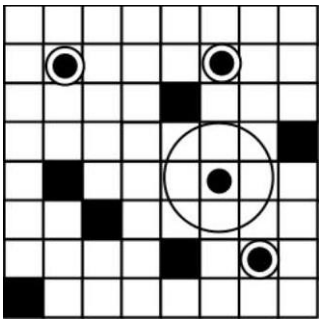
Practical 8 (due 2022-04-29 @ 09:00)

The purpose of this assignment is to introduce the type of problems presented during the semester test 2 and exam. 2017 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.

DARKNESS

The fearsome nocturnal Zoorkian Grue has escaped from the Utopian Zoological Gardens and is attacking people under the cover of darkness. The Grue is extremely sensitive to light and will not attack anyone who is holding a working torch, in fact the Grue will hide from sight as long as the torch is on. Unfortunately, the torches at the facility are of very low quality and each one can only last for 3 turns. When the torch is on the player can see the whole environment, when the torch is off they can only see in a one square radius. The player must avoid the pit-traps dug by the Grue and find extra batteries for their torch as they wait for the sun to rise.



Player (Black Circle) Empty squares (space) Batteries (double circles) Pit-Traps (filled squares) Norma/ visibility (large circle) In the game you will need to move a player controlled character around a two dimensional playing area. The player will need to collect enough batteries to survive until dawn. Your logic must be placed in the ZorkSpace namespace.

Initialisation:

- The size of the environment is specified via command line arguments.
- A command line argument provided number of batteries are spread throughout the environment.
- Every empty space has a 15% chance of containing a pit-trap.
- The player is placed in a random row and column but may not be placed on top of either pittraps or batteries
- The player starts with one battery
- The number of turns until dawn is calculated as being two times the number of batteries.

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 either turn their torch on or off.
- If the player moves on top of a cell containing a battery they pick it up and the number of turns of light they have available is increased by 3.
- If they do not have enough battery power their torch must be off.
- If the torch is on the player can see the whole environment as is safe from the Grue
- If the torch is off the player can only see in a one square radius and suffers a 20% chance of being attacked by the Grue.

End-game:

- The game ends when the sun rises, the player is attacked, or the player steps into a pit-trap. Consider the competencies as laid out in the mark sheet.

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

Original Marksheet

C7 will be evaluated differently. We will not consider assertions yet.

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