

# Scratch Space Invaders

## Project structure:

- Game design
- Sprites and backdrops
- Building defences
- Motion, control and actions
- Game features
- Testing and feedback

The scratch project is called CCC Space Invaders and can be found here:

<https://scratch.mit.edu/projects/1283932022>

In this project we will build a version of the classic 1978 arcade Space Invaders game...



# Scratch Space Invaders

Over five sessions we will:

- Think about the design for our game
- Design sprites for the characters
- Code our sprites so they behave as we want
- Add game features like scores and sound effects
- Test each others' games and give feedback

We will use many feature of Scratch, including: backdrops, sprites and costumes, coordinates and animation, clones, broadcasts, variables, sensing and sound effects!

# Scratch Space Invaders

Game design

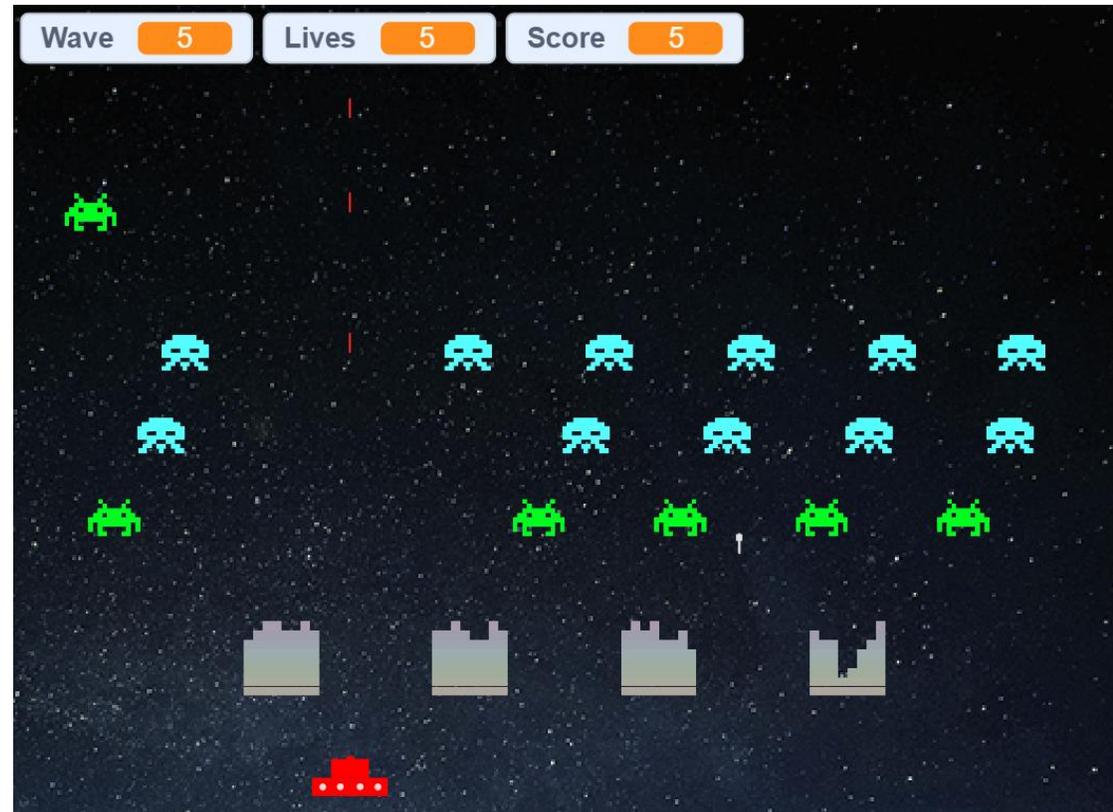
# Scratch Space Invaders

## Game design:

Before we start to design and code, we should think about how we want our game to play.

What sprites do we need? What can they do? How will they interact with each other? How do you win or lose the game?

Always remember to save your work as you go!



