**Game Specification Form Student ID: \_\_\_hqfl21\_\_\_\_\_\_\_ Level 3**

|  |  |
| --- | --- |
| **Marking Criteria** | **Describe how your game matches the criteria** |
| **Game design (10%)** | |
| Game Goals: | Its year 2520, Durham university has crumbled to the ground and a cult guards the remains of the knowledge. As a rogue member of this cult, your goal is to kill their leader by collecting their treasure and knowledge pages to have enough to bribe the cult followers and enough power to defeat the leader.  Each level is completed by collecting that levels page and treasure through navigating the level and avoiding/killing enemies . |
| Game Type: | Scrolling 2D platformer with multiple levels. |
| **Core development (30%)** | |
| Game scene (visual representation [2D, 2.5D or 3D], internal data structure): | The game object stores the player object and lists of enemy objects, item objects and tile objects. The game object contains the game loop and updates the player and each enemy and item. The player stores its projectiles and messages in lists and updates them. Each enemy stores it’s messages and updates them. Both players and enemies store their own values for movement. |
| Game flow / game progression (e.g., navigation, screen scrolling, levels): | Game consists of 5 levels, each with 2 treasures to be found (knowledge page and treasure chest).  The screen scrolls with the player across the platform, navigation is done using the keyboard (W, A and D).  Levels are completed once both treasures are found, the next level is loaded and the player is moved to the start.  Game gets increasingly difficult, but your weapon and health also scale with it, ready for the boss fight. |
| Game interaction (e.g., action detection and response generation): | Event loop used to allow player movement and weapon usage based on user keyboard inputs. Enemies move according to the player. |
| Game object (e.g., use of sprite, 3D objects, animation, multimedia): | Each tile, enemy, item and the player are all draw in 2D and rendered accordingly. Animation is simulated by drawing different images when needed, where need is calculated via actions/events within the game.  Game sounds are stored and played at various stages of the game |
| **Game mechanics (30%)** | |
| Game rules / logics: | Each player, enemy, tile and item have their own hitbox which is used for collision detection. The player must navigate through each level and get the treasures. The tiles in the level map define where the player can go.  Player loses health when an enemy attacks them. If the player loses all their health, the level restarts and they lose any progress in that level. Player can shoot knowledge once they get the book in the library level Player upgrades damage when a knowledge page is received. Player gets health when red heart is collected. Player upgrades health when gold heart is collected. Player deals damage to enemies by firing knowledge. |
| Game challenges: | The navigation, enemy count, and enemy difficulty increases as the game progresses.  Final boss is the most difficult. |
| **Good use of game engine (15%)** | |
| Choice (pyGame, Unity): | pyGame |
| User input (keyboard, mouse, joystick): | Keyboard and mouse |
| Game object interaction (e.g., event triggering, collision detection): | Collision for player: with ground/walls, enemies and items  Collision detection for enemies: with ground/walls  Collision detection is done using pygame rects, with each item/enemy/player/tile having their own rect used as a hitbox.  Event triggering handles with pygames event.get() method with in the gameloop, triggering movement and weapons when keypresses are done |
| Incorporate multimedia content: | .png files store the sprites textures .wav files for audio |
| Other features used (e.g., asset, incorporation of external libraries): | Various python modules used within the game. |
| **Demonstrate creativity (15%)** | |
| Game economy (e.g., support to game type, game feedback, game difficulty): | Book of knowledge and health are increased throughout the game.  Sounds are played when various events occur.  Each level increases in difficulty reaching the final boss.  The player and enemies have health values which are modified accordingly. |
| Advanced Interaction (e.g., game physics, object tracking, steering behaviour): | Player and enemy have vertical/horizontal velocity which simulate gravity and allows for smooth jumping.  Enemies track the player with varying levels of complexity (some can only move side to side, some can jump, some can teleport) |
| **Game optimisation and configurability (50%) [For Level 4 Students Only]** | |
| Include optimisation to enhance game performance (e.g., game related functions, game scene and objects, interaction, rendering, media content): | - |
| Make the game flexible to support making changes (e.g., game scene and objects, game flow / progression): | - |