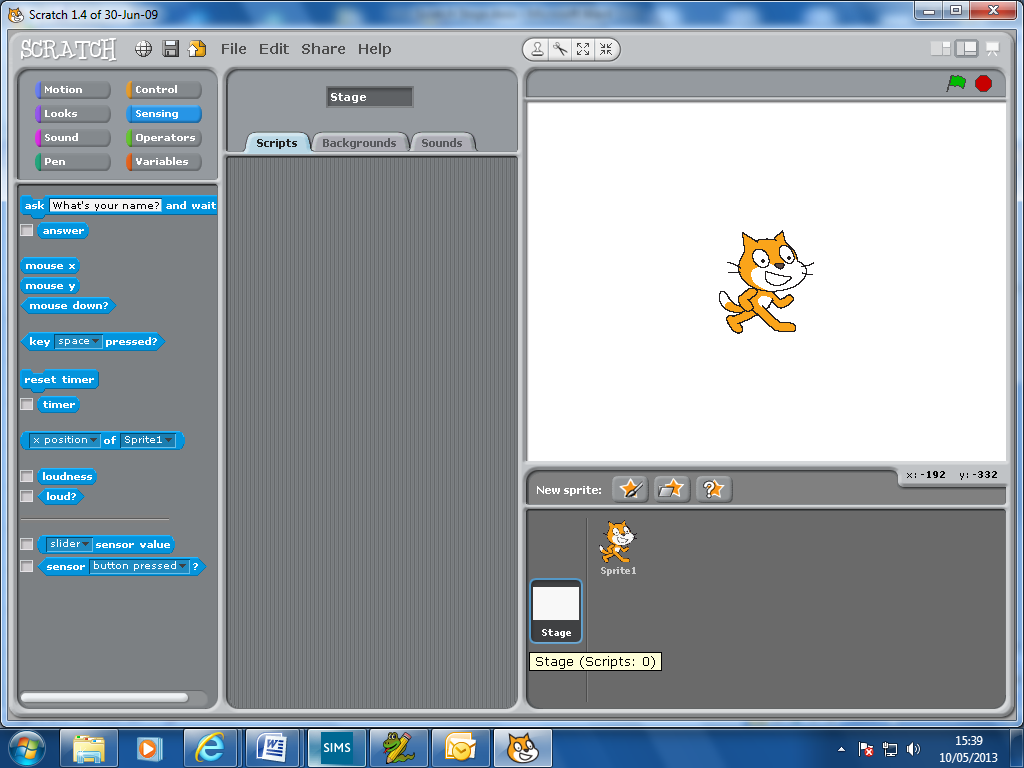
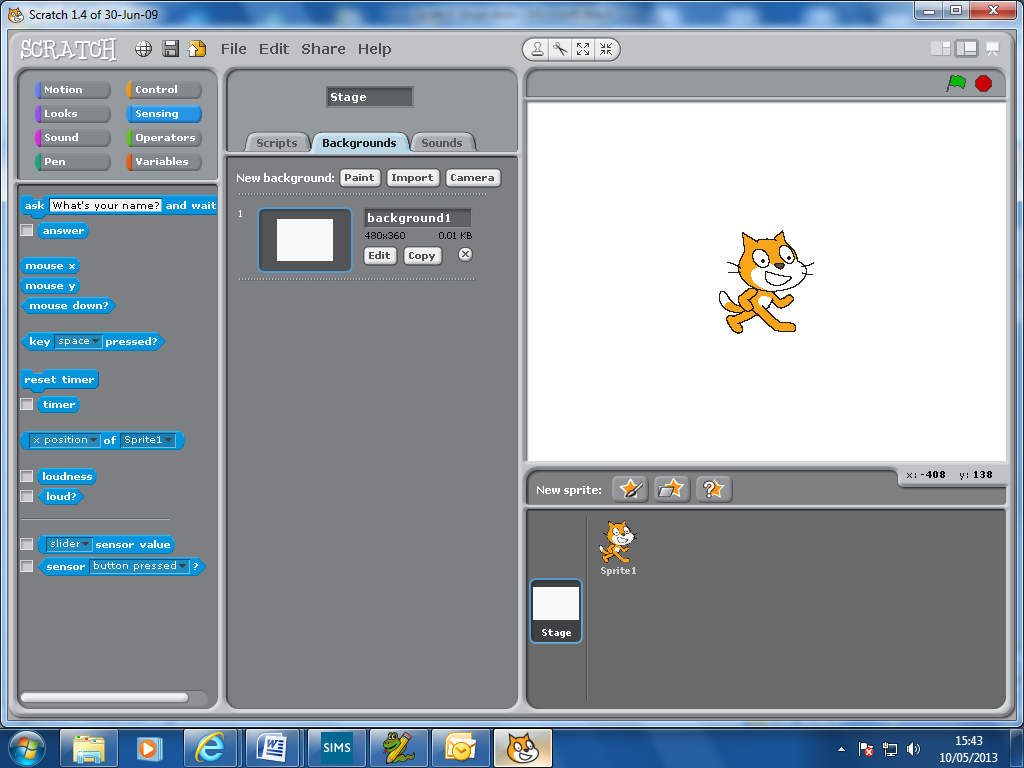