# Scratch Space Invaders

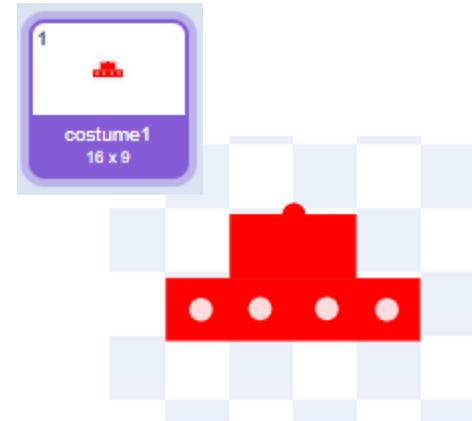
Sprites and Backdrop

# Scratch Space Invaders

## Sprites: Defender

You can design your own sprites. My Defender is a very simple shape based on blocks that I drew in the sprite editor.

Of course you can use any sprite you want. Don't worry too much about the size as we can always adjust the size when we add the sprite to the game.



# Scratch Space Invaders

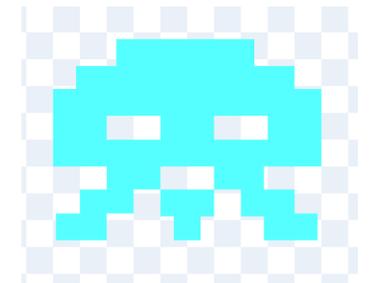
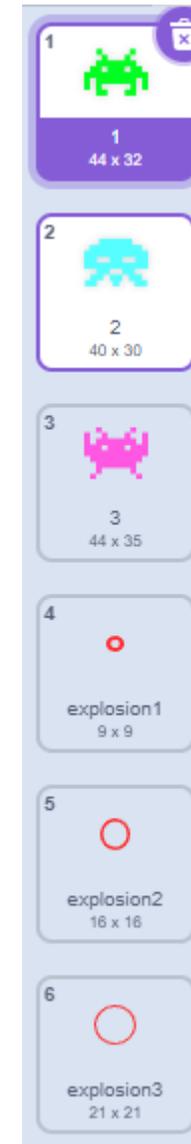
## Sprites: Invaders

I have one invader sprite with several costumes so we have some variety.

I used blocks to create Invader costumes that look like the ones in the classic game. The guide squares in the costume editor are useful when you are doing this. You can create something similar or use any sprites you want.

If you want, you can create more than one costume for each type of Invader so they can be animated. I only have one costume per Invader type.

I also created an explosion effect for the Invaders which we'll use in the code.

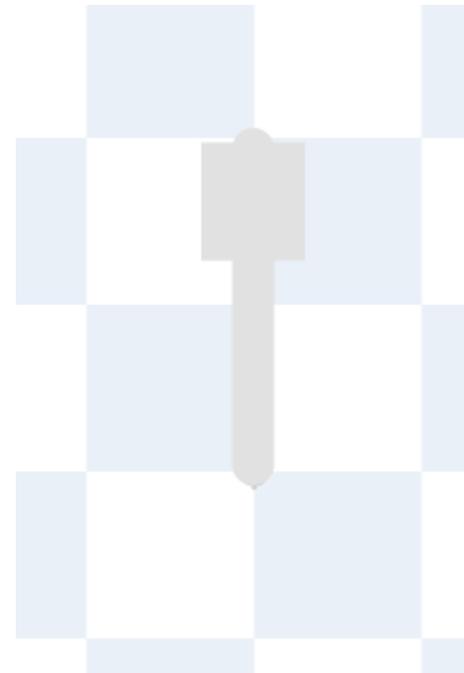


# Scratch Space Invaders

## Sprites: Invader Missile

We need sprites for the missiles that the Defender and the Invaders shoot at each other.

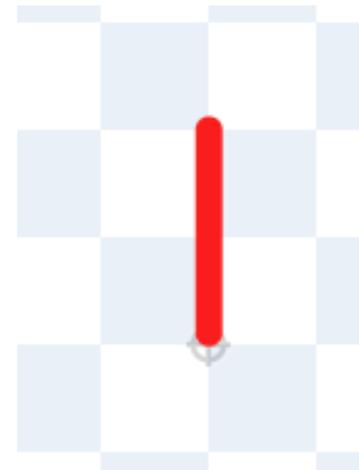
This is the one for the Invader missile. It's a light colour because I'm going to use a dark backdrop.



# Scratch Space Invaders

## Sprites: Defender Missile

My Defender missile is a very simple sprite. This time it's red to match my Defender sprite and so that it can be seen against the dark backdrop.

