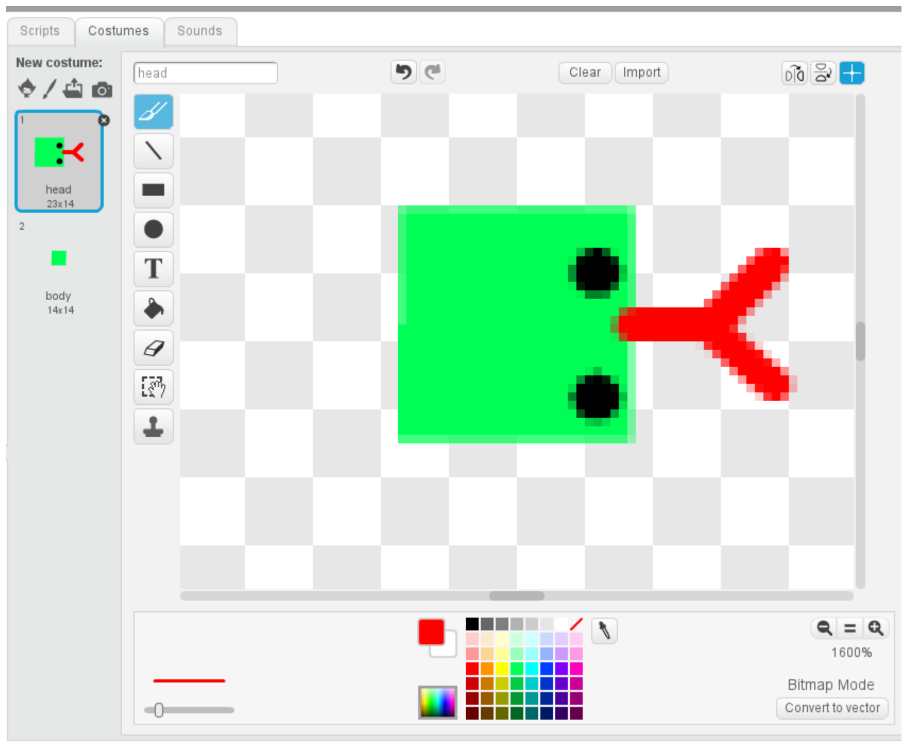
Step 1 We are going to use Scratch 2.0 from now on so everyone should download Scracth 2.0 before starting this exercise. Create a new project, delete the cat and save the project as Snake	Get it here http://scratch.mit.edu/scratch2download/
Step 2 We will create the snake sprite. There are only 2 simple costumes so its quite easy First create the body costume, a simple green square 14x14 pixels. Name this costume body. To get the pixel size correct create the costume in vector mode. Then duplicate this costume and add 2 black eyes and a red tongue as in the picture. Save as head. Zoom in to get the sizes correct and make sure to centre the 2 costumes on the screen Click on the I of the sprite now and rename the sprite to snake Also create a simple red spot sprite and save as point. It needs to be exactly 14x14 pixels also to match the snake. This will be the snake food!	 Scripts Costumes Sounds New costume: head Clear Import 1 head 23x14 2 body 14x14 1600% Bitmap Mode Convert to vector

Step 3

Lets write some script for the snake game now.

Create variables in Data called length, score, direction and delay. Only score is visible. Direction is local to the snake. The other variables are global

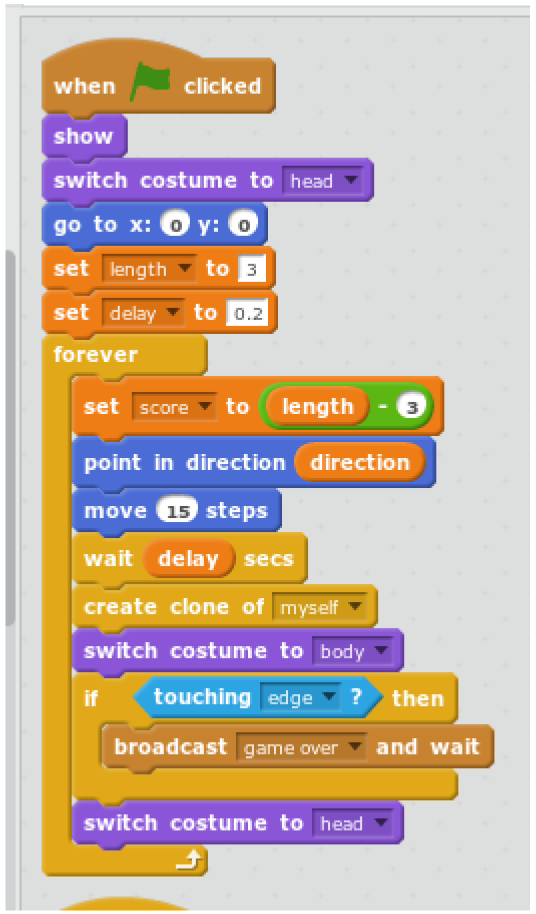
Now in the snake sprite add the script on the right. This will setup the snake and make the first part of the snake the head.