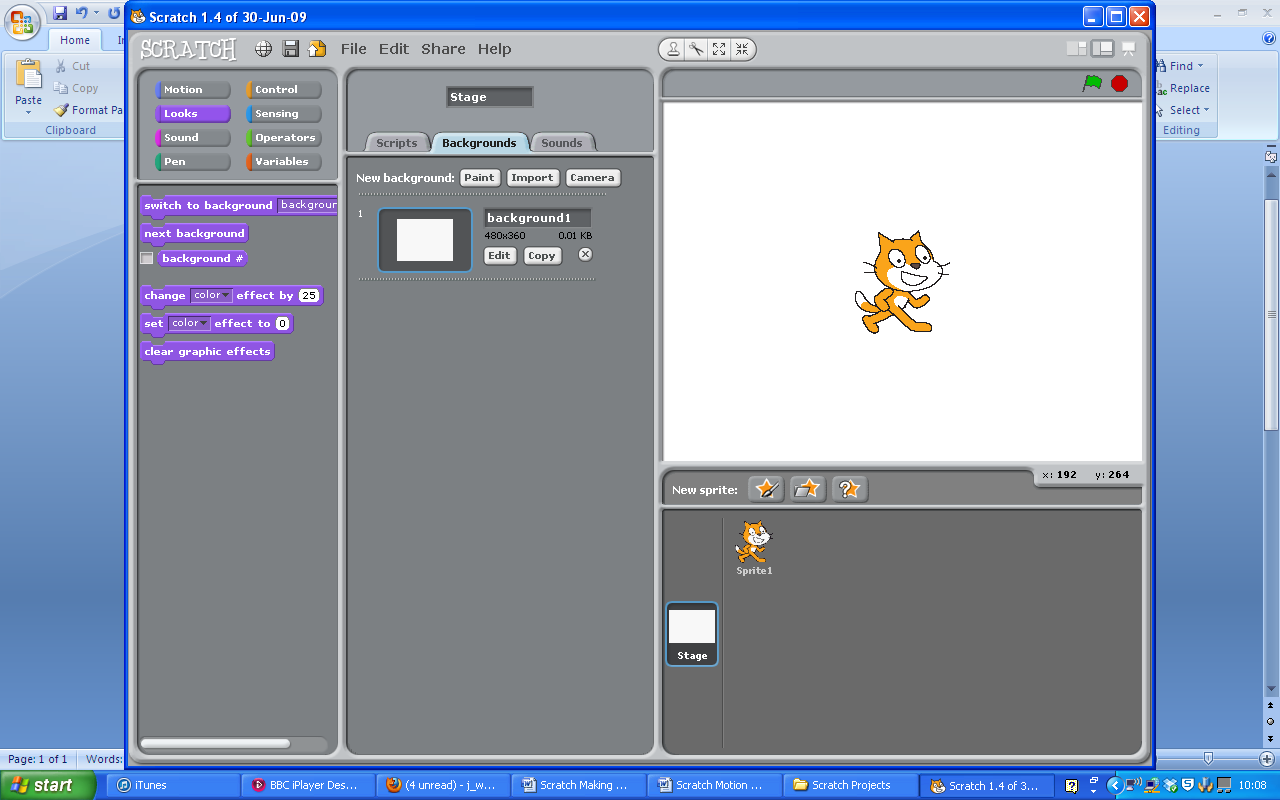
