Government of **Western Australia**
School Curriculum and Standards Authority

Sample assessment task

Year level	4
Learning area	Technologies
Subject	Digital Technologies
Title of task	Program my sprite

Task details

Description of task	Students will use programming skills learnt during previous lessons to design and create a simple computer game, using a free online website.
Type of assessment	Summative
Purpose of assessment	For students to explore programming a sprite to develop their computational thinking
Assessment strategy	Design plan and students' sprites
Evidence to be collected	Students will demonstrate their program working for the teacher to record against a checklist. Depending on the website used, teachers may wish to save the finished project to a USB or hard drive to review at a later date.
Suggested time	1 hour lesson

Content description

Content from the Western Australian Curriculum	***Processes and production skills*** **Digital implementation** Use simple visual programming environments that include a sequence of steps (algorithm) involving decisions made by the user (branching) **Investigating and defining** Define a sequence of steps to design a solution for a given task **Designing** Develop and communicate design ideas and decisions, using annotated drawings and appropriate technical terms **Producing and implementing** Select, and safely use, appropriate components and equipment to make solutions **Evaluating** Use criteria to evaluate and justify simple design processes and solutions

Task preparation

Prior learning	Students have participated in a series of lessons learning about basic programming in a visual programming environment, such as Scratch, Play Lab at Code.org, Tynker, etc. They have been taught to create a simple algorithm to make a character move and 'if' statements to change the character's behaviour, such as turning when they bump into the edge of the screen, or moving left and right when the arrow keys of the keyboard are pressed.
Assessment differentiation	Teachers should differentiate their teaching and assessment to meet the specific learning needs of their students, based on their level of readiness to learn and their need to be challenged. Where appropriate, teachers may either scaffold or extend the scope of the assessment tasks.

Assessment task	
Assessment conditions	Students will work individually on the task. They will use the worksheet to design the movements of their character before using the computer or iPad to code the movements.
Resources	• Computer or iPad for each student • Access to Visual Programming Environment, such as Scratch (website or offline version), Play Lab at Code.org website, Tynker app for iPad (or others students are familiar with). The programming environment should be familiar to students from previous lessons. • Worksheet for planning (attached) • *Optional:* Reliable internet connection if working online, if coding environment is online (e.g. Play Lab at Code.org) • *Optional:* Headphones

Instructions for teacher

NOTE: To complete this task, the teacher will need to be familiar with, and have worked with, a programming environment as well as taught the students to use it prior to attempting the task.

The task will be introduced and modelled by the teacher, explaining the purpose and expected outcome. If students are not able to save the program they have created, marking of the programming task throughout the lesson may be required or, alternatively, video recording of students' work.

Actual task instructions

- Planning worksheet given to students. Read instructions with the class and clarify any questions student have.
- On the worksheet, students draw a diagram and label it to show what they plan to achieve, as well as describe, using appropriate vocabulary what they will do.
- Once the planning is complete, students are to open the website Scratch (or offline Scratch environment if there is no Internet connection, Play Lab at Code.org website or Tynker app, as used in previous lessons) and log in if they have an account.
- Independently, students create a sprite (character) which responds to simple 'if' statements *according to their plan*.
- When complete, they will test their work and evaluate the success. Students should record if they had to make any changes to their plan and why these were made on their planning page.

Any worksheets or scaffolding specific to the task

Worksheet *(Final Task: Program a sprite to respond to a command)* is included for planning.

Alternately, students could work in a project/subject book or use loose paper to draw and write their plans, and evaluate the project.

The prompts on the worksheet may be listed on a board for students to follow if they do not use the worksheet itself.

Sample marking key

Description	Marks
Question 1	
Sprite responds to more than one command which matches the student's plan.	3
Sprite responds to a command, which matches the student's plan.	2
Sprite responds to a command, such as a key press or mouse click, and responds with an action. The command or action is different to that in the student's plan.	1
Subtotal	**3**
Answers could include, but are not limited to: • A correctly programmed sprite will complete a command, such as moving left or right, when the command is given. For example, when the right arrow key is pressed, the sprite moves to the right. For full marks, the sprite should complete more than one simple command, such as moving left and right with arrow keys. The command should also match the plans which the student has drawn and described. **Example: of what the allocation of the marks would look like to demonstrate the components that would give a student full marks in a question.**	
A student receiving one mark for this task could make the character follow a single command; for example, say hello when it is clicked with the mouse button. The plan the student drew and wrote about did not include this command.	[1]
A student receiving two marks will successfully create an algorithm to make the sprite complete a command; for example, say hello when it is clicked with the mouse button. This command was drawn and described by the student in the plan. The student may or may not have included other commands in the plan, which have not been implemented successfully.	[2]
To achieve full marks, the student will successfully create an algorithm, or more than one algorithm, to make the sprite complete a command; for example, move left when the left arrow is pressed, then right when the right arrow is pressed. These commands were drawn and described by the student in the plan.	[3]

Description	**Marks**
Question 2	
Provides a detailed drawing with labels, which accurately describe the commands the sprite will follow. When describing the steps, correct technical terminology will be included. The evaluation will accurately reflect the success of the programming in some detail.	5–6
Provides a drawing with labels to describe planned commands. Basic steps to describe the algorithm are included, with some technical vocabulary. The evaluation will reasonably reflect the success of the programming.	3–4
Provides a limited drawing to describe the planned commands. There may be attempts at simple labels. When describing the programming steps, the student will use simple language with little or no technical vocabulary. The evaluation contains little detail and/or does not accurately reflect the success of the programming.	1–2
Subtotal	**6**
Answers could include, but is not limited to: • The design of the program reflects the completed project and includes technical vocabulary. The evaluation is accurate and includes reasons, which references the testing of the program. **Example: of what the allocation of the marks would look like to demonstrate the components that would give a student full marks in a question.**	
A drawing of the character with simple labels.	[1]
Inclusion of technical vocabulary in labels, such as 'on mouse click' or 'sprite'.	[1]
Steps of how to create the algorithm are described.	
Detail and technical vocabulary is included in the planning, such as specific scripts. Eg 'when up arrow'.	[1]
An attempt to evaluate the program's success is included which states if the program worked according to plan or not.	[1]
Detail of the testing, reasons for success and/or changes to the plan which were made, as part of the evaluation.	[1]
Total	**9**