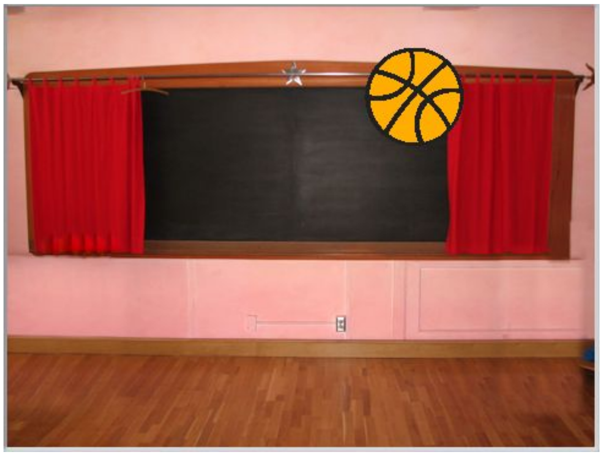
Scratch Game - Math Quiz

Description

This is a scratch game that challenges the player to a series of multiplication questions. They have to provide the correct answer before a dropped Ball reaches the bottom. The catch is, as they play the Ball drops faster and faster.



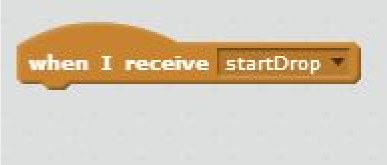
Walkthrough

Delete the “Scratch” character sprite, add a new sprite from the library and choose the item that will drop (for example the “baseball” from the “Things” folder), from now on I will call this the “Ball” sprite.
Click on the Stage, select “Backgrounds” and choose a suitable stage.

From the “Variables” category create the following 5 variables:
level - This is the game level and will control how fast the Ball drops.
score - This is the player’s score.
playing - This indicates when the game is running.
first - this is the first number to multiply.
second - this is the second number to multiply.

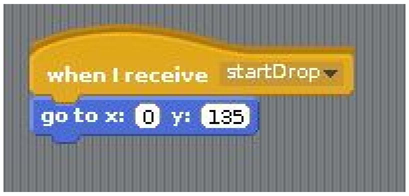
1. Create the Ball Drop Sequence.

First we are going to create the Ball drop sequence:
Select the Ball sprite, and check that the “scripts” tab is selected.
From “Control” category, select the “when I receive ...” block. Then select the drop down and enter a message of “startDrop”.



```
when I receive startDrop
```

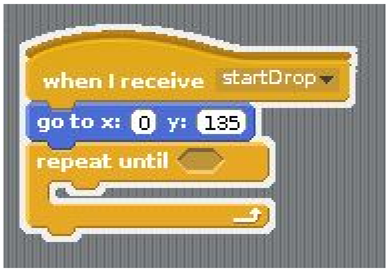
Move the Ball so that it is very near the top of the “stage”, but not touching it.
From the “Motion” category, drag “go to x: ... y: ...” and lock it below the event placed above (your x: and y: might be slightly different to mine).



```
when I receive startDrop
go to x: 0 y: 135
```

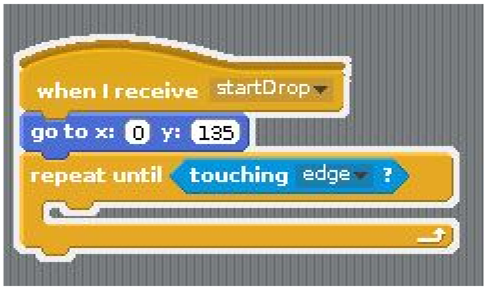
We want to move the Ball down the screen until it reaches the bottom. We will use a “forever if” loop to do this.

From the “Control” category, drag “repeat until …” and lock it below the “go to …” block.



```
when I receive startDrop
go to x: 0 y: 135
repeat until
```

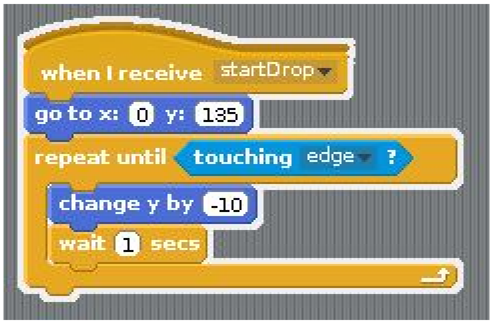
From the “Sensing” category drag “touching …” and place it in the condition area of the “repeat until ... “ block. Select “edge” from the drop down list of the “touching …” condition.



```
when I receive startDrop
go to x: 0 y: 135
repeat until touching edge ?
```

From the “Motion” category, drag the “change y by …” block inside the repeat loop. Change the value to “-10” (this is how far the Ball will move down each time around the loop).

