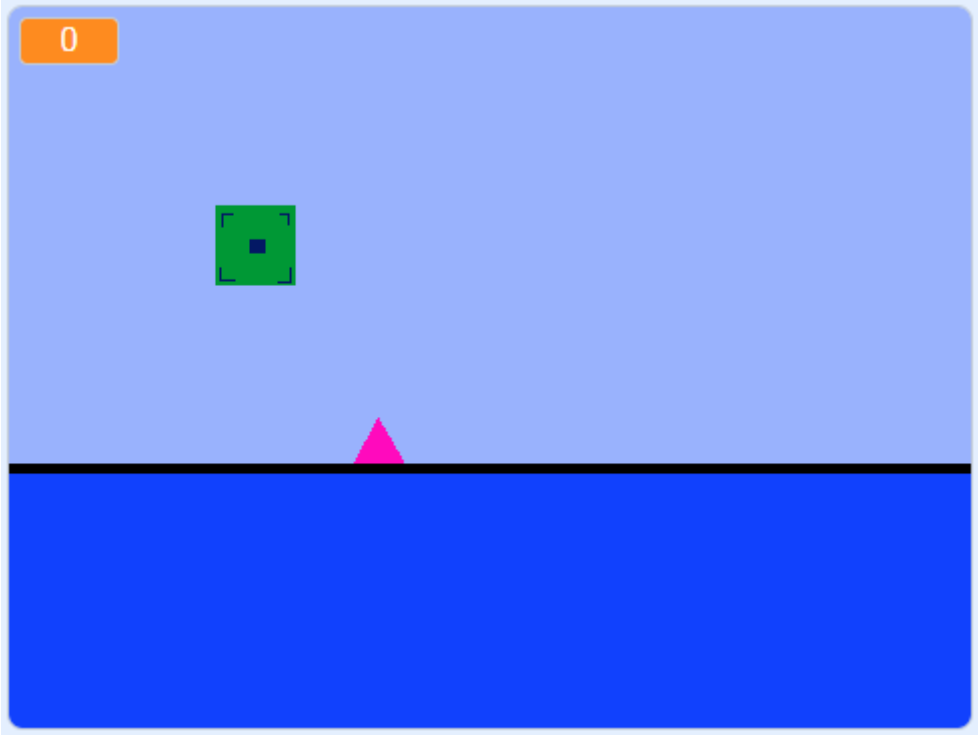
Geometry run!

With thanks to the very popular game with a similar name.

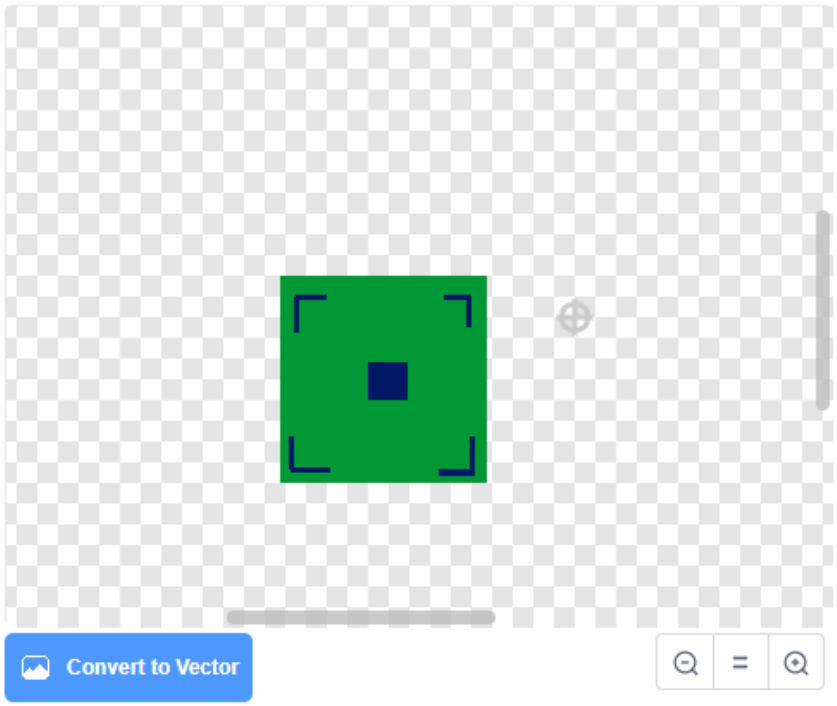


You can see a version of this project here:
https://scratch.mit.edu/projects/196683316/