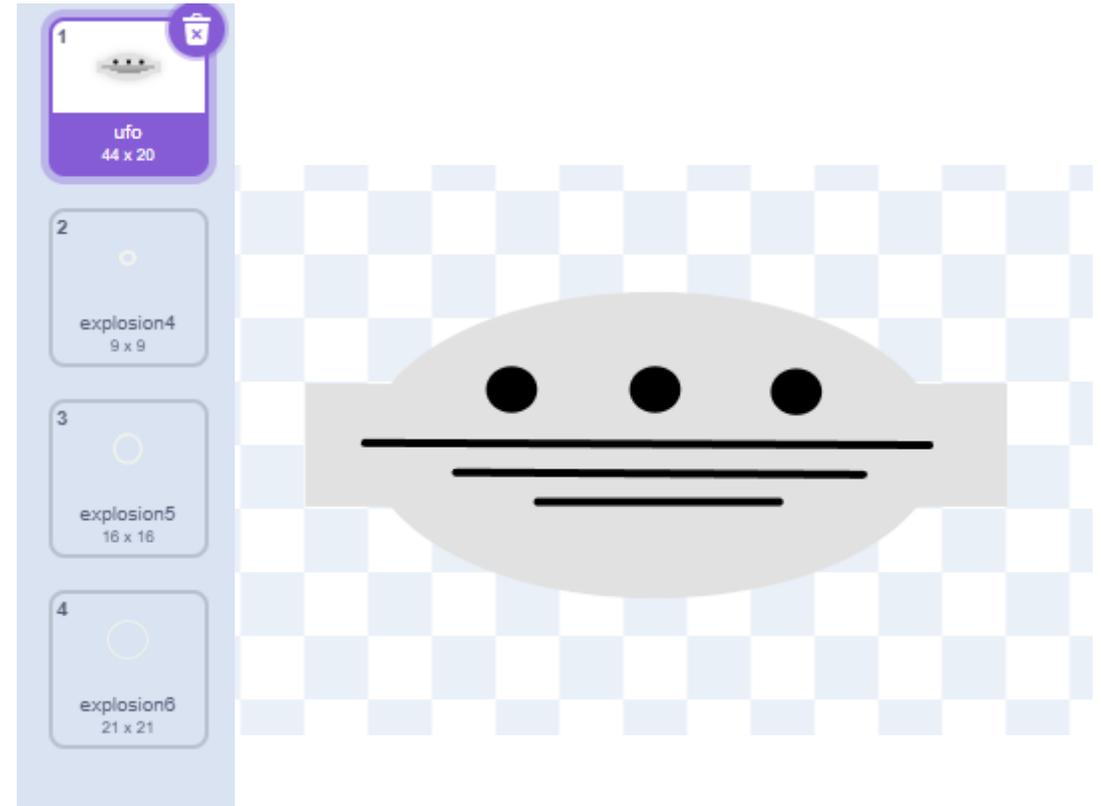


# Scratch Space Invaders

## Sprites: UFO

I added an invader UFO sprite that we can use to score bonus points.

I added costumes to create an explosion effect in the same way as for the invader sprite.

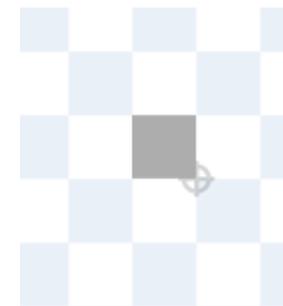


# Scratch Space Invaders

## Sprite: Defences

In the classic game, the defender has some fortifications that it can hide behind. However, the fortifications can be damaged by both the invader and the defender missiles...

The sprite for the defences is very simple....it is just a 4x4 square. In the code, we'll use the basic sprite to build up the fortifications using clones so that they can be damaged like the ones in the classic game.



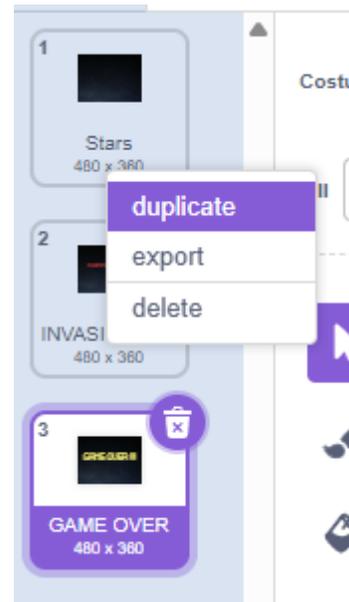
# Scratch Space Invaders

## Backdrop

We also need to make backdrops for the Stage.

