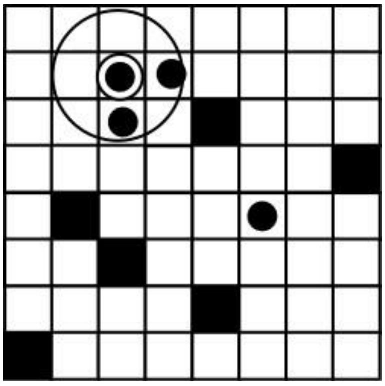
Practical 9 (due 2022-05-06 @ 09:00)

The purpose of this assignment is to introduce the type of problems presented during the semester test 2 and exam. 2018 ST2 PAPER A.

Please note: Submissions will be checked for originality. If you use someone else's code or code is taken from the Internet, then your prac will come under scrutiny for a potential copy, which may result in zero marks being awarded along with potential further disciplinary action.

Crime Scene Investigation

The Utopian Department of Justice needs you to create a serious game to spark interest in becoming a crime scene investigator. You will do so by creating a turn based game where an investigator must move around a crime scene and uncover clues.



Player (Double circle) Empty squares (space) Clues (black circles) Potential clues (filled squares) Detection range (large circle)

In the game you will need to move a player controlled character around a two dimensional playing area. Your logic must be placed in the `CrimeSpace` **namespace**.

Initialisation:

- The size of the environment, number of turns left, and number of clues are specified via command line arguments.
- The clues are spread randomly throughout the environment. For every clue which is placed an additional two potential clues must be placed in the environment. You must check that there is enough space in the game world to accommodate this. □ Initially clues and potential clues look the same
- The player is placed in a random row and column.

Moving:

- The player may move north (up), south (down), east (right), or west (left). The player may not move outside of the game area. The player may chose to investigate instead of moving.
- The player may not disturb the crime scene (they may not step on clues or potential clues).
- If the player chooses to investigate all of the potential clues in a one square radius disappear and those which were actually real clues are revealed (they are displayed as clues for the rest of the game).

End-game:

- The game ends in failure when the number of turns runs out and ends in victory when all of the clues are revealed.

The design must be based on the investigate function. Any submission that does not compile will be capped to 40%.

Original Marksheet

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 • Structures (5) • Allocation (5) • Initialisation (5) • Deallocation (5)	/20
C10	Algorithm implementation • Logical Correctness (5) • Effectiveness / Efficiency of approach (5) • Correct output (5)	/15
B	Bonus	/10
Total:		**/100**