C++ Advanced – Exam Retake (15 Mar 2020)

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++11 standard.

Submit your solutions here: https://judge.softuni.bg/Contests/1813/CPlusPlus-Advanced-Exam-15-Mar-2020

Any code files that are part of the task are provided under the folder **Skeleton**.

Please follow the exact instructions on uploading the solutions for each task.

Task 3 – Snake

Your task is to write a program that implements the famous **Snake** game. Most of the game is already coded.
Your only role is to **provide an implementation for the Snake class**.

The input to the program consists of (in that order):

- The field (simple 2D array of chars) size will be read from the console. Field size is represented by ROWS and COLS variables.
- Starting position of the snake (startRowIndex and startColIndex)
- Number of following obstacles **N**
- **N** rows of obstacles (each containing obstacleRow and obstacleCol)
- Number of following powerUps **M**
- **M** rows of powerUps (each containing powerUpRow and powerUpCol)

The next part of the input consists of commands that your Game should execute:

```
enum InputCommands
{
    MOVE_SNAKE         = 0,
    GENERATE_OBSTACLE  = 1,
    GENERATE_POWER_UP  = 2
};
```

The `GENERATE_OBSTACLE` generates a new obstacle and run-time and `GENERATE_POWER_UP` respectively a power up at run-time. Both those command are already provided in the Skeleton.
You are **assured** that an obstacle or power up will **NOT** be generated on top of your snake at run-time.
Your role is to implement the functionality behind the `MOVE_SNAKE` command.

The `MOVE_SNAKE` command takes as argument a Direction.

```
enum class Direction
{
    UP    = 0,
    RIGHT = 1,
    DOWN  = 2,
    LEFT  = 3
};
```

Where your snake should move 1 tile up or 1 tile right or 1 tile down or 1 tile left (depending on the provided Direction).

SoftUni Foundation
© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.
Follow us:

The `MOVE_SNAKE` command can have different outcomes, which are described by the rules of the game (provide them as a return value of the method).

```
enum class StatusCode
{
    SNAKE_MOVING   = 0,
    SNAKE_GROWING  = 1,
    SNAKE_DEAD     = 2,

    STATUS_UNKNOWN = 255
};
```

- Return `SNAKE_MOVING` StatusCode - when the snake moves freely in the provided direction (which means no obstacles, powerUps or snake body parts should be present in the designated tile).
  Additionally, this tile **must be within the field boundaries**.

- Return `SNAKE_DEAD` StatusCode – when the snake would make an illegal move (hit an obstacle, hit it's own body part or goes out of field bounds).
  NOTE: when making a move **first move the head** of the snake and just **after that move it's remaining body parts.**
  If `SNAKE_DEAD` StatusCode is received – the game immediately is stopped (no further commands are executed).

- Return `SNAKE_GROWING` StatusCode - when the snake encounters a powerUp on the field.
  In that case the snake should grow it's body part by 1 node to it's tail(the previous snake last body position before the move).
  NOTE: every powerUp that is collected by the snake should be **removed** from the powerUps container!

On each iteration of the game – the Field is automatically printed for you.

The symbols used for markers can be found in **Defines.h**

```
enum FieldMarkerDefines
{
    EMPTY_FIELD_MARKER = '.',
    OBSTACLE_MARKER    = 'o',
    POWER_UP_MARKER    = '*',
    SNAKE_HEAD_MARKER  = '@',
    SNAKE_BODY_MARKER  = 'x'
};
```

As you can see the snake HEAD has a different print symbol than any other snake body part. This has nothing to do with the snake functionalities and only have the purpose to help you visualize the game field better.

Important reminder: You can make a helper functions in your Snake.cpp file that are not part of the class.

Restrictions

You should only submit **.h** and **.cpp** files compressed in a **.zip** archive.
There should be no folders in your **.zip** archive.

© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.
Follow us:

Examples

Input	Output
3 3 1 1 0 0 2 0 0 0 0	Printing initial Field state: ... .@. ... MOVE_SNAKE in dir: UP, status: SNAKE_MOVING Printing Field: .@. ... ... MOVE_SNAKE in dir: UP, status: SNAKE_DEAD
3 3 2 2 1 1 1 2 0 2 0 0 8 0 0 1 2 2 2 2 1 0 0 0 3 0 3 0 2 0 1	Printing initial Field state: *.* .o. ..@ MOVE_SNAKE in dir: UP, status: SNAKE_MOVING Printing Field: *.* .o@ ... GENERATE_OBSTACLE at row: 2, col: 2 Printing Field: *.* .o@ ..o GENERATE_POWER_UP at row: 2, col: 1 Printing Field: *.* .o@ .*o

SoftUni Foundation
© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.
Follow us:

	MOVE_SNAKE in dir: UP, status: SNAKE_GROWING Printing Field: *.@ .ox .*o MOVE_SNAKE in dir: LEFT, status: SNAKE_MOVING Printing Field: *@x .o. .*o MOVE_SNAKE in dir: LEFT, status: SNAKE_GROWING Printing Field: @xx .o. .*o MOVE_SNAKE in dir: DOWN, status: SNAKE_MOVING Printing Field: xx. @o. .*o MOVE_SNAKE in dir: RIGHT, status: SNAKE_DEAD
4 4 2 3 0 3 1 3 0 2 0 0 13 0 0 0 0 0 3 0 3 0 3 0 2 0 1 1 3 2 2 3 3 0 0	Printing initial Field state: *.*. ...* ...@ .... MOVE_SNAKE in dir: UP, status: SNAKE_GROWING Printing Field: *.*. ...@ ...x

© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.

0 1 0 1 0 1	.... MOVE_SNAKE in dir: UP, status: SNAKE_MOVING Printing Field: *.*@ ...x .... MOVE_SNAKE in dir: LEFT, status: SNAKE_GROWING Printing Field: *.@x ...x .... MOVE_SNAKE in dir: LEFT, status: SNAKE_MOVING Printing Field: *@xx .... MOVE_SNAKE in dir: LEFT, status: SNAKE_GROWING Printing Field: @xxx .... MOVE_SNAKE in dir: DOWN, status: SNAKE_MOVING Printing Field: xxx. @... .... MOVE_SNAKE in dir: RIGHT, status: SNAKE_MOVING

SoftUni Foundation
© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.
Follow us:

	Printing Field: xx.. x@.. .... GENERATE_OBSTACLE at row: 3, col: 2 Printing Field: xx.. x@.. .... ..o. GENERATE_POWER_UP at row: 3, col: 3 Printing Field: xx.. x@.. .... ..o* MOVE_SNAKE in dir: UP, status: SNAKE_DEAD

© Software University Foundation. This work is licensed under the CC-BY-NC-SA license.