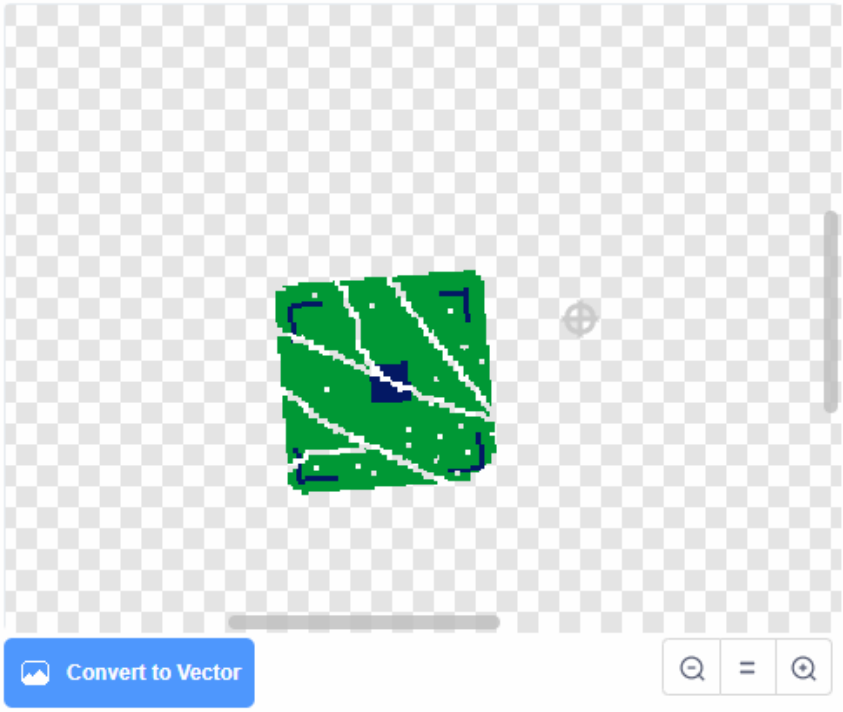
This game is all about jumping over obstacles and surviving as long as you can!

Step 1: Choose your player

- Since the word ‘geometry’ occurs in the project name, we had better use geometric shapes for our sprites! I chose a square for my player sprite, and drew a costume for it, see below.
- The magnifying glass tool is very useful to allow you to zoom in and add fine detail to your sprite.
- Once I had my first costume, I renamed it ‘ok’ as this is what my player looks like when they are playing the game.
- Next, I copied this costume, and called the new one ‘broken’. I edited the broken one to look damaged, as this is the one I will switch to when I have hit an obstacle and lost the game :-(.



'ok' costume



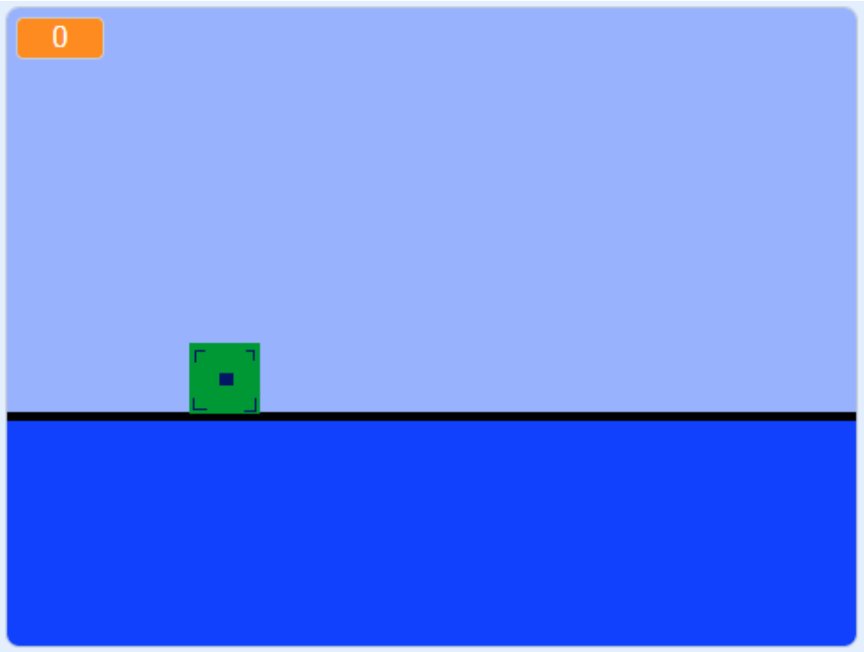
'broken' costume

- Still using the magnifying glass to zoom in, I used the rubber tool with the slider set to the smallest setting, to draw cracks and holes in my sprite, to make it look broken. I also rotated it slightly, to look off-balance.

