|  |  |  |
| --- | --- | --- |
|  | **Faculty of Computing, Engineering and Science** | Final mark awarded:\_\_\_\_\_ |

**Assessment Cover Sheet and Feedback Form 2017-18**

|  |  |  |  |
| --- | --- | --- | --- |
| Module Code:  CS2S566 | Module Title:  Tool Development for Computer Games | | Module Lecturer:  Ben Daubney, Gaius Mulley |
| Assessment Title:  Haunted Dungeons Level Creation Tool | | | Assessment No.  2 |
| No. of pages submitted in total including this page:  Completed by student | | | Word Count of submission  (if applicable): Completed by student |
| Date Set:  22-Jan-2018 00:00:00 | | Submission Date:  27-Apr-2018 23:55:00 | Return Date:  25-May-2018 23:55:00 |

|  |  |
| --- | --- |
| ***Part A: Record of Submission (to be completed by Student)*** | |
| **Extenuating Circumstances**  If there are any exceptional circumstances that may have affected your ability to undertake or submit this assignment, make sure you contact the Advice Centre on your campus prior to your submission deadline. | |
| **Fit to sit policy**:  The University operates a fit to sit policy whereby you, in submitting or presenting yourself for an assessment, are declaring that you are fit to sit the assessment. You cannot subsequently claim that your performance in this assessment was affected by extenuating factors. | |
| **Plagiarism and Unfair Practice Declaration:**  By submitting this assessment, you declare that it is your own work and that the sources of information and material you have used (including the internet) have been fully identified and properly acknowledged as required[[1]](#footnote-1). Additionally, the work presented has not been submitted for any other assessment. You also understand that the Faculty reserves the right to investigate allegations of plagiarism or unfair practice which, if proven, could result in a fail in this assessment and may affect your progress. | |
| **Intellectual Property and Retention of Student Work:**  You understand that the University will retain a copy of any assessments submitted electronically for evidence and quality assurance purposes; requests for the removal of assessments will only be considered if the work contains information that is either politically and/or commercially sensitive (as determined by the University) and where requests are made by the relevant module leader or dissertation supervisor. | |
| **Details of Submission:**  Note that all work handed in after the submission date and within 5 working days will be capped at 40%[[2]](#footnote-2). No marks will be awarded if the assessment is submitted after the late submission date unless extenuating circumstances are applied for and accepted (Advice Centre to be consulted). | |
| You are required to acknowledge that you have read the above statements by writing your student number(s) in the box: | Student Number(s):  15005216 |

**IT IS YOUR RESPONSIBILITY TO KEEP RECORDS OF ALL WORK SUBMITTED**

|  |
| --- |
| **Part B: Marking and Assessment**  **(to be completed by Module Lecturer)** |
| This assignment will be marked out of 100%  This assignment contributes to 50% of the total module marks.  This assignment is bonded |
| **Learning Outcomes to be assessed** (as specified in the validated module descriptor <https://icis.southwales.ac.uk/> ):  *1) To identify the functional and non-functional requirements of a game engine / game design 2) Apply relevant software engineering techniques to develop applications to generate data for use in a game engine* |

|  |  |  |
| --- | --- | --- |
| **Feedback/feed-forward** (linked to assessment criteria):   * Areas where you have done well: * Feedback from this assessment to help you to improve future assessments: * Other comments | | |
| **Mark:** | **Marker’s Signature:** | **Date:** |
| * **Work on this module has been marked, double marked/moderated in line with USW procedures.** | | |
| *Provisional mark only: subject to change and/or confirmation by the Assessment Board* | | |