Then it loops around noting the score (everytime the snake eats a point he gets 1 bigger and this equates to the score)

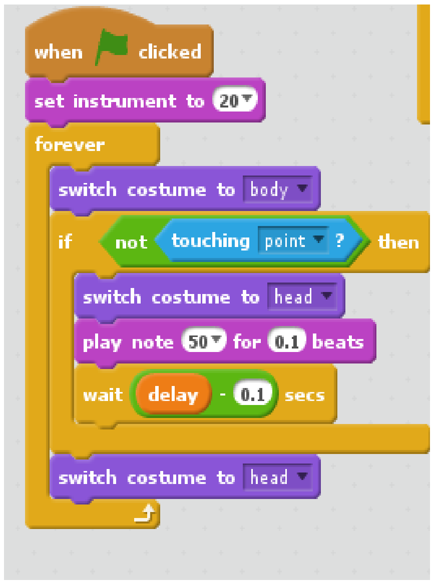
It points the snake in the direction being set in Step 4a/4b based on key presses. In the loop then it makes the snake move 15 pixels (this is important is its one more than the size of body). Then it waits the specified delay amount creating that stutter effect.

Then using a new Scratch 2.0 feature it creates a clone of the sprite . This is how the snake grows and moves. The cloned version is left in the spot that the original has just moved from. More on the clone in a later step. Then it checks to see if its touching the side and if so the game is over. If noi the loop continues and the snake continues to move.

The other script on the right simply handles the audio and some more costume switching. It runs the step noise just out of synch (- 0.1) with the actual movement for effect.



```
when green flag clicked
show
switch costume to head
go to x: 0 y: 0
set length to 3
set delay to 0.2
forever
    set score to (length - 3)
    point in direction direction
    move 15 steps
    wait delay secs
    create clone of myself
    switch costume to body
    if touching edge ? then
        broadcast game over and wait
    switch costume to head
```



```
when green flag clicked
set instrument to 20
forever
    switch costume to body
    if not touching point ? then
        switch costume to head
        play note 50 for 0.1 beats
        wait (delay - 0.1) secs
    switch costume to head
```

