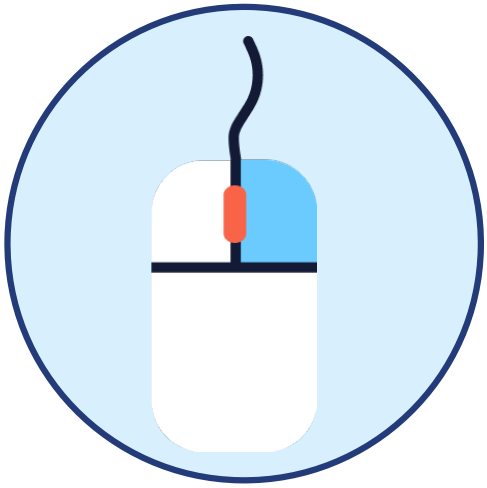
MOONHACK 2023

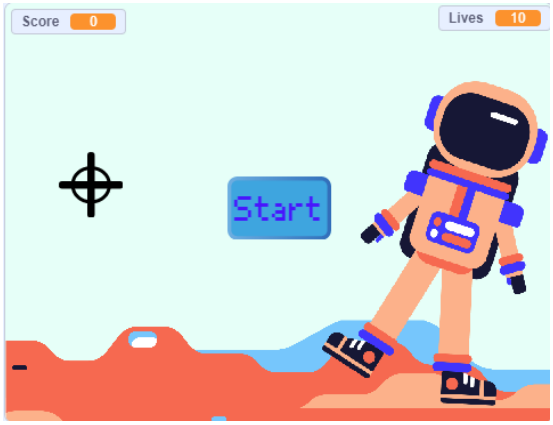
PROTECT THE PLANET



Shoot the stars to protect the planet. What is the highest score you can get?

INTRODUCTION

Make a game where the user aims and shoots the stars to protect the planet.



What you will need

HARDWARE

A computer capable of running Scratch 3

SOFTWARE

Scratch 3:
either online
rpf.io/scratchon
or offline
rpf.io/scratchoff

What you will learn

- How to use **clones**
- How to create actions based on **score**
- How to **use the mouse** to control actions

Project Links

Starter project

https://scratch.mit.edu/projects/872291676/

Completed project

https://scratch.mit.edu/projects/872291611/

Additional notes for educators

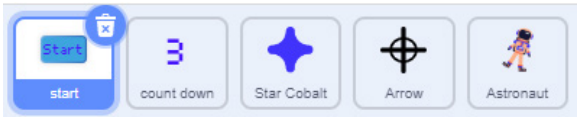
Check out our blog post for this project with tips, curriculum and supporting material at **codeclubau.medium.com/**

To watch the code along video go to our **YouTube** channel

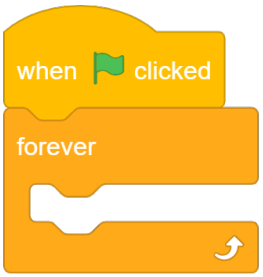
STEP 1 - STARTING THE GAME

Open the starter project - https://scratch.mit.edu/projects/872291676/. The first step is to set up the start button for the player to start the game.

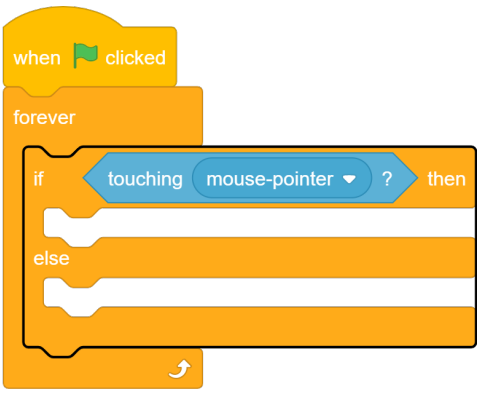
Click on the start sprite.



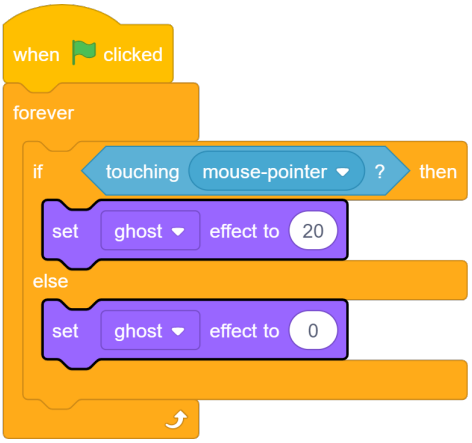
Start your first algorithm with the event, when green flag is clicked. Add a forever block.