I have one plain space backdrop that you can find in scratch, then versions for winning and one for losing the game. You can use the duplicate tool in the costume editor to create the different versions.

I used the pixel font to look like the classic arcade game.



# Scratch Space Invaders

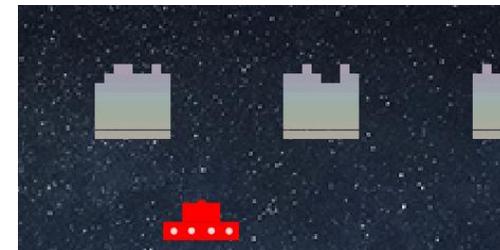
Building defences

# Scratch Space Invaders

## Defences

In the classic game, the fortifications can be damaged by the Defender and Invader missiles. How can we create something similar?

In scratch we can create clones of sprites. I decided to use the basic 'rock' sprite and then build up fortification shapes using clones.



# Scratch Space Invaders

## Building a fortification

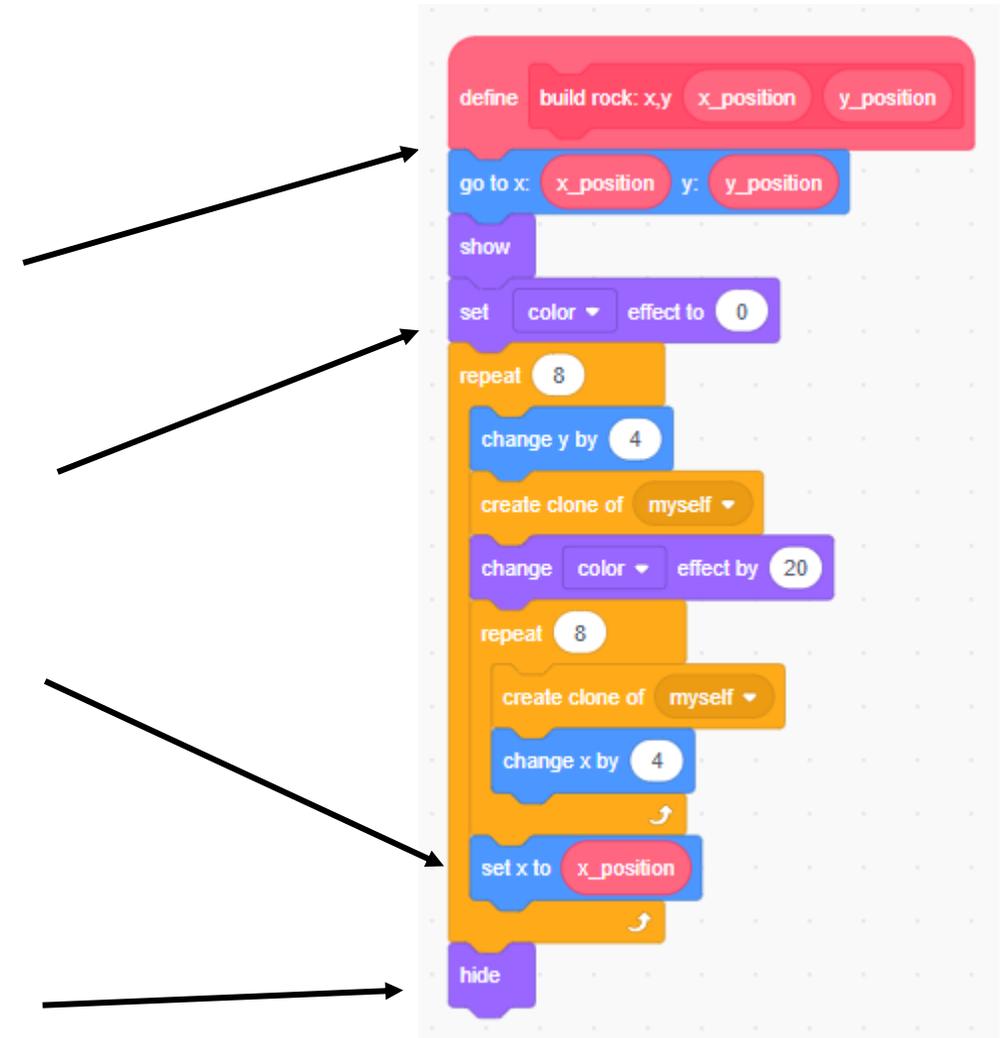
First we need to create a single fortification. I used a code block so we can re-use it. We need to tell scratch where we want to place the fortification, so we pass the screen coordinates (x,y) to the code block, then go to that position.