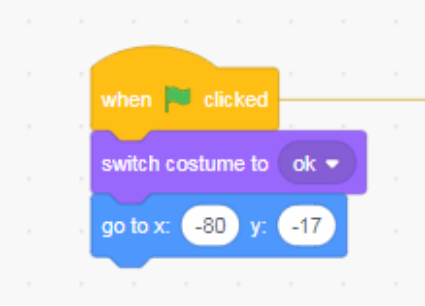
Save your project

Step 2: Design your background, and start jumping!

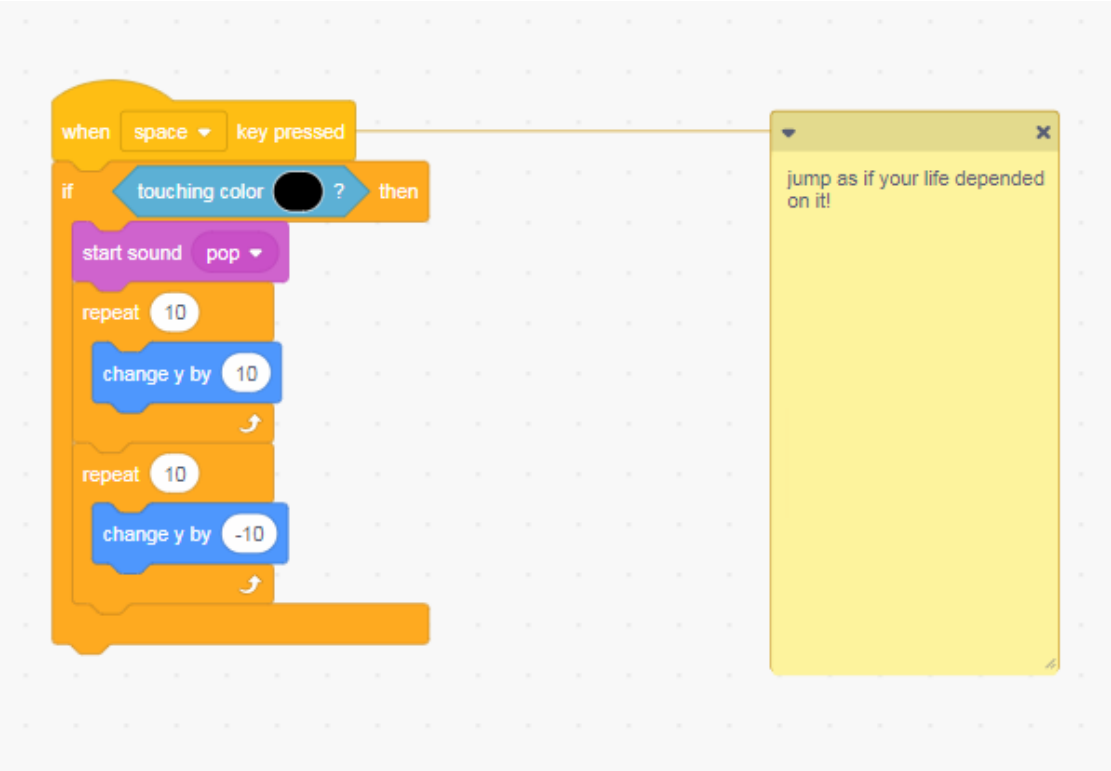
- You can use any colours you like for your background, but the black line in the middle is important – we will use this in the code, to control our jumping.
- If you choose a different colour for your line, then choose that colour in your code, where I choose black.
- Once you have drawn your line, and coloured your background, drag your player sprite to its starting position, as shown below:



- It should be sitting on the black line, and towards the left of the screen, as the obstacles will appear from the right.
- [An important thing to notice about this game, is that our player sprite does not move to the right, only up and down as it jumps. Although when you are playing, it may feel as if you are moving along!]
- Once your sprite is in position, drag in the 'go to x: y:' block, and add the rest of the code below. Your numbers should be different from mine!