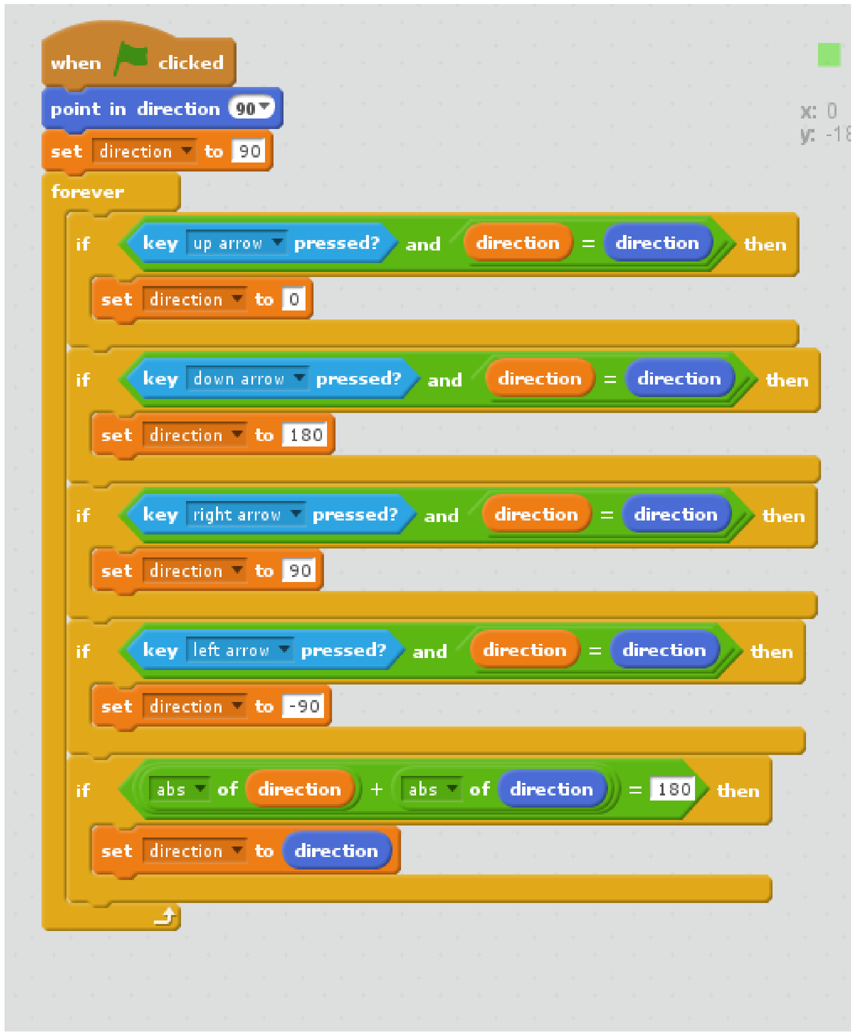
Step 4a

Now we are going to add some motion for snake using the script to the right

First it sets the direction on green flag start

The it checks for key presses and sets the direction based on key press.

The last bit of code does a check to see if the key press was trying to turn the snake around 180 degrees which is impossible. The snake can only turn 90 degrees or -90 degrees from where he is going. If this is the case then it just leaves the direction going as it is and ignores the key press

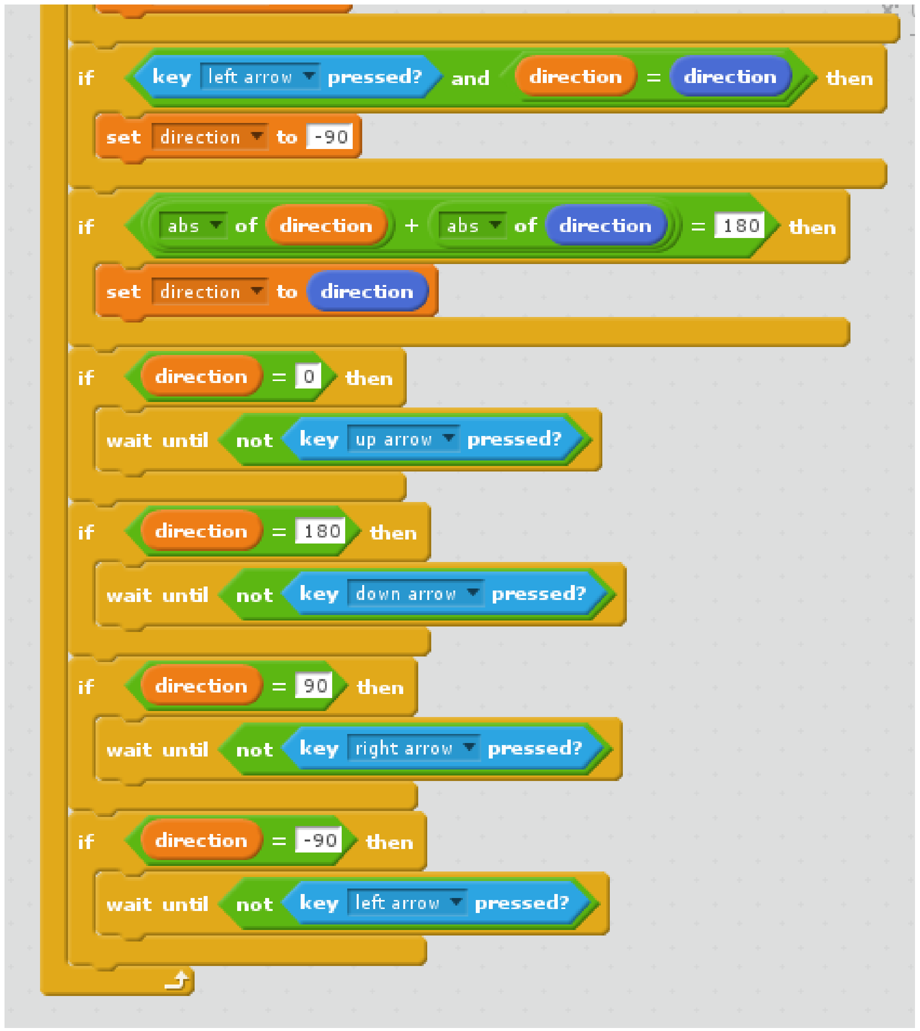


```
when [flag] clicked
point in direction 90
set direction to 90
forever
    if key up arrow pressed? and direction = direction then
        set direction to 0
    if key down arrow pressed? and direction = direction then
        set direction to 180
    if key right arrow pressed? and direction = direction then
        set direction to 90
    if key left arrow pressed? and direction = direction then
        set direction to -90
    if abs of direction + abs of direction = 180 then
        set direction to direction
```