|  |  |
| --- | --- |
| **Part C: Reflections on Assessment**  **(to be completed by student – optional)** | |
| **Use of previous feedback:**  In this assessment, I have taken/took note of the following points in feedback on previous work: | |
| **Please indicate which of the following you feel/felt applies/applied to your submitted work**   * A reasonable attempt. I could have developed some of the   sections further.   * A good attempt, displaying my understanding and learning, with   analysis in some parts.   * A very good attempt. The work demonstrates my clear   understanding of the learning supported by relevant literature and  scholarly work with good analysis and evaluation.   * An excellent attempt, with clear application of literature and   scholarly work, demonstrating significant analysis and evaluation. | |
| **What I found most difficult about this assessment:** |  |
| **The areas where I would value/would have valued feedback:** |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Fail | Narrow Fail | 3rd Class / Pass | Lower 2nd Class / Pass | Upper 2nd Class / Merit | 1st Class / Distinction |
| The tool must be able to load and display a level in its initial configuration 10% | * The level does not render and little attempt has been made to implement this functionality | * An attempt has been made to implement this functionality, however, very little of the game is rendered | * The majority of the level renders correctly, however, there are some substantial defects | * The level is rendered correctly except for some very minor issues * The code would need major changes to be of production quality | * The level is correctly rendered in it's initial state * The code would need minor changes for it to be of production quality | * The level is correctly rendered in it's initial state * The code is well written and documented |
| The tool must allow the user to save new levels that can be loaded into the Treasure Hunter game 10% | * Very little attempt at this functionality, beyond adding controls to GUI | * Minor attempt at functionality (e.g. controls added and some supporting code, but incomplete) | * Functionality partially works but is incomplete (e.g. not all changes saved) | * Level partially saves, but code has major issues or is of poor quality | * Functionality works, code is of good standard * Functionality has minor issues | * Functionality works and code is of excellent quality * User validation has been performed |
| The tool must allow the user to be able to define the amount of time a player has to complete the game 10% | * Very little attempt at this functionality, beyond adding controls to GUI | * Minor attempt at functionality (e.g. controls added and some supporting code, but incomplete) | * Functionality partially works but is incomplete (e.g. not added to CLevel instance) | * Functionality works, but code has minor issues or is of poor quality | * Functionality works and code is of good quality | * Functionality works and code is of excellent quality * User input has been correctly validated |
| The tool must allow the user to be able to add and remove fire or ghosts from the level by directly clicking on a tile 20% | * Very little attempt at this functionality, beyond adding controls to GUI | * Minor attempt at functionality (e.g. controls added and some supporting code, but incomplete) | * Functionality works to some degree though has significant problems (e.g. ghosts can be added but not removed) | * Majority of functionality works, though has major issues (e.g. screen doesn't correctly update following addition or removal of ghost) | * Functionality works apart from very minor issues | * Functionality works correctly * The code is of a high quality |
| The tool must allow the user to be able to change the appearance of each texture/icon 30% | * Very little attempt has been made at this functionality (e.g. controls placed on GUI, but no supporting logic) | * Minor attempt at this functionality | * Functionality allows a new texture to be selected, but board or level is not correctly updated | * Both a texture can be selected and the board is updated. however, code has significant issues e.g. not all textures can be correctly updated | * Both a texture can be selected and the board is updated. however, code has minor issues | * Both a texture can be selected and the board is updated * The code is of a high quality |
| The tool must allow a user to toggle a tile between floor and wall 20% | * Very little attempt made at this functionality | * An attempt has been made to deliver this functionality, however, the code is incorrect and does not work | * The toggle functionality has significant flaws | * The functionality has minor issues * Significant re-work would be required to release the code | * The toggle functionality works as expected * The code would require minor changes before release | * The toggle functionality works as expected * The code is well written and clear |
| To achieve the highest marks please check the XAML used in your solution. | | | | | | |