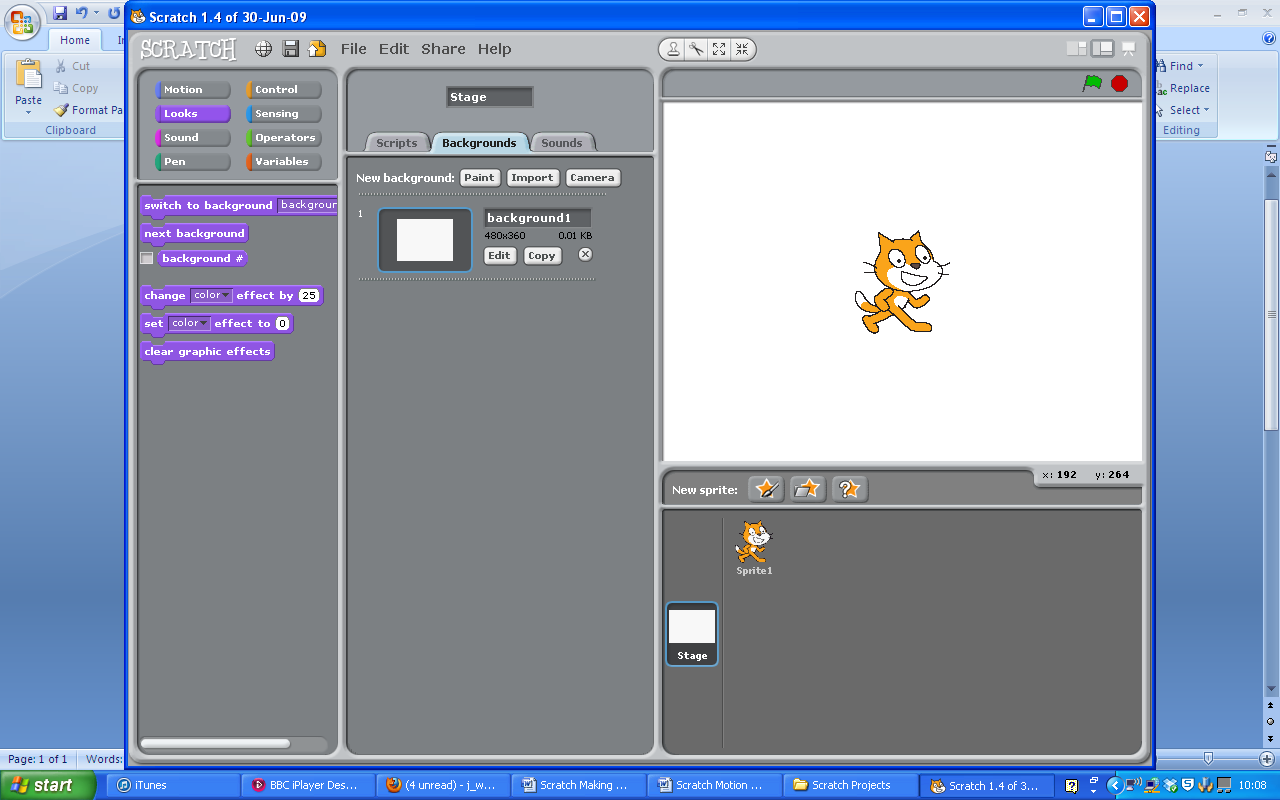
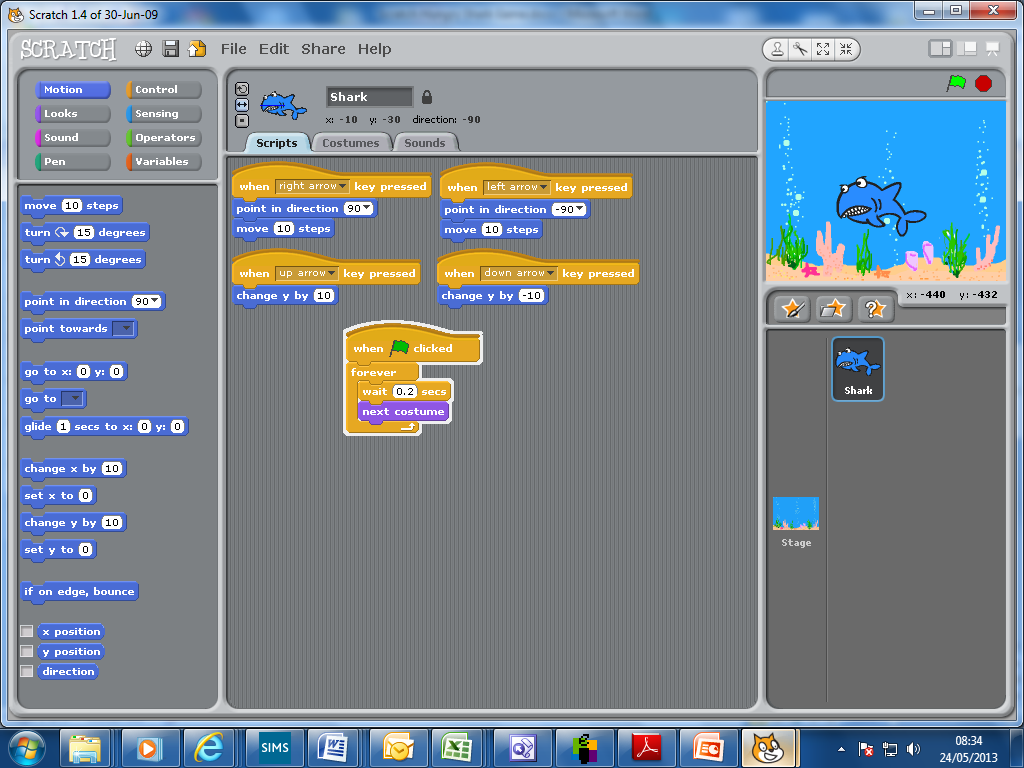
