MOD003310 Report  
  
For our MOD003310 assignment we have been tasked to create a digital version of the board game Descent: Journeys in the dark using the Unity game engine.

The aims of the project are to create an accurate representation of the reference game following the simplified ruleset that we have been given as well as adjusting the game to be fair and balanced based on the new ruleset.

In the game, a player will take the role of the ‘Overlord’ (Gamemaster) and is tasked to hinder the player as much as possible and prevent them from killing the Boss monster and leaving the stage from the exit gate. Each other player should be able to select a hero from a roster to use when fighting the Overlords minions. The difficulty and number of enemies that the heroes must fight is based on the number of players participating. The game also must be able to run smoothly on a GeForce GTX 760 graphics card, this is to ensure that the game has efficient coding and assets.

The game board the game takes place on is taken from an example map layout from the simplified ruleset booklet.

Our team consists of four members each responsible with a certain aspect of our project: the Team Leader: which primary role is to ensure the group stays on track and meet deadlines for the project, Lead Programmer: who is responsible for making sure the code is functional and efficient, Lead Designer: who makes sure that the game stays true and accurate to the physical game as well as change aspects of the games design to make them more enjoyable for the players, Lead Game Tester: who is responsible for making sure the game is functional and progresses in the way that is intended by planning and organising internal and external test sessions.  
  
A Gantt chart was used to organise the team and manage the work flow, this made sure that tasks were finished on time. We also used a variety of UML diagrams so that everyone knew how the game was supposed to be coded and how classes interact with each other. Our lead designer had a major role in making the UML diagrams and making sure they’re coherent and complies with the project brief.  
  
To store the project we primarily used GitHub, which is a development platform that allowed us to review code, manage the project and collaborate using different computers. We originally used Unity collaborate however it stopped their free service forcing us to find an alternative. This highlights the importance to have alternative forms of storage and communication to minimize the risk if one was to stop working. We did this by storing back up versions of the project on an external hard drive as well as on the GitHub platform, as well as having multiple ways of communicating to other members of the group by using discord and Trello as well as liaising weekly in person.

We have also incorporated databases into our project which is used to store the variables of the game such as monsters and players. This is helpful to us as developers as it allows us to easily make additions and make changes to the values either while developing the game or after the game has been release and changes need to be made as we find out more information on how to properly balance the game. we have decided to use SQL for the project instead of alternatives such as xml and Excel as “SQL is good for storage & searching. XML is good for transmitting & formatting.”(StackOverflow) And Excel while easy to use to store and browse data, “[is] not very good for **working with multiple datasets** in combination.” (SchoolofData) because of this, we chose to use SQL as it is able to handle tables in such a way that changing values can be done within the database editor and multiple tables can be stored which can hold user preferences which can be implemented into the game as custom games or player preferences. This also allows the player to send said preferences and customizations across computers either by transferring the save physically or through a cloud syncing network.

Our UI went through a number of changes throughout development. Initially we started with a basic form of UI consisting of simple buttons in the GUI which the player could use to select their action. This was confusing as the player did not know which character they were controlling nor what constituted a valid action. We resolved this issue by this by adding way of communicating to the player what is valid and what isn’t. For the movement, we implemented a renderer which changed the colour of the current title the player had their mouse over which changed hue according to if the space is a valid move or not, this gave a visual representation to the player which is more user friendly, as well as this we added a light to the current active player allowing the player to easily see which character is currently in play. We tried to make this intuitive as possible so that even a player that has no experience in the game will be able to understand what the GUI meant. The current player glow initially started as a green hue and we felt that this may be misinterpreted as the enemies would also glow green when it was their turn so we decided to change to a more neutral colour of yellow which is the same as the hue for the titles when it was on a valid space.

Discord usage:   
update log - <https://puu.sh/yEgKh/6672b1d61e.png>  
bug log - <https://puu.sh/yEgL5/1753723d17.png>

Info - <https://puu.sh/yEgLC/f6b3eea296.png> , <https://puu.sh/yEgQb/8831d6ebd0.png>

Trello - <https://trello.com/b/LgoWF0Vk/mod330010>

GitHub - <https://github.com/Dokitsu/ascension>

Gantt chart - <https://docs.google.com/spreadsheets/d/1UdWEFvgr55lTEyzZyg_c0QIEm0EWG-tYmcA6vygnp1k/edit?usp=sharing>

Reference:   
SQL over XML - <https://stackoverflow.com/questions/201568/when-would-i-use-xml-instead-of-sql>  
SQL over excel - <https://schoolofdata.org/2013/11/07/sql-databases-vs-excel/>