Step 4b

The second part of this movement script is on the right and should be added below the last set direction if loop code above

This bit just stops the loop constantly going around if a key is still being pressed after it has turned. It wont affect the snake movement as this happens in the other script in Step 3



```
if < key left arrow pressed? and direction = direction > then
    set direction to -90
if < abs of direction + abs of direction = 180 > then
    set direction to direction
if < direction = 0 > then
    wait until < not key up arrow pressed? >
if < direction = 180 > then
    wait until < not key down arrow pressed? >
if < direction = 90 > then
    wait until < not key right arrow pressed? >
if < direction = -90 > then
    wait until < not key left arrow pressed? >
```

Step 5

The final bits of code for the snake are on the right

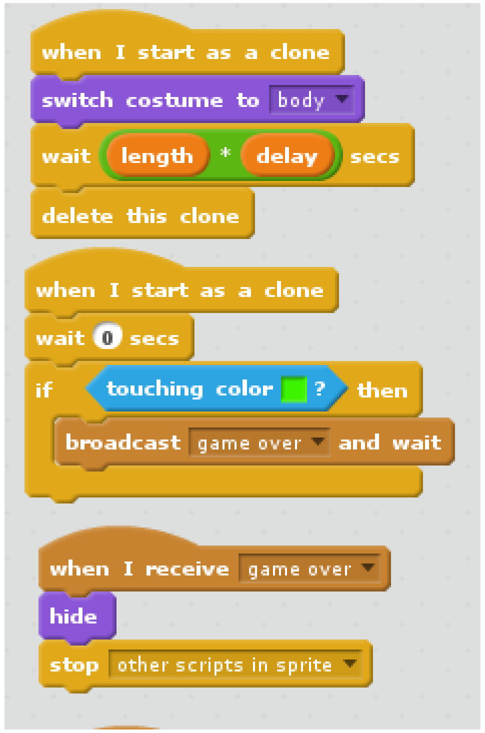
This tells the new cloned version of the sprite thats left behind to become a body costume (a cloned element is always a body)

It then simply says to just wait (no movement) for the delay value by the length of the snake. This creates the effect of movement when really only the front part of the snake is moving. The other cloned parts are stationary.

When the wait is finshed it removes itself and disappears thereby helping the motion effect.

The other clone script just checks if the snake has crossed over on himself and if so finishes the game in the same way touching the edge does.

The final bit of script tells the sprite to stop everything its doing if it receives a game over message and hide the sprite.



```
when I start as a clone
switch costume to body
wait (length * delay) secs
delete this clone

when I start as a clone
wait 0 secs
if touching color 🟩 ? then
    broadcast game over and wait

when I receive game over
hide
stop other scripts in sprite
```

