**Remix and Unplugged Math Facts Lesson**

Focus: In this activity guide or lesson plan there are two main goals.

The first is to review using an actual project some possible ways to use the project with your students. Why would you want to have different ways to play/remixes? They can meet the needs of students who have varying coding experiences, from just beginning to more exposure. This can mean one remix for your whole class or an ability to differentiate amongst your students in one class.

The second is to present an activity to help your students “see” a connection from an unplugged task to the coding task. Each time we can “break down” a task for coding we are developing our students’ ability to develop algorithms for problem solving.

There are a variety of four levels to play/remix this one Scratch project.

The **first**, and the one that requires and inspires the **least understanding** is to **simply play it**. Put two numbers in and the sum of those two numbers and if you are correct the score will increase. Repeat this five times and the project ends.

The **second** level is to **remix a portion of the coding**. This idea is one of the tenets of Scratch and is encouraged at all levels. So for this project the original coding could be remixed any number of ways.

Here are 4 possible remixes:

 

Remix #1 Changes the number of questions from 5 to 10; Remix #2 Changes the range of numbers to add from 10 -20 to 1-50; Remix #3 Changes the score points from 1 to 2.

I could also remix the operation from addition to subtraction …

  

 Remix #4 Make a new variable called Difference, make the set of random numbers greater than the other, change the operator block to subtraction, change the If block so the answer is the Difference between the two random numbers and chance the text in the blocks to reflect a subtraction question.

<https://scratch.mit.edu/projects/131536576/>

 

The **third** level is for our students to **construct the coding train based on their understanding/plan** of what they want the outcomes of the coding to be. The following is a guide to this task.

Introduction: Act out the coding of this project as an unplugged activity. You might want to divide your class up into groups of 4 or 5

Supplies two sets of cards with the numbers from 10 to 20; one set of cards with the numbers from 1 to 5; and a calculator

Roles: One Student (or Teacher) Robot; One Student Calculator; One Student Score Keeper; One Student Coder; and One Student Player

The activity should unfold in this fashion:

The Coder to give the Robot instructions to pick up two random cards and show them to the Calculator and the Player.

The Coder tells the Student Player to say the sum of these two numbers.

The Coder is to give instructions to have the Calculator add them together.

The Coder will give the Calculator instructions to determine if the Player’s sum matches the Calculator’s sum. If yes, The Coder will instruct the Score Keeper to increase the score by 1.

If no, The Coder will instruct the Score Keeper not to increase the score by 1.

The Coder will repeat the instructions.

Problem you might encounter: Students can give either too few or too many instructions to the robot. For example, saying “move hand up” can have the robot move their hand above their head or barely move it at all – both situations do follow the instructions and this can sometimes derail the lesson objective of breaking apart the sections so a block or blocks can carry out each step. If this might be the case then I suggest the teacher be the robot and the rest of the roles be students.

After the role playing has been completed review with the class the instructions given and write them down. If the students are familiar with the blocks then they can suggest which ones will carry out the instructions needed. If the students are not familiar with the blocks then have a Scratch screen projected and browse through them as needed.

Here is a possible list of instructions:

Provide instructions on how to play the game

Randomly pick a first number between \_\_ and \_\_

Randomly pick a second number between \_\_ and \_\_

Ask the Player to add the two numbers together and say the sum

Calculate the sum of the two numbers

Decide if the Player’s sum is correct

If correct add a point to the Player’s score

If not correct tell the Player and do not add a point to the score

Repeat process \_\_ times

At this point depending on the experiences of your class you can project a Scratch screen and build the blocks with their directions or the students can go into Scratch and code the blocks in pairs or individually.

**Ontario Mathematics Curriculum**

NSN Overall Expectations

By the end of Grade 4, students will:

• solve problems involving the addition, subtraction, multiplication, and division of single- and multi-digit whole numbers, and involving the addition and subtraction of decimal numbers to tenths and money amounts, using a variety of strategies;

*Operational Sense*

By the end of Grade 4, students will:

– add and subtract two-digit numbers, using a variety of mental strategies (e.g., one way to calculate 73 – 39 is to subtract 40 from 73 to get 33, and then add 1 back to get 34);

– solve problems involving the addition and subtraction of four-digit numbers, using student-generated algorithms and standard algorithms (e.g., “I added 4217 + 1914 using 5000 + 1100 + 20 + 11.”);

– multiply to 9 x 9 and divide to 81 ÷ 9, using a variety of mental strategies (e.g., doubles, doubles plus another set, skip counting);

– solve problems involving the multiplication of one-digit whole numbers, using a variety of mental strategies (e.g., 6 x 8 can be thought of as 5 x 8 + 1 x 8);

– multiply two-digit whole numbers by one-digit whole numbers, using a variety of tools (e.g., base ten materials or drawings of them, arrays), student-generated algorithms, and standard algorithms;

– divide two-digit whole numbers by one digit whole numbers, using a variety of tools (e.g., concrete materials, drawings) and student-generated algorithms;