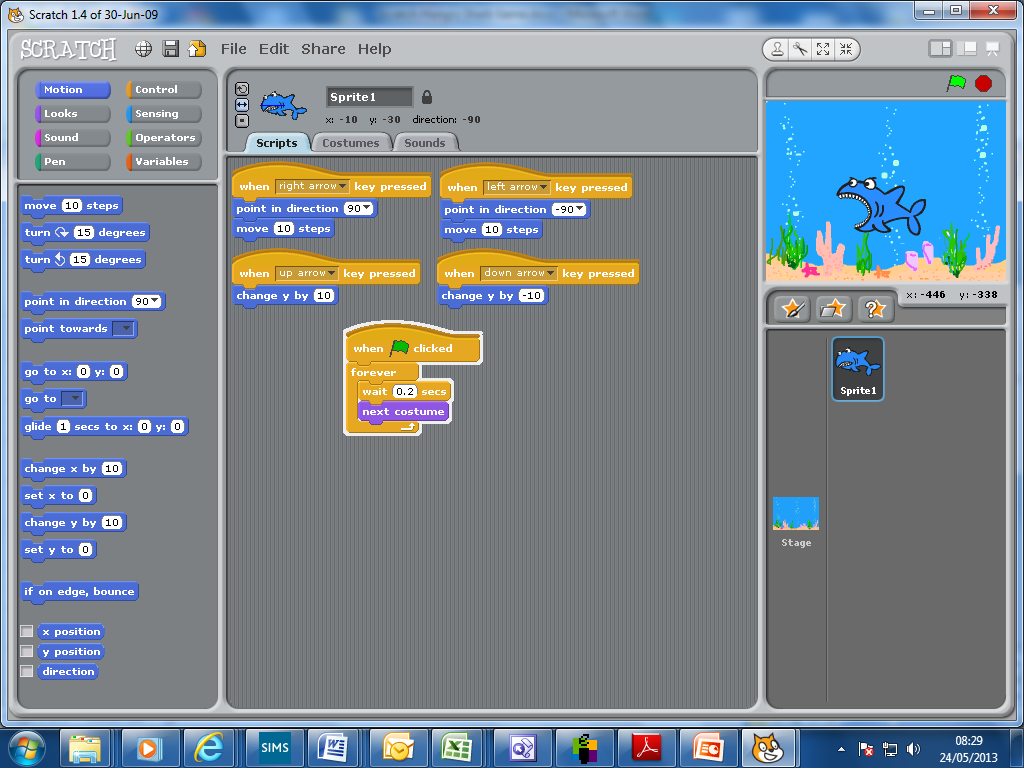
**Scratch Hungry Shark Game**