Add an if-then-else block. We want players to know if they are touching the start button, so we will make it slightly change colour. Add a sensing block for touching mouse pointer.



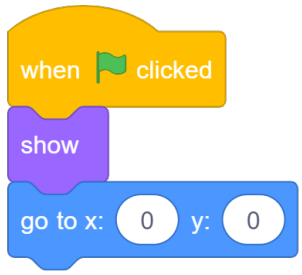
In the then section add a block to set ghost effect to 20. In the else section add a block to set ghost effect to 0.



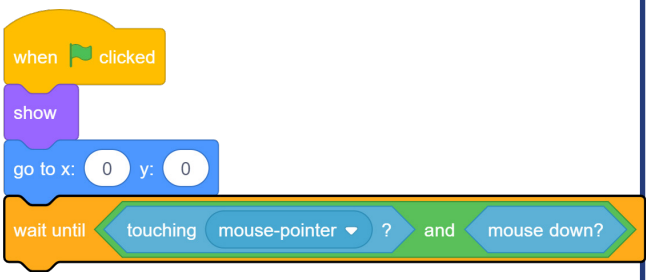
Test your button. It should change colour slightly when the mouse hovers over it.

Let's add code so that the player can click the button to start the game.

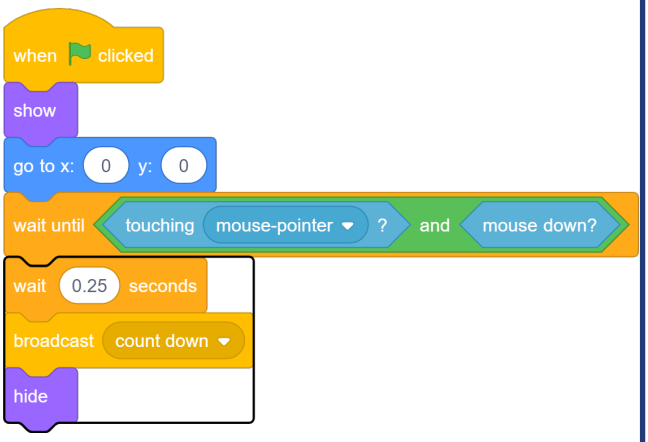
Start with the event when the green flag is clicked. Have the sprite show in position.



We want the sprite to react when the mouse pointer is touching it and the player clicks their mouse button. Add a wait until control block. Use the and operator and two sensing buttons to make this code work.

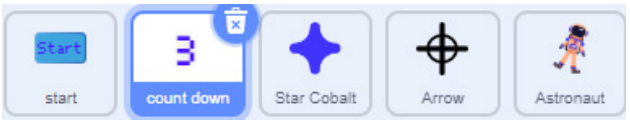


When the player clicks the sprite, add a block to wait 0.25 seconds. Create a new broadcast called countdown and add the block in. Then hide the sprite.

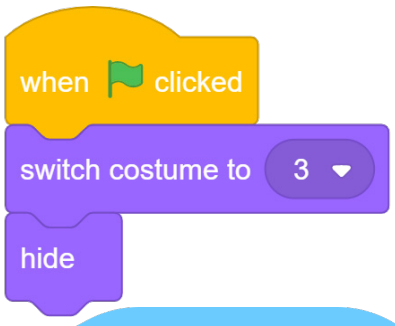


Test your button. It should hide once you have clicked it. Let's code what will happen when the 'count down' broadcast is sent.

Select the countdown sprite.

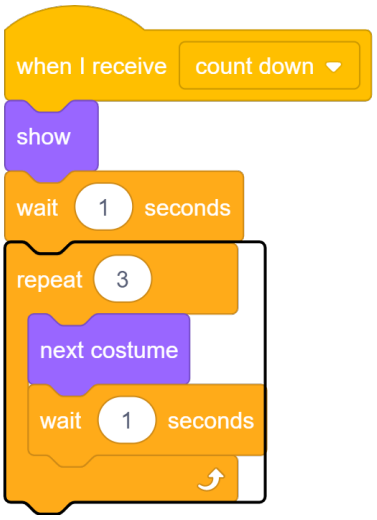


Start the algorithm with when the flag is clicked. Add blocks to switch to costume 3 and hide.