I decided to build my fortification using 8x8 blocks. I use two repeat loops to do this. I create a clone, then move 4 to the right (**change x by 4**) create another clone and so on until I have a row of 8 blocks.

Then, I send the block back to the starting x position (**set x to x\_position**), move 4 up (**change y by 4**) and create another row. I repeat this 8 times so eventually I have an 8x8 block.

The loop to change x is inside the loop to change y so that we create each row then start the next one. You can change the repeat values to make different fortification shapes

Finally I hide the 'rock' sprite so only the clones that we have created are visible.



```

define build rock: x,y x_position y_position
go to x: x_position y: y_position
show
set color effect to 0
repeat 8
  change y by 4
  create clone of myself
  change color effect by 20
  repeat 8
    create clone of myself
    change x by 4
  set x to x_position
hide
  
```

The code block is a Scratch script with the following steps:

- define build rock: x,y x\_position y\_position** (Function definition)
- go to x: x\_position y: y\_position** (Move to the specified coordinates)
- show** (Make the sprite visible)
- set color effect to 0** (Reset the color effect)
- repeat 8** (Outer loop for rows)
  - change y by 4** (Move up 4 pixels)
  - create clone of myself** (Create a clone)
  - change color effect by 20** (Change the color effect)
  - repeat 8** (Inner loop for columns)
    - create clone of myself** (Create a clone)
    - change x by 4** (Move right 4 pixels)
  - set x to x\_position** (Reset x position for the next row)
- hide** (Hide the original sprite)

# Scratch Space Invaders

## More fortifications

Now we have the code block that builds a single fortification, we can create as many as we like by calling the code block.

When we receive the broadcast to start a new level (we'll look at this later), we clear any old sprites from the screen, then call the code block as often as we want. Each new fortification will be built at the (x,y) position we send to the code block.

I created four fortifications across the screen, so each starts at a different x position but the same y\_position.

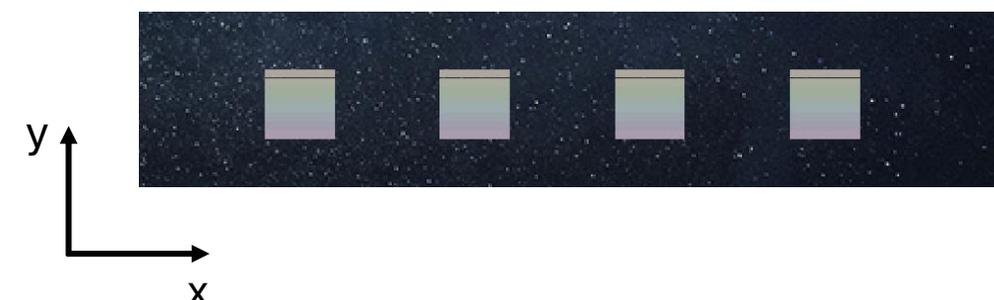
Once the fortifications are built, the game can begin. We use another broadcast message to tell the sprites the game is starting.

I used some colour effects to show layers to make the fortifications look a bit more interesting.



```

when I receive new level
  clear graphic effects
  go to back layer
  build rock: x,y -136 -120
  build rock: x,y -56 -120
  build rock: x,y 24 -120
  build rock: x,y 104 -120
  repeat 3
    play sound Alert until done
  broadcast start
  
```



# Scratch Space Invaders

Coding sprites

# Scratch Space Invaders

## Coding sprites: Defender

Let's code our Defender first.

The game starts once the fortifications have been built. `

The Defender can move from left to right and shoot missiles at the Invaders. We need to code the movement and controls. We also need to code how the Defender behaves when it's hit by an Invader missile.

We'll give the Defender 5 lives (or as many as you want). Each time the Defender is hit, it loses a life. If all its lives are lost....game over!

