Resource – Feedback & Validation Checklist

**Overview:**

This is simply a checklist of things to ensure that you get external feedback and validation for this unit. You also store feedback in this document, producing a one-stop place for all of your validation and feedback for the project. Feel free to expand sections to accommodate as much text or imagery as needed – take space to keep the document clear and legible.

**PHASE 1 - START OF PROJECT**

You must go through these steps prior to developing your project.

**Concept Validation**

Explain your game concept and the required mechanics to script for it to peers and your teacher. They both provide feedback, and your teacher approves the project. This is prior to even planning scripts.

|  |  |
| --- | --- |
| **INITIAL PROJECT CONCEPT PEER VALIDATED BY 2 OTHER STUDENTS**   * They understand the concept, and the role of planned mechanics for this concept * They must flag up to three things they find confusing or problematic in your plans | |
| **Student Name 1:** <Kade F> | **Feedback:**   * Emphasise the cooldown as a punishment for spamming movement * “It looks fine” * Be careful of the scripting might be overly complicated |
| **Student Name 2:** <Ruby A> | **Feedback:**   * Make player die if they collide with a wall at certain speed * Make sure to detail to player the mechanics of game * Make hazards and environmental objects clear on what they do. |
| **INITIAL PROJECT CONCEPT APPROVED BY TEACHER, INCLUDING NOTES ON REQUIRED CHANGES** | |
| **Teacher Review Done:** | Yes |
| **Feedback Items:** | * Look at minigolf package. * Try to do a leader board. * Create a sandbox room to test mechanics. |

**Script Design & Planning**

You must design and plan scripts using pseudocode and flowcharts, and have these first reviewed by peers for issues, iterated upon, and then presented to the teacher.

|  |  |
| --- | --- |
| **SCRIPT DESIGN, PSEUDOCODE & FLOWCHART PEER REVIEW BY 2 OTHER STUDENTS**   * They understand the intended purpose and outcome of the scripts * They must flag anything confusing or problematic they find in your plans, pseudocode and flowcharts | |
| **Student Name 1:** <Kade R> | **Feedback:**  - Understood function of code  - Doesn’t see any major issues with the pseudocode |
| **Student Name 2:** <James W> | - Reads too much like actual code than pseudocode - should function properly. |
| **ITERATED DESIGN, PSEUDOCODE & FLOWCHARTS REVIEW BY TEACHER**   * The work has been iterated on after peer review by students. * The teacher confirms the iteration and approves plans – With possible further feedback | |
| **Teacher Review Done:** | Yes |
| **Feedback Items (if any):** | * Avoid writing Booleans as if they retain information, that’s a separate bool function. * Never have a binary decision point that has one outcome be empty. |

**PHASE 2 – IMPLEMENTATION & PRODUCTION**

These are feedback & validation requirements during production of your project.

**Environment Plans**

|  |  |
| --- | --- |
| **PRESENT ENVIRONMENT & ENVIRONMENT PLANS TO TEACHER**   * Itemise data sources being used in the environment prior to review * Present game environment in engine to teacher * Present gameplay plans within the environment, including the use of designed scripts | |
| **Itemise Data Sources used in Environment:** | * **EDS** – ProBuilder – Used in creating the game environments geometry. * **FDS** – PlayerControllerScript – Used to sling the player around, interacts with 2 other scripts for full script functionality (LineTrajectoryScript + EnergyMeterScript) |
| **Teacher Review Done:** |  |
| **Feedback Items:** | * Level plan is evident even without any questions or explanation – That is a really excellent outcome! * Final hurdle is overly mean, doesn’t look like the controller allows a lot of fine controls for tight fits. * For the efficiency obsessed, a ‘flicks’ counter would be good. * Level will teach the mechanics without need for a tutorial. Also ideal. |

**Script Implementation & Testing**

Once implemented, you must test your scripts, and once they are working have them reviewed by a peer, and the teacher.

|  |  |
| --- | --- |
| **PEER REVIEW BY 1 OTHER STUDENT**   * They understand the intended purpose and outcome of the scripts * They must confirm they see the scripts functioning in build, and the scripted code * They must provide feedback on implementation of code (efficiency/functionality) * They must provide feedback on clarity of code, naming and commenting practices | |
| **Student Name 1:** <NAME> | **Peer confirms scripts function, and review script code:** |
| **Feedback Items:** | * Item 1 * Item 2 * Item 3, etc |
| **SCRIPT IMPLEMENTATION REVIEW BY TEACHER**   * The teacher will review the build and script * The teacher will provide feedback on implementation, clarity of code, naming and commenting practices | |
| **Teacher Review Done:** | DT |
| **Feedback Items:** | * Really nice implementation. It felt extremely consistent, which is all important for a controller like this. |

**GUI Script Initial Review & Testing**

You must have your GUI scripting reviewed on initial implementation. Once

|  |  |
| --- | --- |
| **GUI SCRIPT IMPLEMENTATION REVIEW BY TEACHER**   * The teacher will review the build and GUI script * The teacher will provide feedback on implementation, and suggest at least one change | |
| **Teacher Review of Initial Implementation:** | DT |
| **Feedback Items:** | * Your UI is super clean, well done! This doesn’t really require heavy iteration, it’s already nice. * Maybe add ‘energy’ to the meter. Though it’s honestly pretty obvious right away due to it being an inevitable discovery. The main benefit might be people realising they spend varying amounts. * Maybe add an inputs counter. Either show at the end, or during play. * Maybe add a timer. This could absolutely become a competitive experience, for leaderboards or just beating your own records, etc. |

**GUI Script Iteration Review**

After you have finalised / iterated on your GUI from

|  |  |
| --- | --- |
| **GUI SCRIPT ITERATION REVIEW BY TEACHER**   * The teacher will review the build and GUI script * The teacher will sign off on review, confirming the GUI was iterated on appropriately | |
| **Teacher Review Done:** |  |