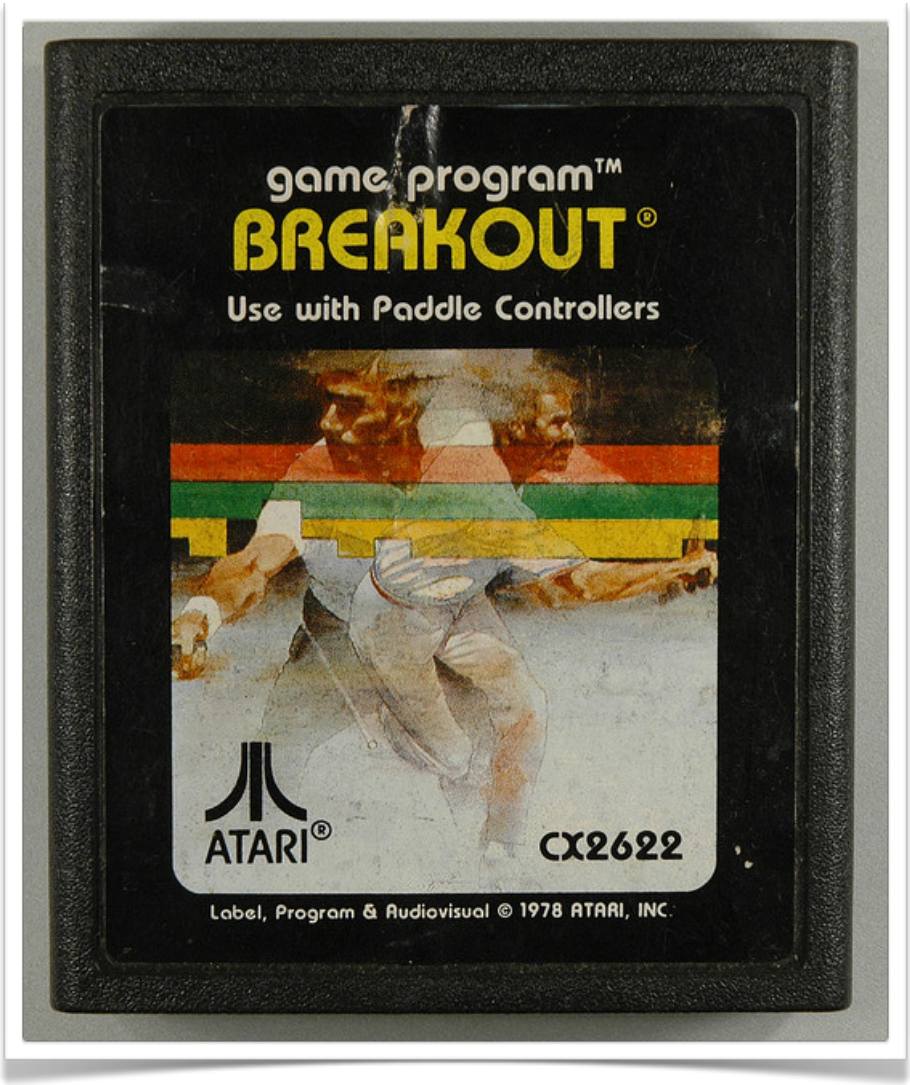
Breakout

A Simple Scratch Game



http://www.flickr.com/photos/oflittleinterest/4354422494/

By MWClarkson

Part 1- Sprites

http://www.flickr.com/photos/dividedxwexstand/4139740627/

This tutorial assumes you have used Scratch before - it will give you guidance and hints on how to make the game, but it will require some THOUGHT in order to get through all of the tasks!