# Scratch Space Invaders

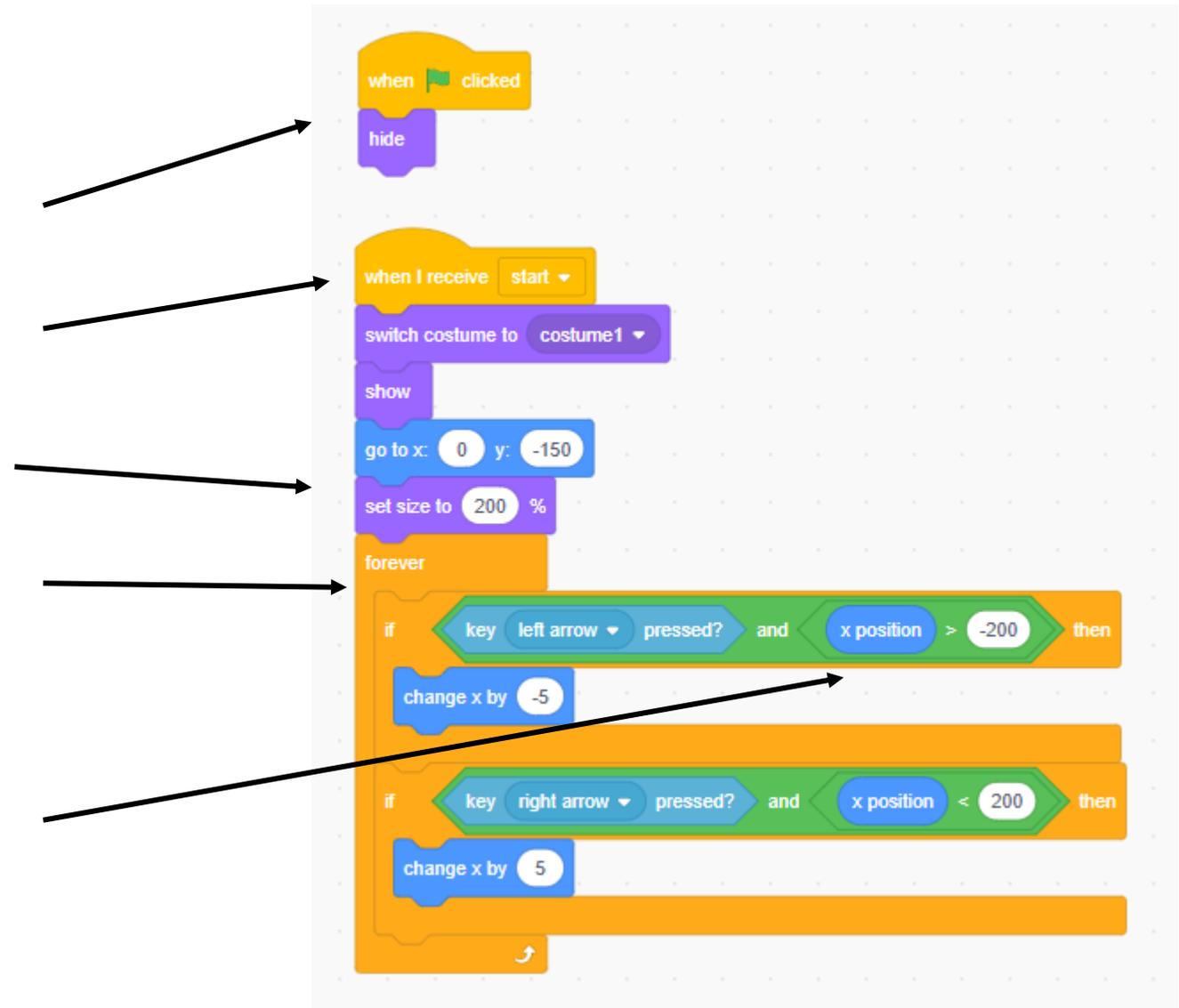
## Coding sprites: Defender

We'll hide the Defender sprite while the fortifications are being built. When the Defender receives the start broadcast message starts, it is placed on the screen in the centre (x=0).

I made a size adjustment so that it looks right.

Now we add the controls to move the Defender left and right inside a forever loop. We use the sensing blocks in scratch to decide when and which direction to move.

