Unit:
Designing and Developing Object-Oriented Computer Programs

Assignment title:
Sokoban

Sample Assignment

Important notes

- Please refer to the Assignment Presentation Requirements for advice on how to set out your assignment. These can be found on the NCC Education VLE. Click on Policies and Advice in the left-hand menu and look under the Advice section.
- You must read the NCC Education documents 'What is Academic Misconduct? Guidance for Candidates' and 'Avoiding Plagiarism and Collusion: Guidance for Candidates' and ensure that you acknowledge all the sources that you use in your work. These documents are available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- You **must** complete the **'Statement and Confirmation of Own Work'**. The form is available on *Campus*. Click on Policies and Advice in the left-hand menu and look under the Policies section.
- Please make a note of the recommended word count. You could lose marks if you write 10% more or less than this.
- You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media that cannot be run directly, will result in a fail grade being awarded for this assessment.
- All electronic media will be checked for plagiarism.

Introduction

A local software development company has decided to branch out into the development of a range of brain-training exercises designed to support cognitive development in the young and to prevent cognitive decay in the elderly. For the first of these they intend to develop a version of the puzzle game Sokoban, and they have handed the responsibility for its development on to you.

In Sokoban, players push crates around a warehouse full of impassable walls. The aim of each level is to get all the crates to suitable locations. These are indicated by the presence of special graphical icons onto which crates must be pushed. Each warehouse level is made up of a two-dimensional grid, but with a configuration that makes use of awkward angles and corners to provide the puzzle. The player is represented in the game by a particular graphic and is moved around the grid with the arrow keys or GUI buttons. When the player moves into the location of a crate, the crate is displaced in the direction of movement unless obstructed by a wall or another crate. Crates cannot be pulled – they can only ever be pushed.

Combined with the complexity of warehouse layouts, this makes the task of moving the crates to their destination an appropriate challenge for the target audience for the game. When all crates have been pushed onto their target squares, the level is completed. The game should keep track of when this happens, as well as when the level is an unsolvable state. This is defined as when at least one of the crates is not on its target square and there are no valid moves left to be made.

As part of the development of the game, you have been tasked to create THREE (3) different levels. Each of these should be more complicated than the last **and** each should come with a limit on the number of moves that are permitted. Failing to complete the level in the specified number of moves will count as a fail.

Each level aside from the first will also be expected to introduce a new feature to the game.

- Level two will introduce a one-shot pistol that will break the first crate in the direction specified by the player.
- Level three will introduce a one-time **phase** ability that permits a player to walk through a wall **once per level.**

The Wikipedia page https://en.wikipedia.org/wiki/Sokoban contains more information on how the game works, although these power-ups are unique to the version you will be coding.

You will need to create an application that implements a 2D grid of labels, each of which represent a space in the warehouse. All spaces in the grid are of equal size. You should work on the assumption of a 30x30 grid to give yourself sufficient space for the puzzles you will design. Each square of the warehouse can have the following contents, **and** you'll need graphics or symbols to represent each of these:

- The warehouse keeper
- Empty floor sections

- Impassable walls
- Target squares for crates
- Crates (on top of empty floor sections)
- Crates (on top of target squares)

They have already approached you with a list of required features for the first proof of concept build of the application, and these are as follows:

- Creation of an appropriate and reactive GUI
- Representation of warehouse levels with an appropriate data structure
- Responsiveness of the UI to movement commands
- Correct implementation of the movement of crates
- Correct tracking of steps and handling of step-related fail states
- Correctly identifying when a level has been solved and moving on to the next
- Correctly identifying when a level has been put in an unsolvable state
- Proper implementation of defined power-ups.

For testing, your employers have emphasised that they are most interested here in the correct identification of win and fail states.   Focus should be directed there, making sure all the usual test-cases for such things have been taken into account.

Task 1 – 50 Marks

50 Marks are available for a program which fully meets the requirements of the brief as outlined above.

Task 2 – 25 Marks

25 Marks are available for the selection of appropriate testing data along with a ONE HUNDRED (100) word report outlining the reasons for the selection of test data.

Task 3 – 25 Marks

25 Marks are available for the provision of a fully detailed class diagram.

Submission requirements

- Your program must be submitted as a zip file of the full project.
    - Whatever IDE you use, it should be possible to open and run the project directly from the extracted archive.
- Your testing data must be accompanied with a short, ONE HUNDRED (100) word discussion of how the data was selected and executed.

Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education documents ‘What is Academic Misconduct? Guidance for Candidates’ and ‘Avoiding Plagiarism and Collusion: Guidance for Candidates’ and ensured that you have acknowledged all the sources that you have used in your work? □

Have you completed the ‘Statement and Confirmation of Own Work’ form and attached it to your assignment? **You must do this.** □

Have you ensured that your work has not gone over or under the recommended word count by more than 10%? □

Have you ensured that your work does not contain viruses and can be run directly? □