**Game Specification Form Student ID: \_\_\_\_\_\_\_\_\_\_\_\_\_ Level 3/4**

|  |  |
| --- | --- |
| **Marking Criteria** | **Describe how your game matches the criteria** |
| **Game design (10%)** | |
| Game Goals: |  |
| Game Type: |  |
| **Core development (30%)** | |
| Game scene (visual representation [2D, 2.5D or 3D], internal data structure): | 2D  In each level, platforms, enemies and items are randomly generated as you progress upwards. Only the first five platforms are hard-coded. Each level also contains unique platforms and unique music. |
| Game flow / game progression (e.g., navigation, screen scrolling, levels): | Infinite Screen Scrolling Upwards  5 Levels, each introducing new enemies and increasing in difficulty.  Player Navigation is done using left/right arrow keys and the keyboard  Menu Navigation is done using specific keys on keyboard as instructed on the screen |
| Game interaction (e.g., action detection and response generation): | Highscores and levels completed are saved  Using masks to have more accurate collisions against enemies  The following items spawn randomly on platforms:  Heart -> Gives you an extra life  Bomb -> Kills all enemies  Star -> Gives you immunity for a short while against enemies  Collision against enemies makes you lose a heart  Collision with items/treasure/enemies makes sounds  Collision with treasure can complete/end a level  Falling off the edge of platforms makes you lose the level |
| Game object (e.g., use of sprite, 3D objects, animation, multimedia): | Use of sprites and animations and music and vfx sounds |
| **Game mechanics (30%)** | |
| Game rules / logics: |  |
| Game challenges: |  |
| **Good use of game engine (15%)** | |
| Choice (pyGame, Unity): | pyGame |
| User input (keyboard, mouse, joystick): | keyboard |
| Game object interaction (e.g., event triggering, collision detection): | Collision detection happens between player and essentially all sprites in the game  Event triggering is used to navigate main menu / level selection and to control player during levels |
| Incorporate multimedia content: | Use of pixel sprites and music and vfx sounds and background images |
| Other features used (e.g., asset, incorporation of external libraries): |  |
| **Demonstrate creativity (15%)** | |
| Game economy (e.g., support to game type, game feedback, game difficulty): | Game difficulty is increased as you do higher levels in three ways:  New Enemies -> Later levels have enemies that are stronger/faster  Spawn Rate -> New and existing enemies spawn at faster rates in later levels  Level-specific movement: Slippery Ice platforms and strong sideways wind can make player movement harder on certain levels  To help with increasing difficulty, types of items and item spawn rate also increase in higher levels |
| Advanced Interaction (e.g., game physics, object tracking, steering behaviour): | Player steering incorporates velocity, acceleration and friction -> these values are adjusted on certain levels (like snow level where it is more slippery (less friction) or bridge level where there exists sideways wind  Player steering wraps around each level (going left off the screen makes you end up on the right side of the screen and vice versa) |
| **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): |  |