From the “Control” category, drag the “wait 1 secs” block and place it inside the repeat loop after the “change y by -10” block.



```
when I receive startDrop
go to x: 0 y: 135
repeat until touching edge ?
  change y by -10
  wait 1 secs
```

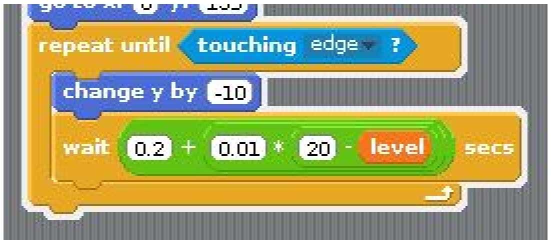
To calculate the speed of the Ball we will use the following formula:
0.2 + (0.01 x (20 - “level”))
This will make the Ball drop faster as the level increases.

From the “Operators” category:

Drag “... + …” and use it to replace the (1) of the “wait … secs”.

Now drag “... * …” to the second part of the “... + …”.

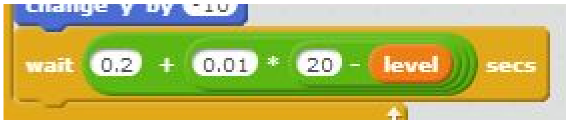
Finally drag “... - …” to the second part of the



```
repeat until touching edge ?
  change y by -10
  wait 0.2 + 0.01 * 20 - level secs
```

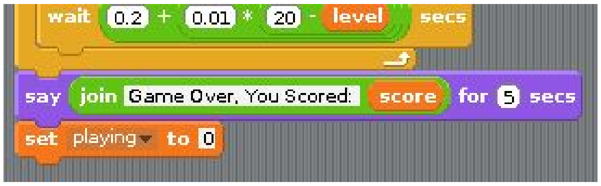
“... * …”.

Now fill in the values for our formula above. (You might want to try changing these values later to make the game easier or more difficult.) These are all actual numbers except for the “level” variable that you need to get from the “Variables” category.



```
change y by -10
wait 0.2 + 0.01 * 20 - level secs
```

If this “repeat until ... ” loop finishes then the Ball has reached the bottom and the game is over.
From the “Looks” category, select the “say … for 2 secs” block and lock it after the repeat loop, change the “2” to a “5”.



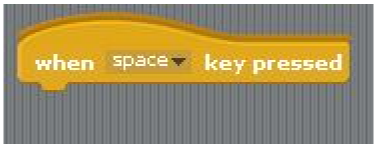
```
wait 0.2 + 0.01 * 20 - level secs
say join Game Over, You Scored: score for 5 secs
set playing to 0
```

Then from the “Operators” category, drag the “join hello world” and place it inside the “Say” block. Replace the “hello” with “Game Over, you scored: “. Replace the “world” with the “score” variable from the “Variables” category.

From the “Variables” category drag “set ….to …” and lock it under the “say …” block. From the drop-down select “playing”, leave the value as “0”. This will indicate that the game has finished.

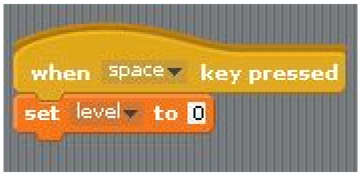
2. Test the Ball Drop Sequence.

From the “Control” category, drag “when space key pressed” to the script area.



