|  |  |
| --- | --- |
| **Username:** |  |
| **Full Name:** |  |
| **Student ID:** |  |

Each section of this document can be extremely brief! Please just add short bullet points, not long explanations.

# Overview

Give a very brief summary (in bullet points) of what your program does. e.g. is it a fast game, a puzzle game, an editor of some kind, a display tool, etc. Basically, you are telling us what it SHOULD do, so that we can judge how well it does what it should.

<Insert your text here>

# Usage

Tell us briefly how to use your program. Do we just run it and it all works or do we need to click on anything on the screen, or press any key to make it do things? i.e. you are telling us how you would like it to be tested.

<Insert your text here>

# Known problems

Please mention any problems that you know about with your program. If you don’t mention them then we will assume that you do not know about them, and/or that your testing was insufficient.

<Insert your text here>

# Files

Please provide a list of the files which you added or modified in the following table, along with a brief statement of what they are used for:

Files which I added/are mine:

|  |  |
| --- | --- |
| **File name(s)** | **Purpose** |
| MyProjectMain.h | Main program sub-class |
| MyProjectMain.cpp | Main program sub-class |
| <new filename> | <Insert any extras into new table rows> |

If you had to change any of the base classes, provide details of the changes that you made and why.

Base class files which were modified, and why:

|  |  |
| --- | --- |
| **File name(s)** | **Changes and reasons** |
| BaseEngine.h |  |
| BaseEngine.cpp |  |
| DisplayableObject.h |  |
| DisplayableObject.cpp |  |
| FontManager.h |  |
| FontManager.cpp |  |
| JPGImage.h |  |
| JPGImage.cpp |  |
| TileManager.h |  |
| TileManager.cpp |  |
| MovementPosition.h |  |
| Templates.h |  |

# Specific requirements

Consider each of the requirements one at a time and give a brief (bullet-pointed) summary of how you have met the requirement, and why your implementation of it is so good.

If you have failed to implement a feature, or there are known problems with your implementation, then you should include the details here. That way we will know that your testing was not to blame. e.g. if you know something goes wrong under certain circumstances then please say so. Most professional applications have known bugs or problems so this is not a disaster, but you will lose marks for bugs and problems. If you do not mention it here, we have to assume that you thought that the features worked correctly so you will not only lose marks for features which don't work but will also lose marks for not testing it correctly.

For each requirement that you did implement, you should mention how it has been implemented.

This documentation should be in a format which will allow the person marking the coursework to easily identify the various functions (and data members, where applicable) that you added or modified in order to achieve the functionality. **Please be clear and concise rather than wordy. It will be quicker for you to write and quicker for us to read.**

Note: There is no need to explicitly attempt to use specific C++ features if they are unnecessary. E.g. you should not try to alter your program just so that you can put some exception handling in, to prove that you can do so. However, if using a C++ feature is the most appropriate way to handle a problem, you should consider highlighting that you used it (e.g. casting or exception handling) in the relevant sections for the features, or for the complexity or efficiency marking criteria. Your knowledge of these features will be tested in the exam, so I see no need for you to also prove this in the coursework.

## Draw an appropriate background.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Have moving objects.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Have interaction between the moving objects and the environment.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Provide player interaction.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Provide AI-controlled objects.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Save and load information to and from files.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Display status information on the screen.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

## Support different states.

What I did:

<Insert your text here>

Files/methods changed to do this:

<Insert your text here>

# Marking criteria

Since you know the marking criteria, you may want to make some comments about some of them, to point out something which you would like us to take into account in the marking. E.g. is there a requirement which you think was particularly well implemented?

## Functionality

What features do you want to draw especial attention to?

<Insert your text here>

## Complexity

What do you think was particularly complex about your program?

<Insert your text here>

## Compilation

Do you know of any issues with compiling your program?

<Insert your text here>

## Efficiency

What part(s) of your program do you think are especially efficient or inefficient?

<Insert your text here>

## Appearance

Is there anything especially good or bad about the appearance of your program, that you want to draw our attention to?

<Insert your text here>

# Additional information

In this section you should make any comments which are useful but did not fit the previous sections.

E.g. you may wish to make a comment about additional features which you added which you would like to have considered in marking, or a justification of why your implementation of something was particularly good.

Alternatively, you may wish to explain any unusual behaviour or problems that your program has, or anything else not mentioned in the previous sections.

Please do NOT mention anything which was already mentioned elsewhere.

<Insert your text here>