***Lesson 6 - Creating Video Game***

**Problem statement:**

Your team must design a new game and present your product to the leadership team of Games R Us, your goal is to get approval from the Games R Us executives to produce your game.

This lesson will teach students the basics of how to make a ball bounce using Scratch.

**Learning objectives:**

Students will work with a partner to design and create an interesting video game that has some requirements. As video game developers students will be given a variety of constraints to choose from to try and make their game profitable.

**Washington State Computer Science Learning Standards (3-5)**

1B-A-2-1

Apply collaboration strategies to support problem solving within the design cycle of a program

1B-A-5-3

Create a plan as part of the iterative design process, both independently and with diverse collaborative teams (e.g., storyboard, flowchart, pseudo-code, story map).

1B-A-5-4

Construct programs, in order to solve a problem or for creative expression, that include sequencing, events, loops, conditionals, parallelism, and variables, using a block-based visual programming language or text-based language, both independently and collaboratively (e.g., pair programming).

1B-A-3-6

Decompose (break down) a larger problem into smaller sub-problems, independently or in a collaborative group.

1B-A-3-7

Construct and execute an algorithm (set of step-by-step instructions) that includes sequencing, loops, and conditionals to accomplish a task, both independently and collaboratively, with or without a computing device.

**Next Generation Science Standards (NGSS)**

3-5 ETS1-1

Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost

3-5 ETS1-2

Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

3-5 ETS1-3

Plan and carry out fair tests in which variables are controlled and failure points are considered to identify aspects of a model or prototype that can be improved.

**Common Core State Standards (CCSS) Mathematical Practices**

1. Make sense of problems and persevere in solving them.

6) Attend to precision.

7) Look for and make use of structure.

8) Look for and express regularity in repeated reasoning.

**Soft Skills:**

**Partnership for 21st Century Learning (P21)**

* Think creatively
* Work creatively with others
* Reason effectively
* Use systems thinking
* Make judgements and decisions
* Solve Problems
* Communicate clearly
* Collaborate with others
* Create media products
* Apply technology effectively
* Adapt to change
* Be flexible
* Manage goals and time
* Work independently
* Be self-directed learners
* Interact effectively with others
* Manage projects
* Produce results
* Guide and lead others
* Be responsible to others

**Connections to career and educational pathways:**

**·** Software Developer/Programmer

· Computer Systems Engineering

· Web Application Developer

· Computer Science/Engineering

· Engineering Research & Development

· Game Developer

**Materials**:

* Computers
* Student’s Scratch account linked to teacher’s studio within Scratch (this allows for students to save the projects that they create)
* Engineering Journal
* Game Requirements worksheet
* Game Planning Sheet worksheet

**Lesson preparation:**

Students should create a free account on Scratch to be able to save their projects.

Teachers, if you have a Scratch Studio you should add another studio called video game. Similar to the studios that were created for 10 Block Challenge (lesson 3) and Bouncy Ball (lesson 4).

If you feel that your students need more understanding of the coordinate grid used for the Scratch stage see this tutorial within Scratch.

<https://scratch.mit.edu/projects/23569571/>

**Time required:**

60 minutes (could take longer depending on complexity of game)

**Grouping of students for instruction:**

Students are continuing to work with their same partner that they had from the lesson before, Game Research Day lesson.

**What is the instruction?**

Students will be planning out specifics of their video game on their planning sheet. Students be using their knowledge of Scratch coding and create their own video game without using any kind of remixing from other games.

**Lesson:**

1 - Present the final challenge to students. Your team must design a new game and present your product to the leadership team of Games R Us, your goal is to get approval from the Games R Us executives to produce your game.

2 - Use the Game Requirements worksheet to go through the requirements for their video game. Also let students know that production costs are set up that each piece of code costs money. Not only to take time to place each piece of code but also the more code blocks you have in your game the more chances of possible error. The other constraints to present to students are possibilities for the developer to include or not which will help lower their production cost.

3 - Teacher should check off students planning sheet before starting on the computer.

4 - Students proceed to create their video game on Scratch. If students get stuck and don’t know how to code something they have a few different resources to use. Students can ask students in the class for help, open a second tab of Scratch on their computer and find a similar game to study the coding sequence. Students can also access the question on the right side of Scratch to get some specific tutorial help. Scratch Wiki is another place on the computer to get questions answered. Any additional resources that the teacher has could help like Scratch Coding book or CS First booklets (see teacher resources list for specifics)

5 - Students present their video game to the class. Teachers can use the rubric for assessing and grading video games that are created. It would be nice if time allows to have students play each other’s games too.

**Accommodations:**

Any students that are physically impaired can partner up with another student to be able to be successful and follow along with the lesson. To make the team stronger you could make it a triad instead of partners.

**Extensions**:

Students that finish early can add more detail and layers to their game.

**Assessment**:

Assessing takes place within the student engineering journals and how they have answered the questions within the lesson and also their reflection question.

1 Torp, L., & Sage, S. (2002) Problems as Possibilities: Problem Based Learning for k16 Education (2nd ed.). Alexandria, VA: Association for Supervision and Curriculum Development