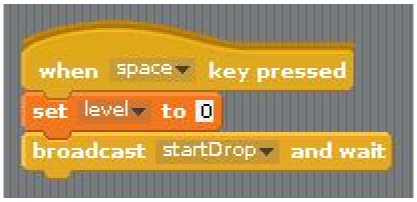
```
when space key pressed
```

From the “Variables” category select “set … to …” and lock it below “when space key pressed”.
From the drop-down select “level”, leave the value as 0 (this represents the starting level).



```
when space key pressed
set level to 0
```

Now, from the “Control” category, drag “broadcast startDrop and wait” and lock it under the “set level to 0” block.



```
when space key pressed
set level to 0
broadcast startDrop and wait
```

Now pressing “space” will run your Ball drop sequence. Try it out, the is the slowest speed. Will you be able to answer a maths question before it gets to the bottom ?

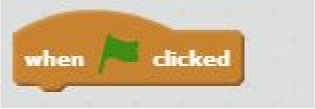
Now change the value of the “set level to 0” block to “20” (this represents the highest level, the fastest Ball drop) - Do you still think that you’ll have enough time ? Well, you’ll find out later.

Q1: Can you see how to change the code so that the Ball starts slower at the lowest level ?

Q2: Can you see how to change the code so that the Ball goes even faster at the highest level ?

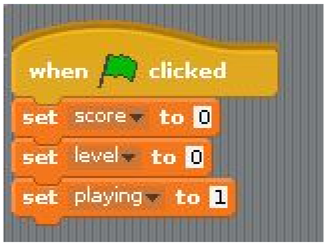
3. The Math Question Sequence

Start a new script by dragging the “when FLAG clicked” event from the “Control” category to the script area.



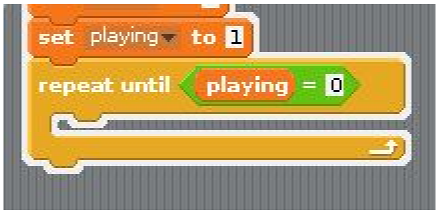
```
when ⚑ clicked
```

We need to start by resetting the variables. From the “Variables” category drag three “set …. to …” blocks under the “when FLAG clicked” block. From the drop down change the first one to “score” and the second one to “level, leave both of these at zero. Set the drop down of the third one to “playing” and set it to “1” (this indicates that the game is active).



```
when ⚑ clicked
set score to 0
set level to 0
set playing to 1
```

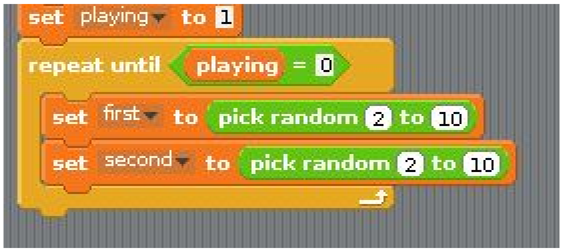
Now drag a “repeat until …” block and place it below the variables. This is the main loop for each question.
Drag a “... = ….” block from the Operators category. From the Variables category, drag the “playing” variable to the left hand side of the equals operator. Set the right hand side to “0”. Note that we have just set playing to “1” and will keep our quiz running until playing = “0”.



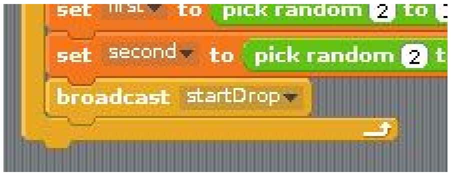
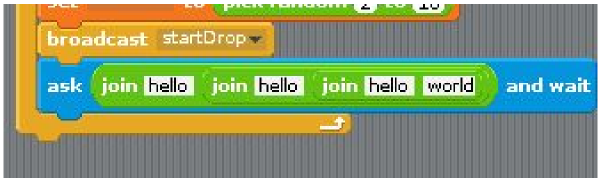
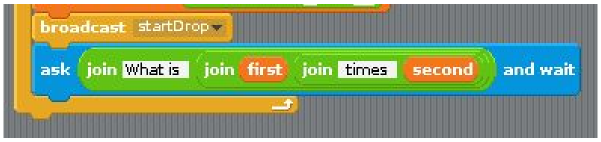
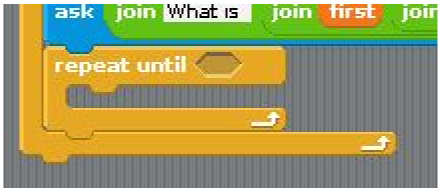
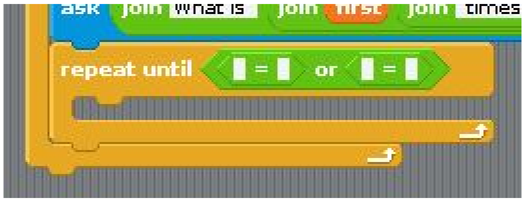
```
set playing to 1
repeat until playing = 0
```

Q3: Where did we set playing to zero ? What is the effect of this ?