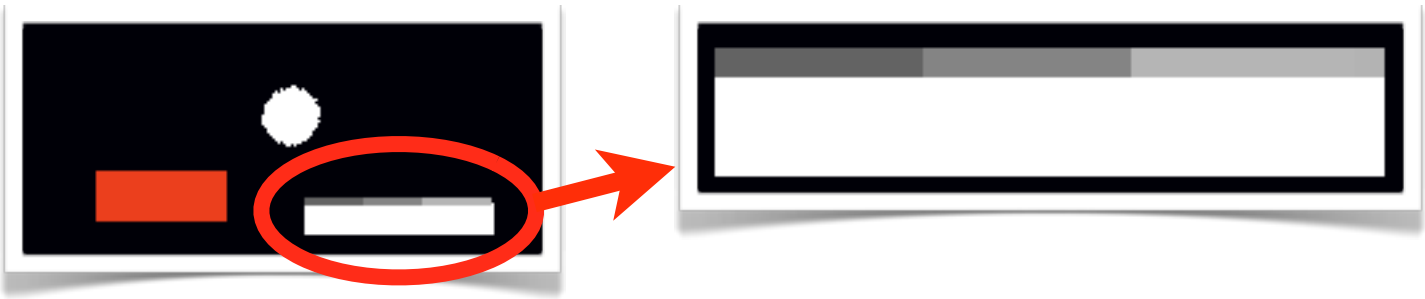
1. We only need 3 sprites (characters) for this game.

    1. Choose the **stage**, click the **backgrounds** tab, **edit** it and fill it black.

    2. Click the **paint new sprite** button and draw a small white ball. Give it the name **ball**.

    3. Draw a red brick and call it **brick**.

    4. Draw a white rectangle and call it **paddle**. Use the line tool to draw different shades of grey across the top:



By MWClarkson

Part 2- Bricks

http://www.flickr.com/photos/salvadhor/424729966/

The bricks are pretty simple to program - they only need to do two things:

1. When the **green flag** is clicked, **show** itself.

2. **Forever**, check **if** it is **touching the ball**. If it is, then **hide** itself.

That should be 4 lines of code in total.

Then you can right-click and duplicate until we have, say, 10 bricks.

Part 3- Paddle

http://www.flickr.com/photos/slack12/248326861/

The paddle is also pretty easy, just a couple of simple instructions.

1. When the **green flag** is clicked, **move** the paddle to x = 0, y = -150.

2. **Forever**, **if** the left key is pressed, the paddle should move left.

3. **If** the right key is pressed, the paddle should move right.

There are slightly different ways you code that - either as two sets of code or one. I did it in one set using 7 lines of code.

As always, make sure it works before you move on.

Part 4- The Ball

http://www.flickr.com/photos/ableman/607118618/

This is the difficult bit now. But also the bit where the game starts to come together.

1. When the game starts we need to put the ball into the very middle of the screen (x = 0, y = 0). We also need to make sure it is pointing down.

2. The ball should always move forwards - 6 steps at a time.

3. If the ball is touching the middle grey colour on the bat it should turn 180 degrees (so a ball going straight down will go straight up).

4.If the ball is touching the grey colour on the left of the bat it should turn 160 degrees, so it will bounce off at an angle.

5. If the ball is touching the right grey colour on right of the bat it should turn 200 degrees, so it will bounce off at a different angle.

6. If the ball touches the edge, it should bounce.

7. If the ball's y position falls below -165 then the game should stop because you missed the bat.

8. If the ball touches red (the brick colour) it should turn 180 degrees.

By MWClarkson

Test the game to see how it works. It should be OK - not perfect, but more-or-less a working game.

Here are a few improvements we can make:

1. The ball sometimes gets stuck on the bat. Make it move up by just 5 pixels once it has changed direction.

2. If you win, the game keeps going. Do fix that we need to use a **variable** to try and keep count of how many bricks we have knocked out. You might need to talk to your teacher - or a friend in class - about how to do this.

3. The angles at which the ball bounces back are often a bit odd. That's because you have to work out the angle which takes a little more effort. I **could** just give you the answer, but that would be a bit easy.

Trigonometry comes in very handy here and, again, your teacher might need to give you a hand to get to the right answer.

4. You could change the colour of some of the bricks by painting a different **costume** (maybe yellow, blue and green). You will need to add some code to the ball to make it bounce when it hits the right colour, so bear that in mind.

5. You could try programming different levels - possibly writing code to make the bricks jump to the right position - again, a variable would be needed to record which level you are on.