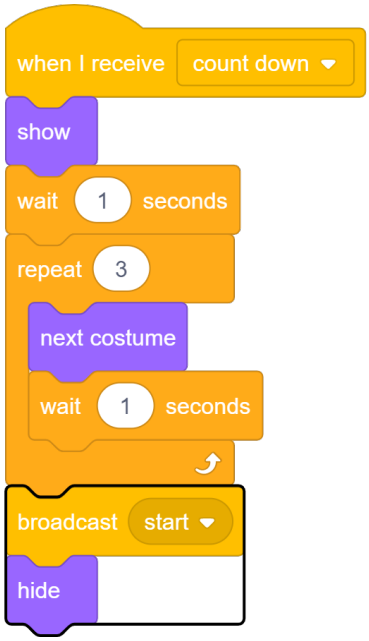


Start a second algorithm with the event when I receive count down. Add blocks for the sprite to show and wait 1 second.

This sprite has costumes that provide a countdown. Add a repeat control block. Inside it add a block to switch to the next costume and wait 1 second.



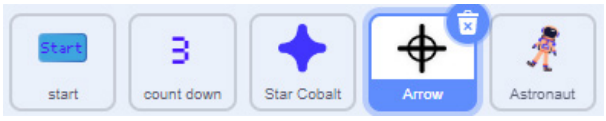
Under the repeat block, create a new broadcast called start. This broadcast will be used to get the game started. Add a block to hide this sprite.



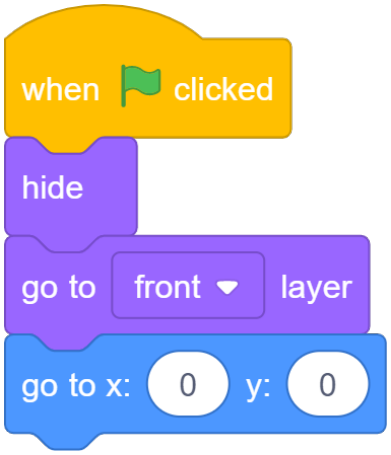
Test your code. Once you click the button you should see a countdown on your screen that counts 3, 2, 1, start.

STEP 2 - CREATING THE CONTROLS

Select the arrow sprite. This sprite is going to be what the player uses to aim at the stars.



Start an algorithm with the event when the flag is clicked. Add blocks for the sprite to hide and go to the front layer. This block is important as we need the 'arrow' to sit on top of the star sprites when playing the game. Add a block to set it to go to the middle of the screen.

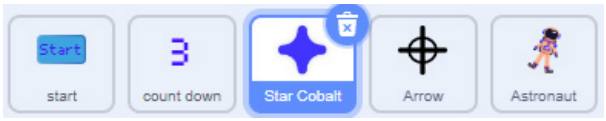


Start a second algorithm with the event when I receive countdown. Add a block for the sprite to show. We want the sprite to follow the user's mouse. Add a forever block. Inside this add a block to set x, and from the sensing menu add the bubble mouse x. Do the same for y. This code tells the sprite to always be in the same location as the mouse pointer.

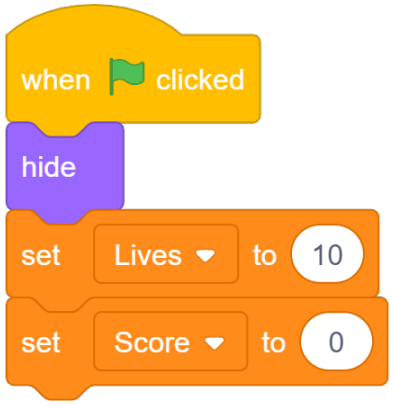
Test your code. Once you've clicked the start button your mouse should become the arrow sprite and you can now move it around the screen.

STEP 3 - SETTING THE VARIABLES

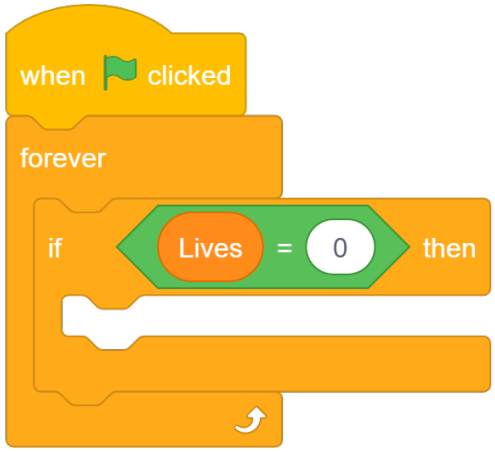
Select the star sprite.