- Now add the code below to make your sprite jump when the space bar is pressed.
- Remember to use the colour of your line where I have used black!
- I have added a sound too – choose one you like (but make it a short one).

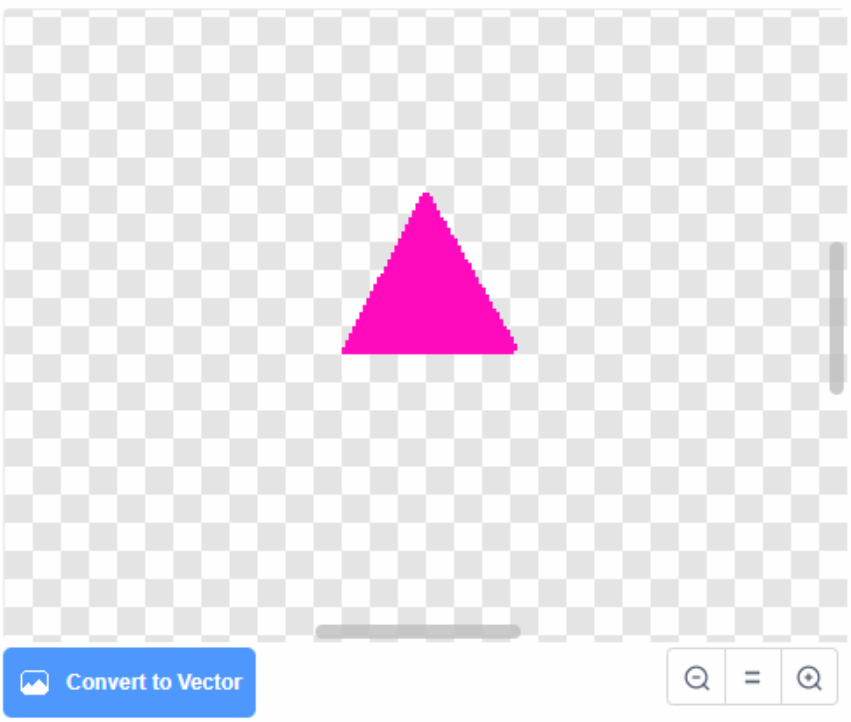


- My player jumps 100 steps high (10 times 10 steps), and then falls down again (10 times –10 steps). You can change these numbers to make the jump higher or lower, but remember you must come down exactly the same number of steps as you went up.
- Why do I check 'if touching colour' in this code? If you are not sure, try leaving it out, and playing the game. (Answer at the end.)
- Test your project a few times. The player sprite should always start in the right place, and your jump should be smooth, and end back on the line.