We also check for the Defender's x position on the screen so it can't go off the edges.



```

when clicked
hide

when I receive start
switch costume to costume1
show
go to x: 0 y: -150
set size to 200 %

forever
if key left arrow pressed? and x position > -200 then
change x by -5
if key right arrow pressed? and x position < 200 then
change x by 5
  
```

The image shows a Scratch code editor with the following blocks for a sprite named 'Defender':

- when clicked** block containing a **hide** block.
- when I receive start** block containing:
  - switch costume to costume1**
  - show**
  - go to x: 0 y: -150**
  - set size to 200 %**
- forever** loop containing:
  - if key left arrow pressed? and x position > -200 then** block containing a **change x by -5** block.
  - if key right arrow pressed? and x position < 200 then** block containing a **change x by 5** block.

Arrows from the text on the left point to these specific parts of the code:

- Arrow 1: Points to the **when clicked** block.
- Arrow 2: Points to the **when I receive start** block.
- Arrow 3: Points to the **set size to 200 %** block.
- Arrow 4: Points to the **forever** loop block.
- Arrow 5: Points to the **if key left arrow pressed? and x position > -200 then** block.

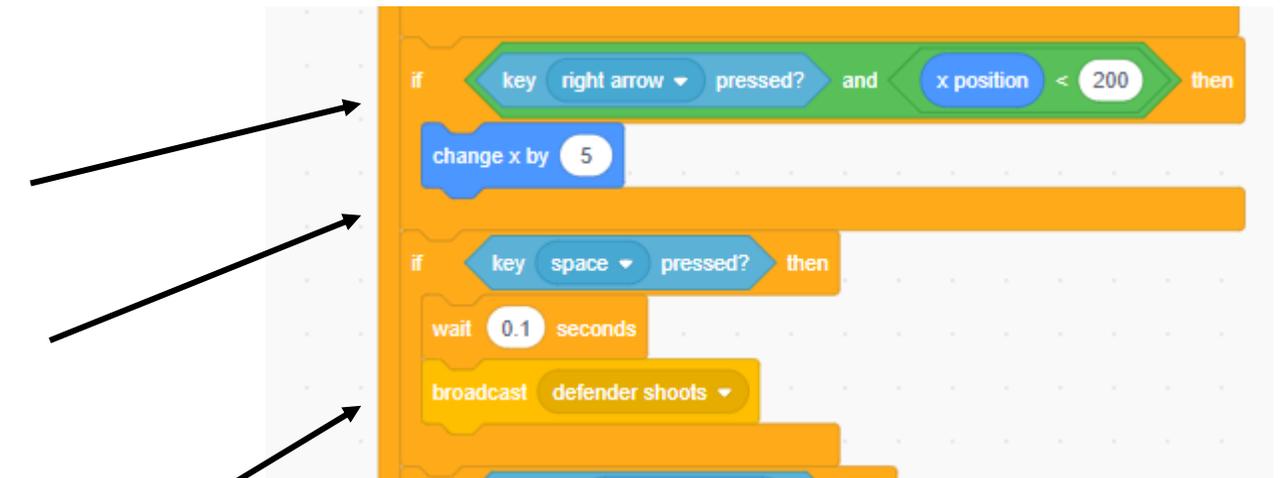
# Scratch Space Invaders

## Coding sprites: Defender

We can also add code inside our forever loop so that the Defender can shoot its missile.

I added a wait control just to make sure there is a pause between missile shots. You can try different values here or leave it out to see what difference it makes.

The Defender missile is a separate sprite so we use another broadcast message to tell it whenever the Defender shoots.



The image shows a Scratch script for a Defender sprite. It consists of the following blocks:

- An **if** block with the condition: `key right arrow pressed?` and `x position < 200`.
- Inside the first **if** block, a **change x by** block with the value `5`.
- A second **if** block with the condition: `key space pressed?`.
- Inside the second **if** block, a **wait** block set to `0.1 seconds`.
- Below the wait block, a **broadcast** block with the message `defender shoots`.

Three black arrows point from the text on the left to the corresponding blocks in the code: the first arrow points to the first **if** block, the second arrow points to the **wait** block, and the third arrow points to the **broadcast** block.

# Scratch Space Invaders

## Coding sprites: Defender

What happens if the Defender is hit by an Invader missile? We can use a sensing block to test if this happens.