Step 6

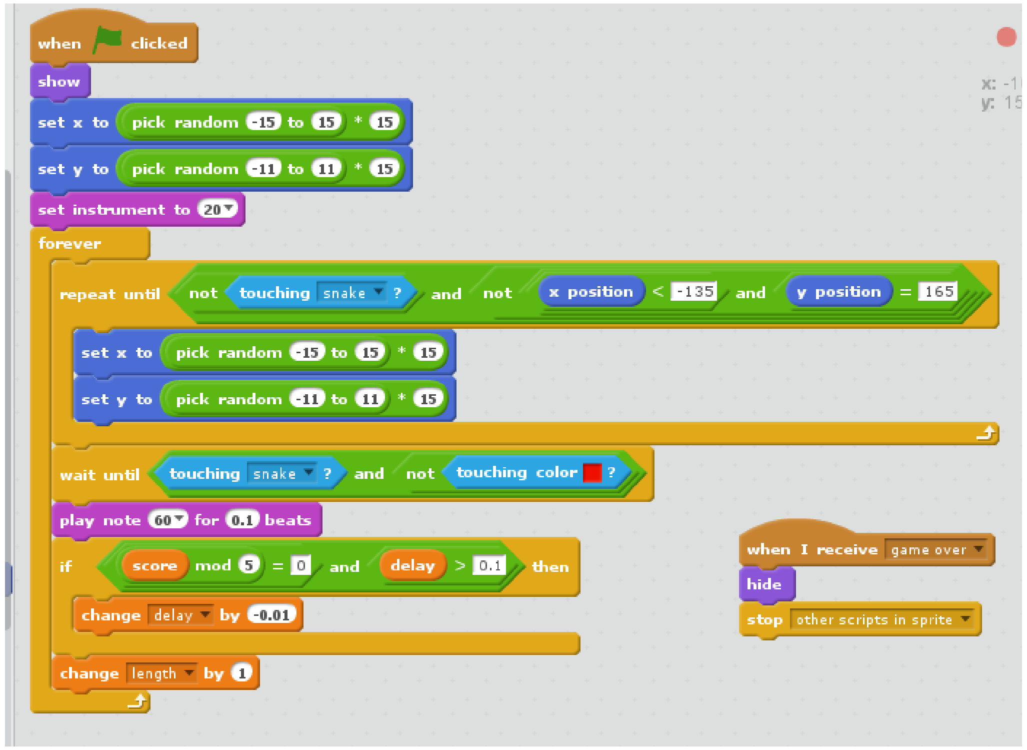
We are now going to add the script for the point (snake food)

This script creates a point at random place on the background. The loop checks if the snake is touching the point and if it is then moves it to another location

When the snake is touching the point (but not with his tongue as he needs to swallow :)) then it plays an eating sound and increases the length by 1 which also increases the score as we saw in the snake script

It also checks to see if the score is now a multiple of 5 and it increases the speed by decreasing the delay time.

Finally we check for a game over message and if we get one we stop everything.



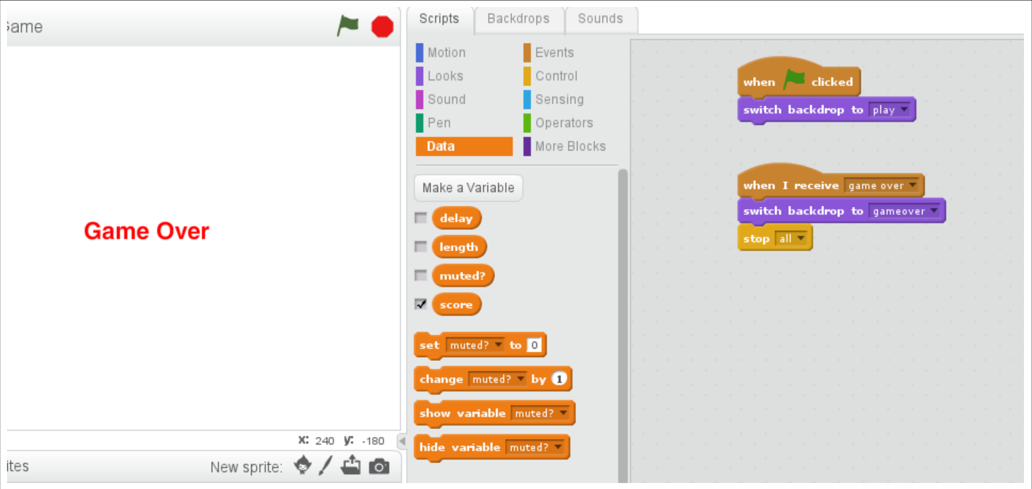
```
when ⚑ clicked
show
set x to (pick random -15 to 15) * 15
set y to (pick random -11 to 11) * 15
set instrument to 20
forever
    repeat until not touching snake ? and not x position < -135 and y position = 165
        set x to (pick random -15 to 15) * 15
        set y to (pick random -11 to 11) * 15
    wait until touching snake ? and not touching color 🟥 ?
    play note 60 for 0.1 beats
    if score mod 5 = 0 and delay > 0.1 then
        change delay by -0.01
    change length by 1
```

```
when I receive game over
hide
stop other scripts in sprite
```

Step 7

Very finally, we will add some simple script to the background to indicate when the game is over and by changing the background costumer and to restart by clicking the green flag

The game should now be complete!!



```
when [green flag] clicked
switch backdrop to play

when I receive game over
switch backdrop to gameover
stop all
```