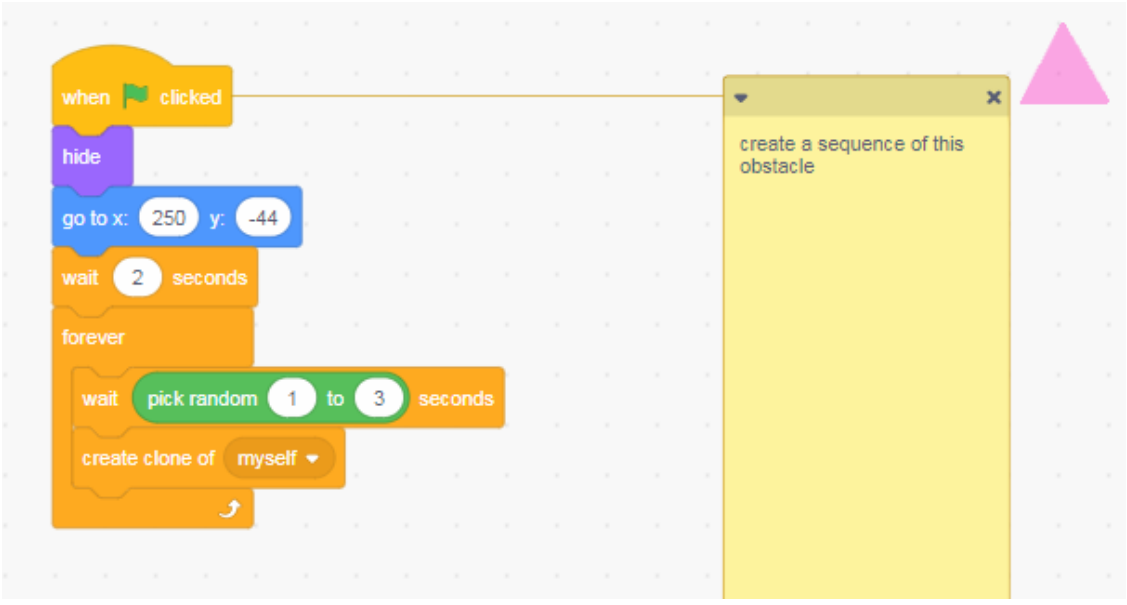
Save your project

Step 3: Create your first obstacle

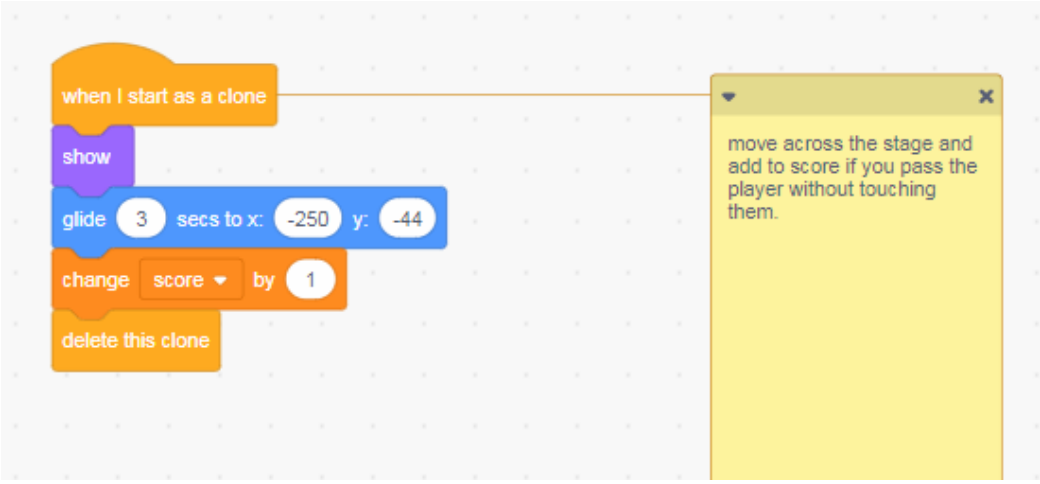
- Our obstacles also have to be geometric shapes, so I chose a triangle for my first one:



- Like before, I zoomed in with the magnifying glass to draw very careful lines for the outline, and then filled it with the paint can.
- This obstacle only needs one costume, so you don't need to rename it.
- We will use clones to make lots of nasty obstacles for our poor player to jump over – first add the code below to create clones...



- If you test this, you won't see anything yet, as we haven't said what each clone should do. Add the next bit of code and test again:



- Remember that the numbers in the 'go to x: y:' and 'glide' blocks will be different for you – the obstacle should start on the far right of the screen, sitting on the black line, and glide across until it disappears on the left side.
- You will notice I have a score variable! Every time an obstacle reaches the left side of the screen, we know the player has jumped over it successfully (because if not, the game ends). So each obstacle that makes it all the way across adds one to the player's score.
- Add a variable called score, and remember to set it to zero at the beginning of the game.

Save your project

Step 4: Music!

- Add music loops

Save your project

Step 4: Bouncepads!

- Add bounce pads

Save your project

More ideas

- We used ‘if touching colour’ in the jumping code because without it, your player can keep jumping up, even when in mid air! So you could fly over all the obstacles, and the game would be too easy to be much fun.
- Can you add new costumes for your player that are only unlocked when you reach a certain score?
- Or unlock new backdrops, new sounds, new obstacles…
- Both the obstacles and the bouncepads in this game use ‘pick random’, so sometimes they appear on top of each other. Can you think of a way to avoid this, while still making them unpredictable?
- How about obstacles that are in the air, so you have to avoid jumping into them?
- Or bounce pads in the air, that bounce you back down again?