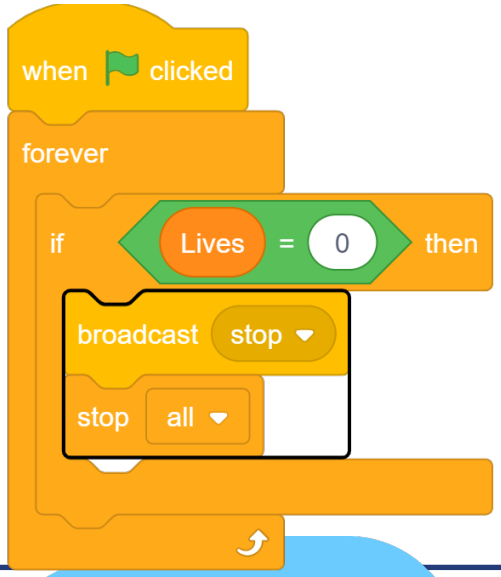
The game will be controlled by lives and the player will earn points. Start your algorithm with the event when the green flag is clicked. Add a block to hide. Create a variable called Lives and set Lives to 10. Create a variable called Score and set Score to 0. Move the counters on your screen to where you would like them.



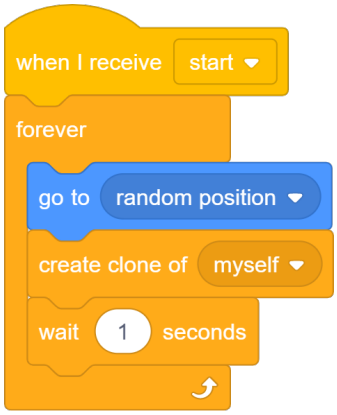
Next we will create conditions that if the player gets to 0 lives the game ends. Start with the event when the green flag is clicked. Add a forever block and inside it an if-then block. Use the = operator. Add the lives bubble from your variables menu to create the control of if Lives = 0 then.



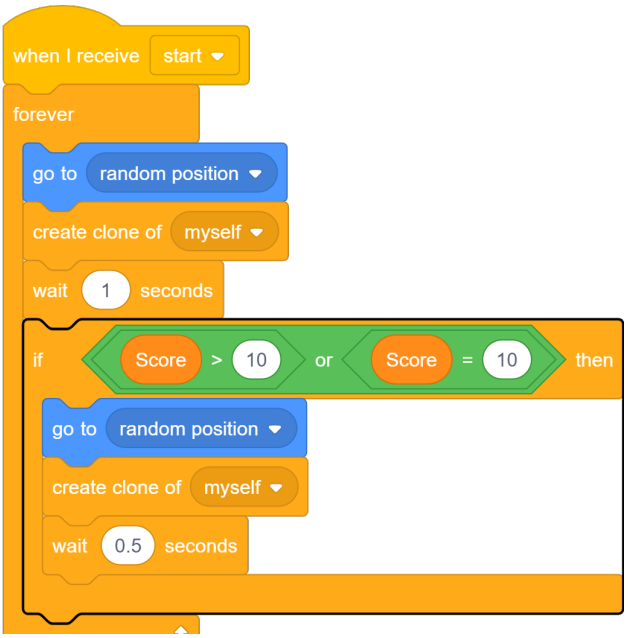
Create a new broadcast called stop and add this block inside the if-then block. Then add a stop all block.



The stars are going to show in random positions for the player to shoot. Let's create this algorithm. Start with the event when I receive start. Add a forever block and inside this blocks to go to random position, create clone of myself and wait 1 second.