**Your Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Your Tutor Group\_\_\_\_\_\_\_\_\_\_\_**

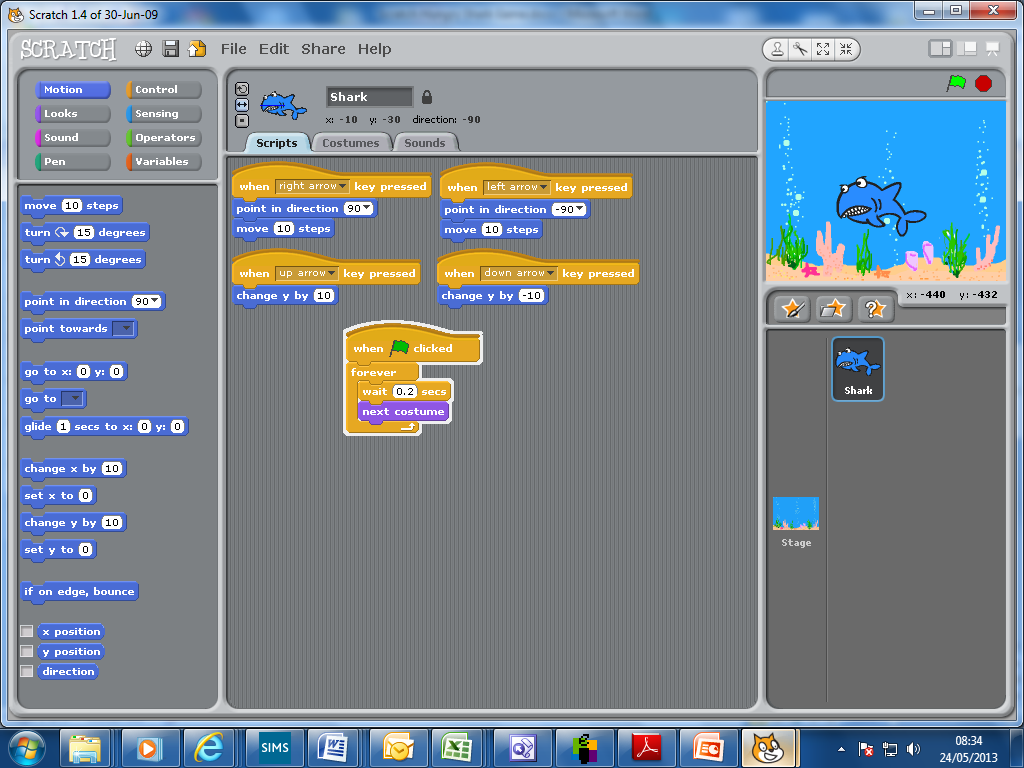
**Follow these instructions to set up the game (tick off each one as you go)**

**First we’ll set up the stage and some characters. Not only can the sprites be programmed, so can the stage area!**

1. **Select the stage area**
2. **Then select backgrounds** 
3. **Click the import button . From the *Nature* folder, import the *underwater* background. Delete the original white background .**
4. **Copy the background and edit it to make it look like the bubbles have moved. Then program the background to switch. **
5. **Next select the cat sprite and delete him (right-click)** **.**
6. **Next Select a new sprite** **. Choose *shark 1a* from the *animals* folder. Make him smaller too, using this button: **
7. **Select Costumes for the shark** **and import costume *shark 1b*.**
8. **Now we programme the shark to get him biting, and so that we can control his movement with the arrow keys on the keyboard. Test it out.**

