|  |
| --- |
| **Stimulus: *2D games with simple collision vector maths.*** |
|  |

Digital Technologies project: digital solution

**Section 1: Explore and Develop**

1. **Analyse**, **determine** and / or **symbolise** a plan for a game that utilises simple two-directional vector maths for collisions. In other words, not a platformer game.

|  |
| --- |
| * For the purposes of this exercise, any game that suffers from “platform stick”, “wall stick” or “collision stick” issues can be considered part of the *platformer* sub-genre (as this is what we are really trying to avoid here). * A “collision stick” issue is when the speed (horizontal or vertical) of an object creates an unwanted condition whereby the object “gets stuck” in another object (such as a wall, boundary or other object). |

Complete this task within a maximum of one A3 landscape page.

**Section 2: Generate and Evaluate**

1. Generate a prototype game that illustrates the plan determined in Section 1.

|  |
| --- |
| Some tips:   * Stay native to the game engine studied in class. Engineering is not about finding a perfect solution – it’s about finding the best solution with what is available. * Utilise [royalty free assets](https://opengameart.org/) if you wish. The focus is on gameplay. * As these games will be exported to HTML5 for web playback, it is recommended you aim for either a 640x480 or 800x600 viewport window. |

1. **/\*comment** at bottom of code in your main object in either STEP or CREATE event:
   1. Explanations of functionality
   2. Refinements
   3. Recommendations
   4. Testing
   5. Evaluation in terms of “***risk, sustainability and potential for innovation and enterprise*” justified by data (use discussion or data from previous points a-d)**

|  |
| --- |
| **Submission Requirements** |
| * Section 1: submit one digital document (maximum of one A3 landscape page) |
| * Section 2: submit entire project folder zipped (include evaluation in code comments) |
| **Important Notes** |
| * Use comments in code to explain understanding of programming structures, as well as pointing out refinements and on-going testing of code. |
| * Keep backups of your files. Save every 10-15 minutes of work. |
| * Final testing, refinements, evaluations and future recommendations should be neatly commented in an event in your most “significant” game object, e.g. **objPlayer.Step** |
| **Getting Started** |
| * Look at the examples of past completed assignments shown in class for inspiration. |
| * Look through past class notes, and resources from the website to help you plan. |
| * Brainstorm some ideas on an A3 sheet about games you like to play, and try to come up with a unique idea from these |
| **Authentication Strategies** |
| * Acknowledge any and all code snippets, tutorials, advice, information or help given. |
| * Students may be asked to explain their solution, or parts there-of, to determine authenticity. |

#### Appendix A: QCAA Years 9 and 10 Digital Technologies standard elaborations (contextualised)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | **A** | **B** | **C** | **D** | **E** |
| **Processes and production skills** | *Investigating and Refining* | *Section 1* | **purposeful definition and decomposition of complex problems in terms of functional and non-functional requirements** | **effective definition and decomposition of complex problems in terms of functional and non-functional requirements** | **definition and decomposition of complex problems in terms of functional and non-functional requirements** | **partial definition and decomposition of complex problems in terms of functional and non-functional requirements** | **fragmented definition and decomposition of complex problems in terms of functional and non-functional requirements** |
| *Generating and designing; producing and implementing* | *Section 2A* | **proficient** implementation of modular programs | **effective** implementation of modular programs | implementation of modular programs | **partial** implementation of modular programs | **fragmented** implementation of modular programs |
| *Evaluating* | *Section 2B* | **discerning evaluation of digital solution in terms of risk, sustainability and potential for innovation and enterprise** | **informed evaluation of digital solution in terms of risk, sustainability and potential for innovation and enterprise** | **evaluation of digital solution in terms of risk, sustainability and potential for innovation and enterprise** | **explanation of digital solution** | **description of digital solution** |

*This will be marked digitally via the submission platform.*

#### Appendix B: Australian Curriculum content descriptions

This assessment instrument is used to allow students to formally demonstrate the following Australian Curriculum Digital Technologies Years 9 and 10 Content Descriptions:

|  |  |
| --- | --- |
| **Explicitly measured** | |
| P&PS | Define and decompose real-world problems precisely, taking into account functional and non-functional requirements and including interviewing stakeholders to identify needs |
| P&PS | Implement modular programs, applying selected algorithms and data structures including using an object-oriented programming language |
| P&PS | Evaluate critically how student solutions and existing information systems and policies, take account of future risks and sustainability, and provide opportunities for innovation and enterprise |
| **Implicit to the task** (not formally measured) | |
| K&U | Analyse simple compression of data and how content data are separated from presentation |
| P&PS | Develop techniques for acquiring, storing, and validating quantitative and qualitative data from a range of sources, considering privacy and security requirements |
| P&PS | Analyse and visualise data to create information and address complex problems, and model processes, entities and their relationships using structured data |
| K&U | Investigate the role of hardware and software in managing, controlling, and securing the movement of and access to data in networked digital systems |
| P&PS | Design the user experience of a digital system by evaluating alternative designs against criteria including functionality, accessibility, usability, and aesthetics |
| P&PS | Design algorithms represented diagrammatically and in structured English and validate algorithms and programs through tracing and test cases |
| P&PS | Create interactive solutions for sharing ideas and information online, taking into account safety, social contexts, and legal responsibilities |
| P&PS | Plan and manage projects using an iterative and collaborative approach, identifying risks, and considering safety and sustainability |

**Key**:

K&U: Knowledge and Understanding

P&PS: Processes and Production Skills