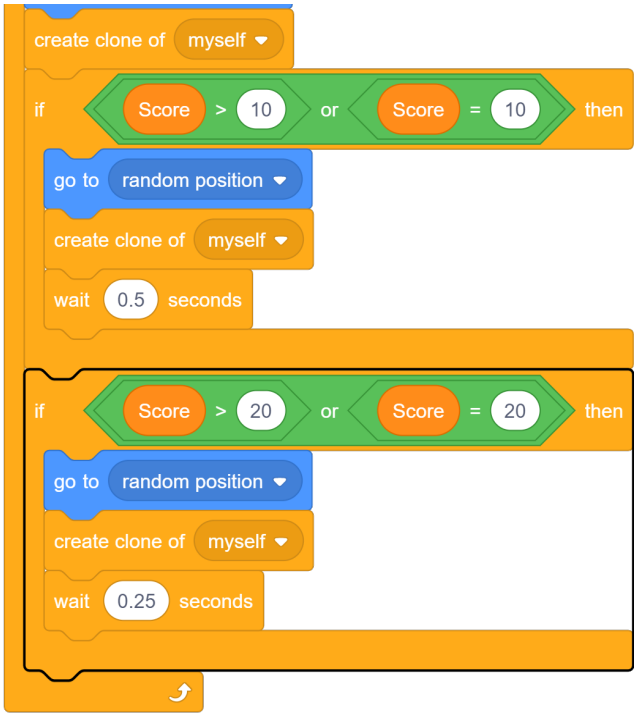
Add in an if-then block. For the if bubble use 3 operators and bubbles from the variables menu to create the condition that if the score is more than or equal to 10. Then add a block to go to random position, create clone of myself and wait 0.5 seconds. These can be duplicated from the code above.



Use your right mouse button to duplicate the if-then block that you just created and add it underneath.

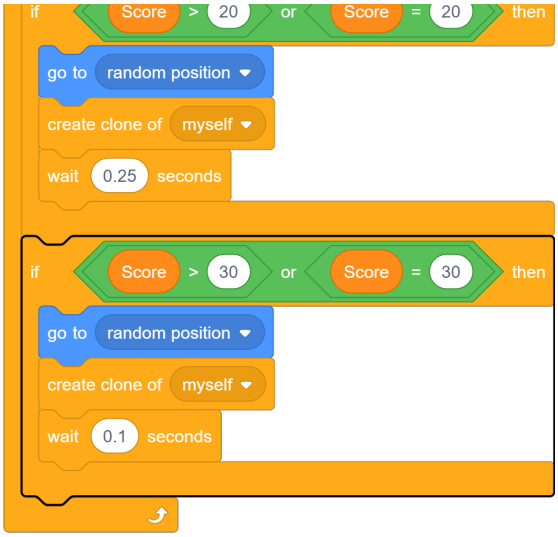
Change the conditions - score is 20 and time is 0.25 seconds.

This will speed up how quickly the star shows on the screen after the score reaches 20.

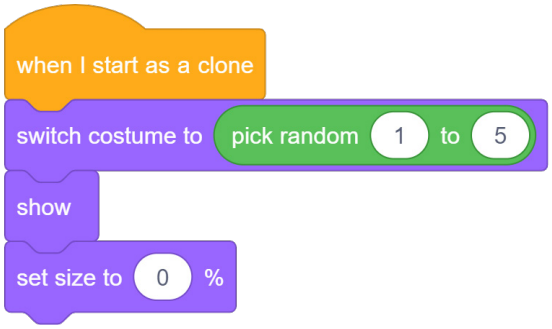


Duplicate the code one more time and change the conditions to reflect a score of 30 and a wait time of 0.1 seconds.

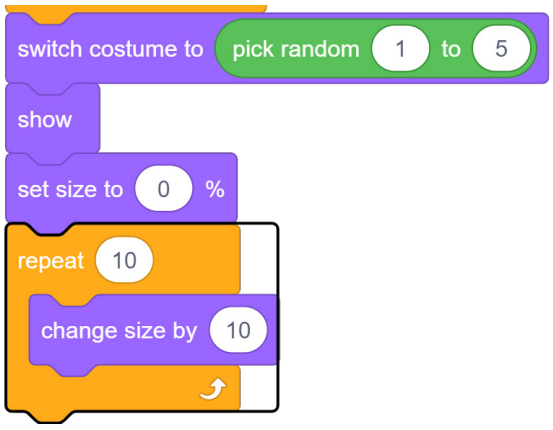
One more step before we can test. We need to code what the sprite will do when it starts as a clone. We will create two algorithms to control this, one that controls what to do if the player clicks the star and one that controls what to do if they miss the star.



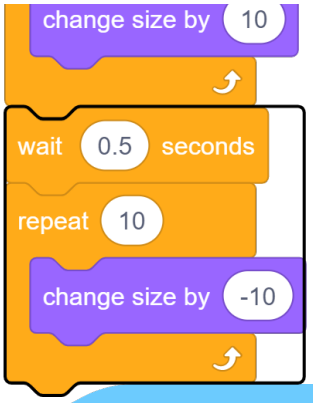
Start an algorithm with the event when I start as a clone. Add a switch costume block and use a pick random operator. (The star has 5 different colours). Add a show block and a block to set size to 0%



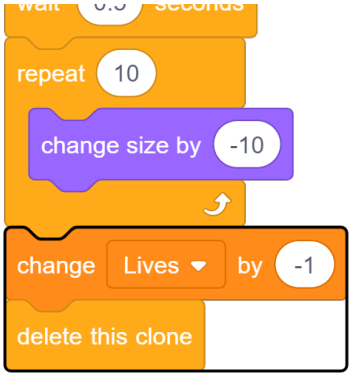
Underneath add a repeat block and inside it place a change size by 10 block. This will give the illusion of the sprite growing from nothing to full size.