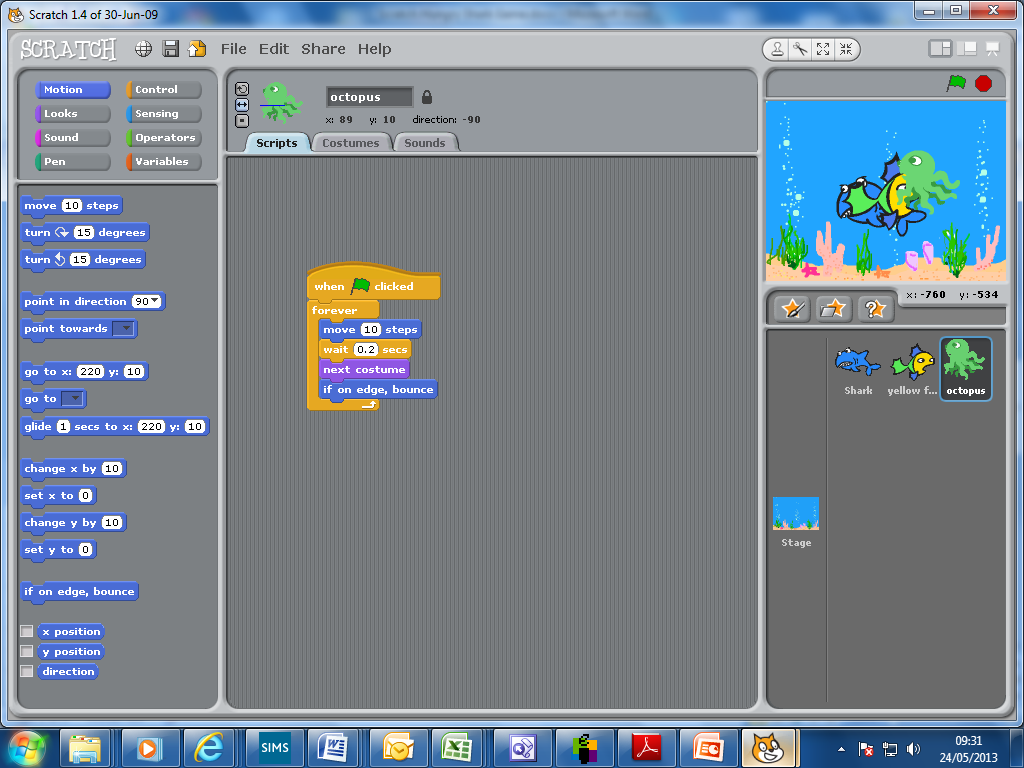
****

1. **You will find that when you turn the shark he will flip upside down. Set this button at the top of the screen to stop this (it restricts rotation to left and right facing only).**



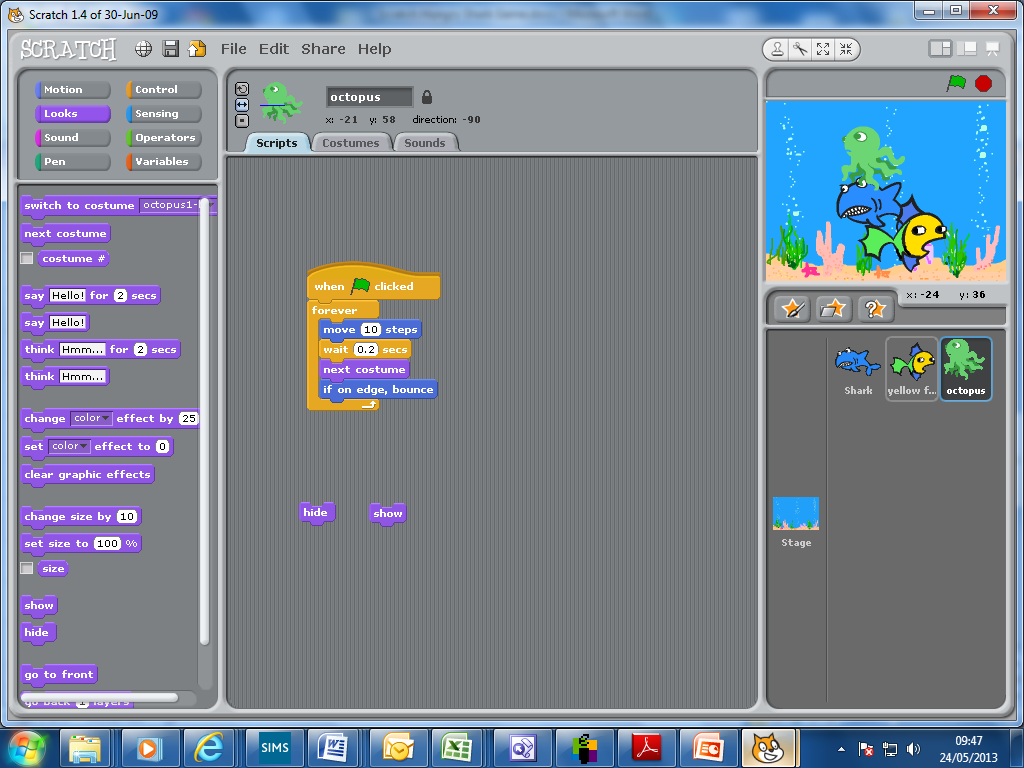
**Whilst we’re here let’s rename him *shark*. And look at the other information shown (co-ordinates of him and direction facing).**