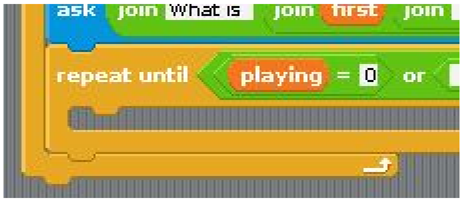
We now want to choose two numbers for our quiz question. Drag two “set ….. to …” blocks and place them inside the “repeat until loop”. Select “first” in the first drop down, and “second” in the second drop down. Now drag a “pick random … to …” from the Operators category to each “to” value in those blocks. Set the values to “2” and “10” for each “pick



```
set playing to 1
repeat until playing = 0
    set first to pick random 2 to 10
    set second to pick random 2 to 10
```

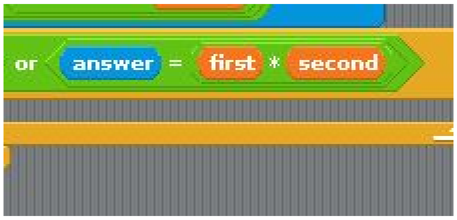
random” (“1” would be too easy).	
Now drag a “broadcast ….” from the “Control” category and select “startDrop” in the drop-down. This will start the Ball falling!	 set second to pick random 2 t broadcast startDrop
Now drag a “ask … and wait” block from the Sensing category and place it under the broadcast message. From the “Operators” category drag a “join …. ….” block and place it in the “ask” block, then place another “join …” in the right hand side of the first join, and yet another “join …” in the right hand side of that one. We can now join four things together.	 broadcast startDrop ask join hello join hello join hello world and wait
Set the first “join” cell to the words “What is “ (put an extra space after the word “is”). Drag the variable “first” to the next cell of the join. Then put the word “ times “ in the third cell (with a space before and after it), and finally drag the variable “second” to the last cell.	 broadcast startDrop ask join What is join first join times second and wait
This script will now wait for the player to answer the question, but remember that the other script is continuing to make the Ball drop! Time will be running out for the player.	
We need to keep asking the play for an answer until they get it right, so drag another repeat loop and place it under the “ask” block but inside the existing “repeat until …” loop. There are two reasons why we must stop this loop; either the player is out of time, **or** the player has got the correct answer.	 ask join What is join first joi repeat until
Drag an “.... or ….” block from the Operators category and place it in the “repeat until” condition box. Now drag an “... = …” block into each side of the “... or …” block.	 repeat until = or =

For the first reason to stop this repeat loop we want to check for when the “playing” variable is “0” (zero).



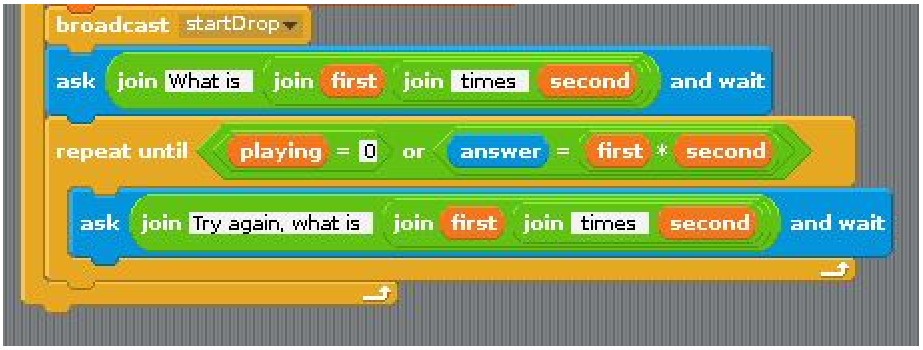
```
ask join What is join first join
repeat until playing = 0 or
```

For the second reason to stop this repeat loop we want to check when the answer is correct.
Drag the “answer” element from the “Sensing” category to the left side and drag a “... * …” element from the Operators category to the right side. Put the “first” and “second” variables in the “.... * ….” element.