Then add a wait 0.5 seconds block. Add another repeat block and inside it place a change size by -10. This will give the illusion of the sprite shrinking from full size to nothing.

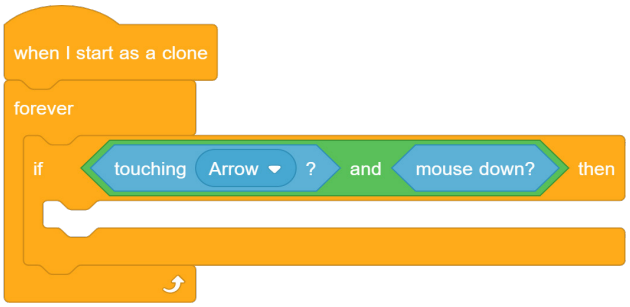


Finish the algorithm with change lives by -1. This means that if they miss the star they will lose a life. Then add a block to delete the clone.

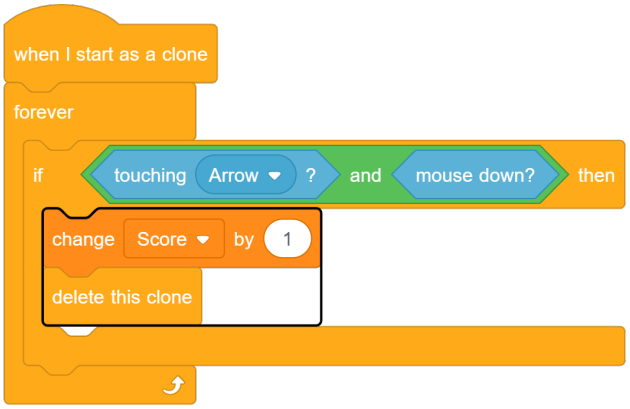


Test your code. The stars should appear in random places on screen. You won't be able to click them yet, and you will lose a life each time.

Start your next algorithm with the control when I start as a clone. Add a forever block and inside it an if-then block. Use the and operator to join the sensing bubbles touching arrow and mouse down. This creates a condition of what to do when the player has the arrow touching the star and they click their mouse button.



Add a block to change the score by 1 and then a block to delete the clone.

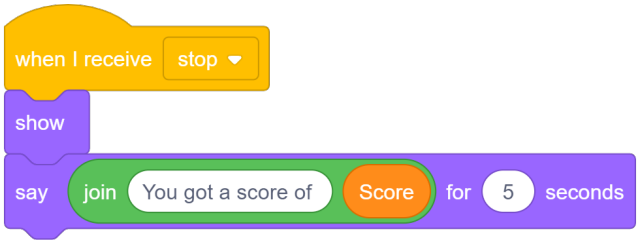


Test your code. Try and shoot the stars. Watch to see that the lives and score change correctly.

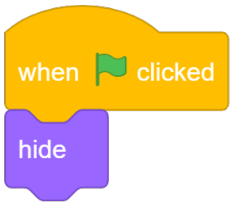
STEP 4 - ENDING THE GAME

Select the astronaut sprite.

Use the event when I receive stop. Add a show block. Add a say block using the join operator to combine your text with the score variable.



Create a new algorithm when green flag is clicked. Add a hide block.



Test your code. Your game is ready to play!

Challenges:

Theme

This game could be designed using any theme. Create this game using your own selection of background, sprites, aiming sprite, and what you are targeting.

New levels

This could be turned into a multi-level game. What ideas can you create for a second level? You could consider -

- swapping backgrounds and sprites to target
- having a timer
- moving targets

Providing information

This task was created so that people could learn how the computer mouse was invented for use in space research. How could you add this information to the project so that players will learn some information as well as play the game?

Congratulations you're a Moonhack changemaker! You can try one of our other projects or try one of the challenges.

Don't forget to talk to an adult about registering your participation and downloading your certificate at moonhack.com