1. **Next we need some fishy creatures to catch and eat! Let’s import two new sprites: from the *animals* folder choose the *octopus* and *fish3*. The octopus has another costume so import that. Fish3 hasn’t got a second costume so make one. Make both sprites smaller than the shark.**
2. **Program both the octopus and fish just to swim backwards and forwards forever.**

****

You know how to speed them up or slow them down.

***Hint: if you right-click on a programme script you can duplicate it and then drag it onto another sprite.***

1. **The idea now is that we move the shark to eat the fish & octopus. When the shark reaches the fish & octopus we need them to disappear, so we’ll be using .**

**We also will need the fish & octopus to know when to disappear**

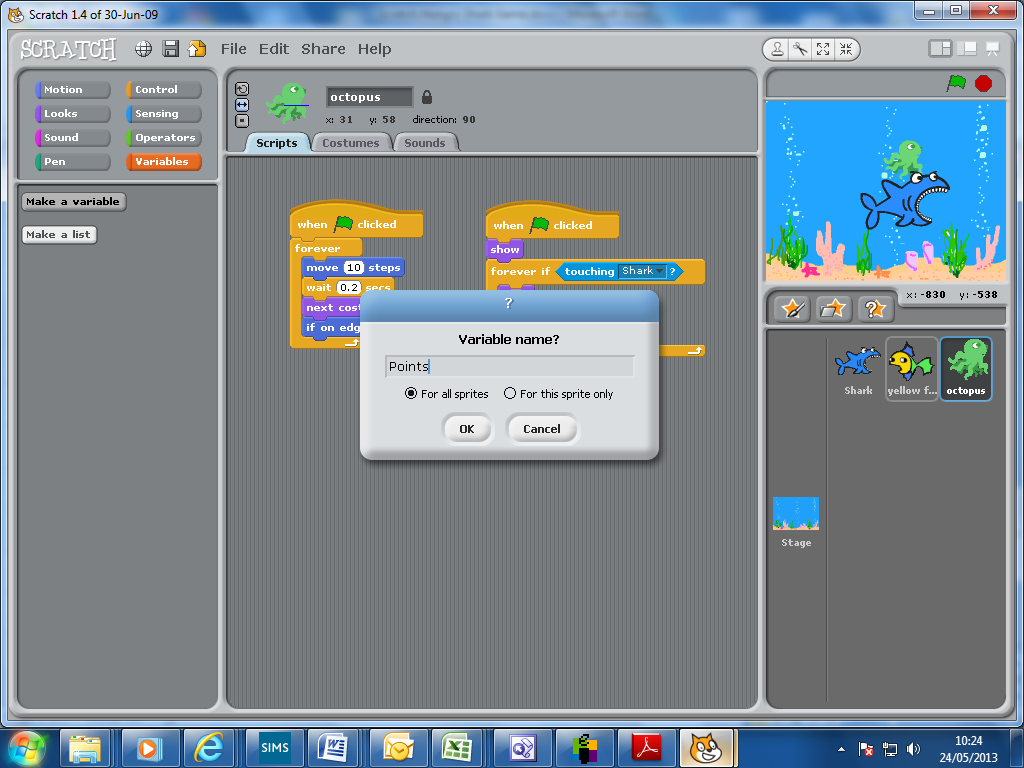
****

**So here is a second script for both sea creatures that gets them to respond to the shark.**

**Now test it out. When the shark touches the fish/octopus they disappear for 3 seconds.**

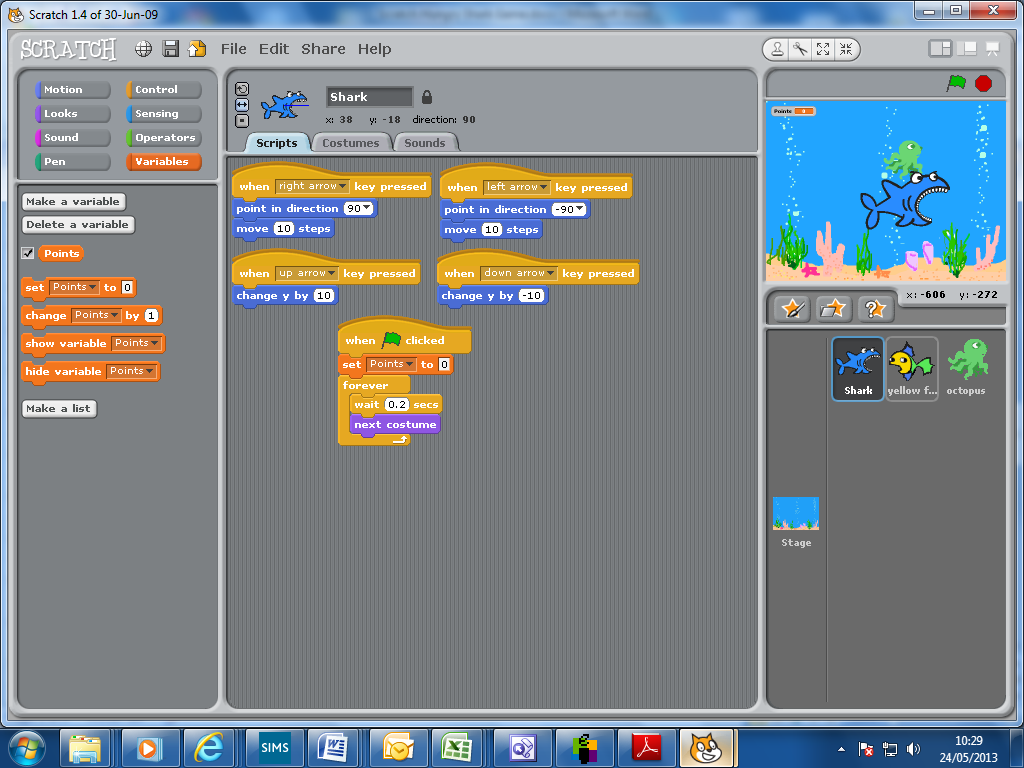
1. **Scoring points. To score points we need something called a variable. These are found in the dark orange variables toolbox.**

**Make a variable and call it *Points.***

****

**It will appear on the background.**

**At the start of the game we need Points to be set to zero.**

****

This sets the Points variable to zero.

**Here, we modify one of the Shark programme scripts**

**We also need to score points when we eat fish.**

****

This increases the Points variable by 1 when the Shark touches them.