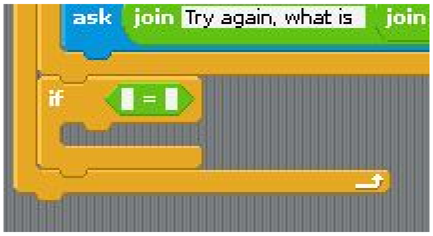
```
or answer = first * second
```

When the player gets the answer wrong we want to tell them and ask them the question again. Right click on the “ask..” section that you’ve already created and select “duplicate”. Place it in a clear area. Note that this has also duplicated the “repeat loop”, which we don’t want, so detach that and discard it. Change the text in the first cell so that it reads “Try again, what is “, now place it inside the repeat loop. Our program will now keep asking until either the time runs out, or gives the correct answer.



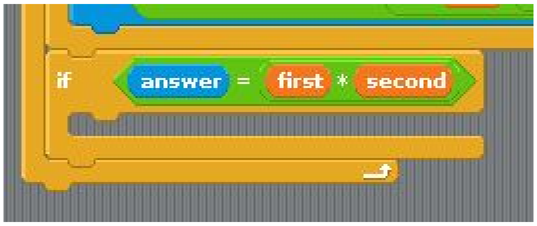
```
broadcast startDrop
ask join What is join first join times second and wait
repeat until playing = 0 or answer = first * second
    ask join Try again, what is join first join times second and wait
```

Now we need to check to see if the player has got the answer right. From the Control category drag an “if … then” block and place it under the inner repeat loop, but inside the outer repeat loop. From the Operators category, drag a “... = ...” block into the “if” condition.



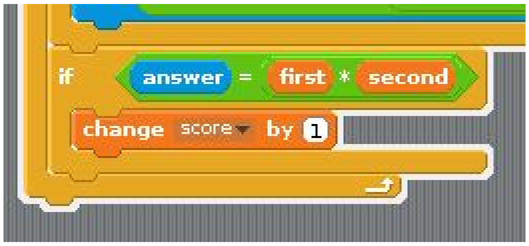
```
ask join Try again, what is join
if [] = []
```

From the Sensing category drag the answer into the left side of the “... = …”. From the Operators category drag a “... * …” into the right hand side. Now from the variables place “first” and “second” in either side of the multiple block.



```
if answer = first * second
```

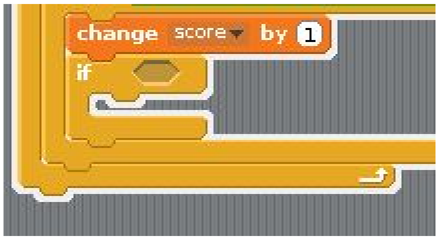
If the answer is correct, then we want to increase the score. From the Data category, drag a “change … by …” inside the “if” block. select “score” from the drop down and set the amount to “1”.



```
if answer = first * second
  change score by 1
```

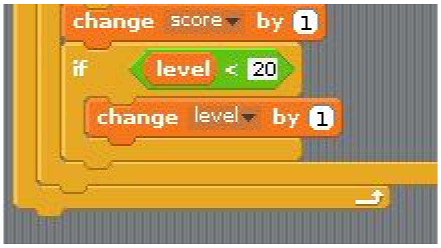
Q4: Can you see how to give the player more points the higher the level they have reached ?

We also need to increase the level (which will make the Ball drop slightly faster), but only up to the highest level possible. From the Control category drag an “if … then” block and place it under the “change score …” block, but still inside the existing “if” block.



```
change score by 1
if
```

The level can only be increased to a maximum value of 20, so from the Operators category drag a “... < …” to the “if” condition. From the Variables category drag the “level” variable to the left hand side of the “... < …” and set the right hand side to “20”. From the Variables category, drag a “change … by …” inside the “if” block. select “level” from the drop down and set the amount to “1”.



```
change score by 1
if level < 20
  change level by 1
```

You can now test your game, what score can you get ? How can you make the game easier or more difficult ?

4. Further Activities:

Add sounds when the player gets a question wrong, right, and loses the game.

Change the game so that it asks you a different maths question (for example to sum three numbers)