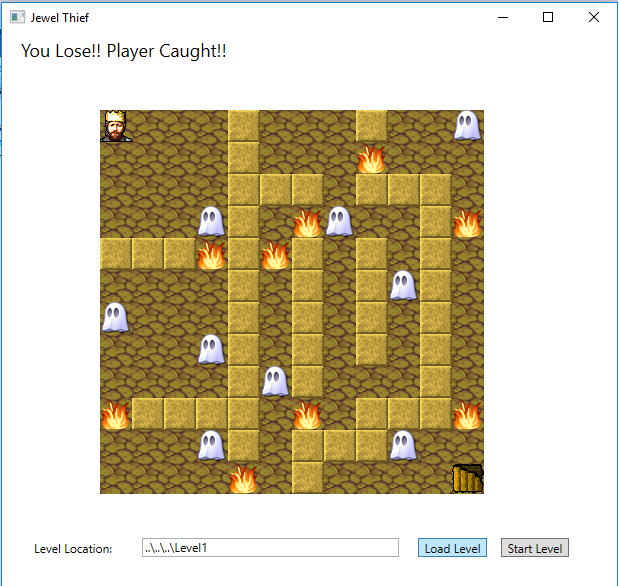
Assessment Task:

Haunted Dungeon is a simple 2D Tile Based Game. The objective of the game is to move the knight from the starting space to the stairs without making contact with a ghost. Each tile on the board can be either a wall or floor tile. Both ghosts and knights can occupy floor tiles and neither can occupy or move through a wall tile. Additionally a tile can be occupied by fire. A knight can move through and occupy this tile, however, ghosts cannot. This allows a level to be laid out as a maze or series of rooms.

A level must be completed in a set time.

The objective of this assignment is to design and implement a level creation tool. Levels can be created using your tool and subsequently be loaded into the Haunted Dungeon game.

A typical level may look as follows:



For the Haunted Dungeon game to be able to load a level it needs various files that represent what different objects look like and the level layout.

The list of files required are as follows:

Enemy.bmp A 32x32 pixel image representing an enemy.

Player.bmp A 32x32 pixel image representing a treasure hunter.

Goal.bmp A 32x32 pixel image representing the treasure.

Wall.bmp A 32x32 pixel image representing a wall tile.

Floor.bmp A 32x32 pixel image representing a floor tile.

Fire.bmp A 32x32 pixel image representing a fire icon.

Level.txt A text file describing the layout of the level (see details below).

These should all be stored in the same directory so that the Haunted Dungeon game can load them from the specified directory.

Level.txt

Level.txt is a text file that contains all the information about the level, e.g. the

positions of the walls and floors, the time to complete a level etc.

Parsers are provided to import and export levels correctly. Example levels are also

provided to get you started.

The requirements of the Level Creation Tool are as follows:

1. The tool must display how the level will appear in the game at its initial configuration, including the knight, ghosts, stairs and layout of walls and floor tiles.
2. The tool must allow the time permitted to complete the level to be set and updated.
3. The tool must allow a tile to be toggled between wall and floor by directly clicking on the tile the user wants to change. Any change should be reflected in the appearance of the board.
4. The tool must allow a designer to be able to change the texture used for each item in the game (i.e. change the texture used for the knight etc).
5. The tool must allow the user to set the position of ghosts by directly clicking on the tile the user wants to place a ghost.

5.1.If the user clicks on a tile that is not already occupied by a ghost a ghost should be added at the tile location.

5.2.If the user clicks on a tile where a ghost is already located it should remove the ghost from the tile location.

1. The tool must allow the user to set the position of fires by directly clicking on the tile the user wants to place a fire.

6.1.If the user clicks on a tile that is not already occupied by a fire a fire should be added at the tile location.

6.2.If the user clicks on a tile where a fire is already located it should remove the fire from the tile location.

1. The tool must allow all the data associated with a level to be saved to a directory in a format that can be read in by the Haunted Dungeon game.
2. The tool must allow an existing level to be loaded.

The source code for the Haunted Dungeon game will be provided along with examples of the levels. You are free to reuse or modify any source code from the Haunted Dungeon game for use in your level creator tool. This contains parsers that can be used for loading and saving level.

1. University Academic Misconduct Regulations [↑](#footnote-ref-1)
2. Information on exclusions to this rule is available from the Advice Centre at each Campus [↑](#footnote-ref-2)