**Own Issue Project Proposal**

|  |  |  |  |
| --- | --- | --- | --- |
| Name : | *Hugh Thompson* | | |
| Provide an overview of what your program is intended to do. | The aim of the game is to make as much money(credits) as possible. This will be achieved by the player controlling two separate robots through a series of ships made up of procedurally generated rooms, beating enemies. If the entire ship is cleared of enemies, a monetary reward (in credits) is given, which can be used to upgrade the robots and their weapons to continue and try harder ships that earn more money if cleared. This will repeat theoretically infinite. | | |
| What are the steps the program needs to complete to be effective?  /  What are the specifications that need to be met? | 1. The player will be taken to the main menu screen where they may pick the which weapon they want for each of their two robots.    * **Weapons** will be loaded in from external storage and have different attributes (damage and fire rate) which will be displayed to the player so they may make an informed decision as to whether which weapon the will want to select.   Weapons may also be upgraded at this stage, this involves a cost (in credits) to upgrade the weapon, this will increase the damage of the weapon by an amount and reduce the fire rate so that the weapon may be more advantageous to the player. The cost of this upgrade will increase exponentially whereas the increase in weapon performance will be linear.   * + **Robots** will have attributes including speed, rotation speed, armour, damage taken and the weapon they are holding. Speed and armour may be upgraded using credits in the same exponential cost/linear upgrade fashion as above described and speed and rotation speed are upgraded together.  1. Once the player has selected their loadout and completed any upgrades they may want, they may press a button to begin play. On the press of that button, the player is presented with a choice of three ships, each with a difficulty rating based on the average of their robots’ levels plus or minus a randomly generated offset from -5 to 5, but where the difficulty may not be 0. This difficulty is used as a modifier to the generation of the ships later in the level generation step. 2. After selecting the ship they want, the ship will be loaded in.    * **Ships** are made up of a randomly generated number (from 5 -10) of rooms.    * **Rooms** are a randomly generated map(from 22 – 12 cells high / wide) of wall tiles(of 128x128 px size) arranged in a series of open spaces and corridors, with a 3x3 cell starting room and another 3x3 cell ending room at the top left and bottom right corner of the map respectively, each of these room will have a door on the outside edge of the map. 3. The two robots will be placed into the first room in the ship. The robots will be independently controllable one at a time using the W, A and D keys, with A and D being used to rotate the robot, and the W key to step forward in the rotated direction. The v key then can be used to switch between which robot the player is currently controlling. This will be a split screen game where screen follows one robot, and another screen is following the other one. 4. The player will then have to fight their way though the room, killing all the enemies. The amount of enemies will be spawned based on the difficulty of the ship(higher difficulty == more enemies).    * **Enemies** will have attributes including health, speed attack range, weapon held, type and level. These attributes will be loaded in from external storage based on the enemy’s type, and then modified by the level in a liner fashion. The level is set by the difficulty of the ship.   The player kills enemies by shooting them by rotating their currently controlled robot to face an enemy and firing a laser using the space key. If the laser hits the enemy, then the enemy will lose health, until their health is zero, whereupon they will die, being removed from the game. The enemies however also have the power to shoot at on of the player’s robots, if the robot is hit with a laser, then it will take an amount of damage based on the player’s armour and the damage of the bullet fired, if the amount of damage taken exceeds 100, then the robot will deactivate.   1. After all enemies in a room are killed, the door at the bottom right-hand corner of the map will open, allowing the player to advance to the next room in the ship if one of their robots enter though. The player may also choose to go back to the previous room by going through the door in the upper left-hand corner of the map, this door is open is ever room. 2. One of the following three conditions will happen to the player as they transverse the ship.    * If the player successfully navigates their robots through all rooms in a ship, then the player is returned to the main menu screen and a reward (in credits) is given, the amount of which will be random but scaled based of the difficulty of the stage completed.    * They player may wish to give up and return to the main menu, which they may do by returning to the first room and from there exiting though the door in the top left-hand corner of the map. No reward is given in this case.    * Both of the players robots are deactivated, the player will be shown a “you lose” screen and will be returned to the main menu from the last save point.   8 When the player is returned to the main menu, they will have the chance to repair their robots for a cost in credits, which exponentially increases the more damage taken. At this stage the game is also saved. The play then continues from stage 1, but they player must have at least 2 active robots to start another ship. | | |
| Identify the Complex Programming Skills you intend to include and how you intend to use them. | 1. *External libraries* | | |
| 1. *classes* | | |
| 1. *complex data structures* | | |
| 1. *persistent storage* | | |
| What will be the biggest challenges you will face with this project? | The room generateion algorithim.  The enemy “AI” algorithim.  Scope of project | | |
| Status: | Approved Rejected | | |
| Teacher: |  | Date : |  |