**Here, we modify one of the fish/octopus programme scripts**

1. **Now make the game more interesting by adding more sea creatures to eat.**

**Make some larger/smaller than others. Make some faster/slower. Make the faster and smaller ones worth more points.**

1. **To make the game even more interesting, let’s make the sea creatures appear and re-appear in random locations.**

****

These green operations create random numbers.

**Here’s the modified programme script from the sea creatures.**

1. **We can even make the amount of time the sea creatures are visible a random amount of seconds.**

****

**Here are the final sea creature scripts.**

1. **Now try and make the game for two players!**

**How do we end the game?**

**You have a few choices on how to finish the game. You should try each one and see which one works best for you. Maybe you can think of others?**

1. **Using time (method 1)**

****

**Here we have a variable called “time left” that starts at 15 seconds. Every second, 1 is subtracted, until we get to zero. At zero we stop the game.**

1. **Using time (method 2)**

****

**There is timer that runs forever in the background of every project. You can see it like this . It always counts up. It can be reset to zero at the start of a game using this . As the game is played, its value can be subtracted from the start time, eventually becoming zero. So, for a twenty second game:**

1. **One player reaches a certain score**

****

**Another way to finish is to say “first player to 10 wins”, for example.**

**A nice touch to the end of a game is to make a “Game Over” screen which you switch to. Try this.**