelterSpace` **`namespace`**.

Initialisation:

- The size of the environment, number of turns, and number of utopians are provided via the command line.
- Utopians are permanently associated with a home location
- Utopians begin in their home location
- The compliance officer begins randomly in one of the four corners of the world

Movement:

- Utopians may only move one step in the up, down, left, and right directions.
- The compliance officer may move one step in any direction.
- If a utopian is at home then there is a 10% chance they will move.
- If a utopian is not at home there is a 90% chance they will move.
- If a utopian moves into another utopian the game ends in failure
- If the compliance officer moves into a utopian then the utopian is moved back to their home location.

End-game:

- The game ends in defeat if a utopian moves into a space occupied by another utopian
- The game ends in victory if the specified number of turns pass

Design

The design must model the movement function. Make use of the design template provided in P01. Make sure to save your design as a PDF document.

Please format your practical for submission according to the structure established in the previous practical (`Docs` directory containing design, `Bin` directory containing executable files, `Source` directory containing `*.cpp, *.h` files and your `.cpb` file). The submission archive must be named according to the convention established in the previous practicals.

The marksheet is on the next page and is included only for self-testing purposes.

UNIVERSITY OF JOHANNESBURG

Mark sheet		
Competency	*Description*	*Result*
C0	Program Design	/10
C1	Boiler plate code • Standard namespace (1) • System library inclusion (3) • Indication of successful termination of program (1)	/5
C2	Coding style • Naming of variables (1) • Indentation (1) • Use of comments (1) • Use of named constants (1) • Program compiles without issuing warnings (1)	/5
C3	Functional Abstraction • Task decomposition (5) • Reduction of repetitive code (5)	/10
C4	Separate Compilation • Header file (1) • Guard conditions (2) • Inclusion of header file (1) • Appropriate content in header file (1) • Use of programmer defined namespace (5)	/10
C5	User Interaction • Menu System (5) • Appropriate use of input, output and error streams (5)	/10
C6	Command Line Argument Handling: • Appropriately overloaded main function (1) • Handling incorrect argument counts (1) • Use of supplied arguments (3)	/5
C7	Error Handling • Use of assertions (2) • Use of conventional error handling techniques (3)	/5
C8	Pseudo-random number generation (5)	/5
C9	Dynamically allocated two dimensional array handling with structures • Allocation (5) • Initialisation (10) • Deallocation (5)	/20
C10	Algorithm implementation • Logical Correctness (5) • Effectiveness / Efficiency of approach (5) • Correct output (5)	/15
B	Bonus	/10
Total:		**/100**