|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **We are creating an online game** | | | | |
| **SOLO LEVEL** | **One** | **Many** | **Relate** | **Extend** | |
| **SOLO VERB** | ***Identify isolated skills*** | ***Describe and combine serial skills*** | ***Integrate skills*** | ***Evaluate skills*** | |
| **DECLARATIVE KNOWLEDGE** Knowing about (talking or writing about) the programming code  Creating a game requiring user input using visual programming language  Success criteria | I can **DEFINE** a problem identifying functional and data requirements  I can **IDENTIFY**  **…** the use of isolated visual programming skills in my digital game  For example, the use of:   * an if/then statement * loops or repetition * user input | I can **DESCRIBE** the programming sequence using the storyboard or flow chart; for example, the use of loops when:   * incorporating repeat instructions * allowing for varied user input * selecting options | … AND I can **EXPLAIN** HOW and WHY my programming choices, when integrating the different visual processing skills, meet the user input purpose of my digital game.  For example, explain how a logical sequence of visual programming skills codes for user input | AND I can **EVALUATE** the effectiveness of my game in meeting its functional requirements for:   * user input * game play | |
| **FUNCTIONING KNOWLEDGE** Knowing how to  Designing a game requires user input using visual programming language  Success criteria | I can use a storyboard to design a game and identify its functional and data requirements  I can create a digital game using a visual programming language **IF** I copy game programming examples created by someone else | I can independently create a digital game using a visual programming language BUT I am not sure about my programming and I struggle to debug any errors that occur | I can independently and confidently create a digital game using a visual programming language  AND I can debug as I build (correct my own code) | AND I can seek and act on feedback to improve the effectiveness of my programming choices as I go, or perhaps when building a game that incorporates user input  Page 1 of 2 | |
| **DECLARATIVE KNOWLEDGE** Knowing about  Designing a game requiring user input using visual programming language  Success criteria | I can **IDENTIFY** the needs of a user  eg a digital design solution – a user input interface   * empathise * define | I can **ELABORATE** on these needs by sketching out different options for the user input interface  I **ANNOTATE** each design to clarify the different options for the user input interface   * ideate | I can **BUILD** models or representations (prototypes) of the user input interface to learn more about the digital design solution   * prototype   For example, I can **SEQUENCE** (storyboard) the development of the user input interface  I can annotate the sequence to **EXPLAIN** how the prototype development ensures the user input interface better meets user needs | I can **TEST** the prototypes to make sure the solution will work as intended  I can **CREATE** an online game that incorporates the user input interface   * test   I can **EVALUATE** the effectiveness of the user input interface against clearly established criteria for the user’s needs | |
| **Digital technologies**  **Way of thinking** | **Design thinking** | **Design thinking**  **Computational thinking** | **Design thinking**  **Computational thinking** | **Systems thinking** | |

As learning progresses, it becomes more complex. SOLO stands for the Structure of the Observed Learning Outcome.  It is a means of classifying learning outcomes in terms of their complexity. It can help differentiate a task to enable students to operate at their level and provide learning tasks that are progressively more challenging.

**For more about SOLO Taxonomy refer to these websites**

[**John Biggs Solo Taxonomy**](http://www.johnbiggs.com.au/academic/solo-taxonomy/)

[**HookED: Solo Taxonomy**](http://pamhook.com/solo-taxonomy/)

Page 2 of 2

Page 2 of 2