If there is a hit, we can play a sound and send the Defender back to the middle position. We also change the Lives variable by -1 to keep track of how many lives the Defender has left.

If the Lives variable reaches 0....game over! We send a broadcast message that the game is over and stop the script from running.

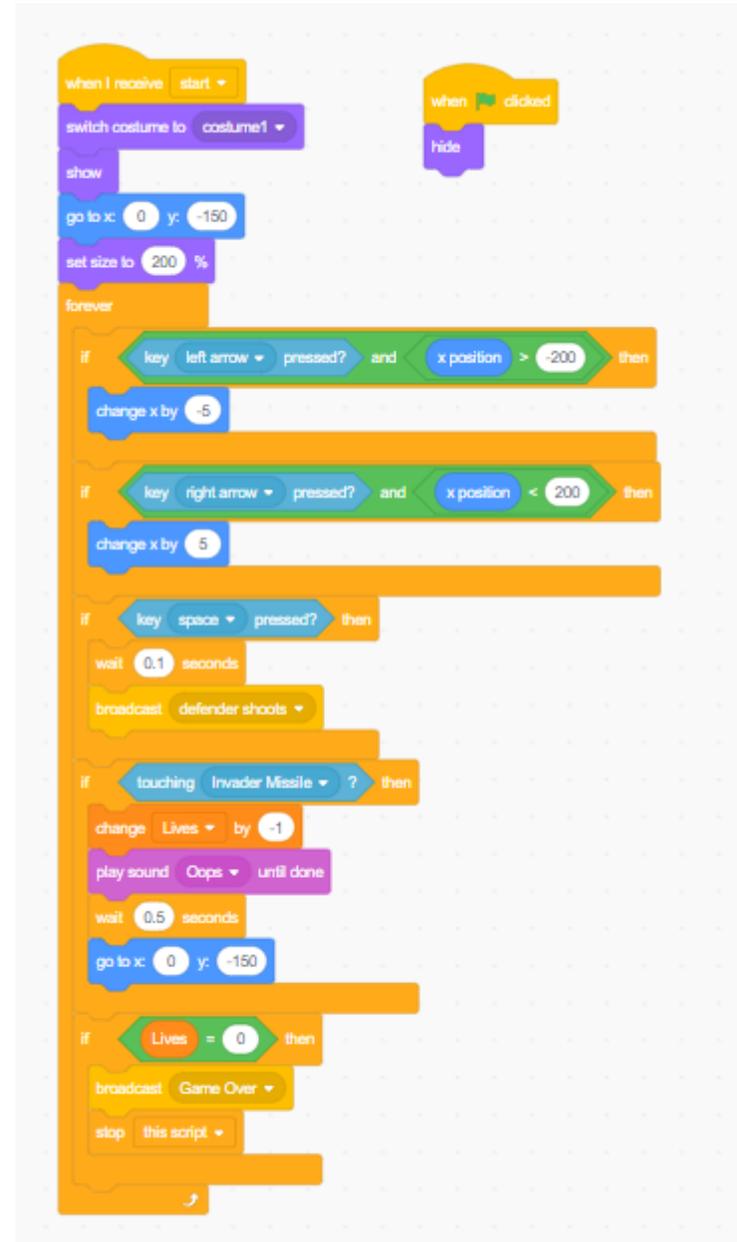
```

if touching Invader Missile? then
  change Lives by -1
  play sound Oops until done
  wait 0.5 seconds
  go to x: 0 y: -150
if Lives = 0 then
  broadcast Game Over
  stop this script
  
```

# Scratch Space Invaders

## Coding sprites: Defender

The complete code for the Defender looks like this.



```

when I receive start
  switch costume to costume1
  show
  go to x: 0 y: -150
  set size to 200 %

when clicked
  hide

forever
  if key left arrow pressed? and x position > -200 then
    change x by -5
  if key right arrow pressed? and x position < 200 then
    change x by 5
  if key space pressed? then
    wait 0.1 seconds
    broadcast defender shoots
  if touching Invader Missile ? then
    change Lives by -1
    play sound Oops until done
    wait 0.5 seconds
    go to x: 0 y: -150
  if Lives = 0 then
    broadcast Game Over
    stop this script
  
```

# Scratch Space Invaders

## Coding sprites: Invaders

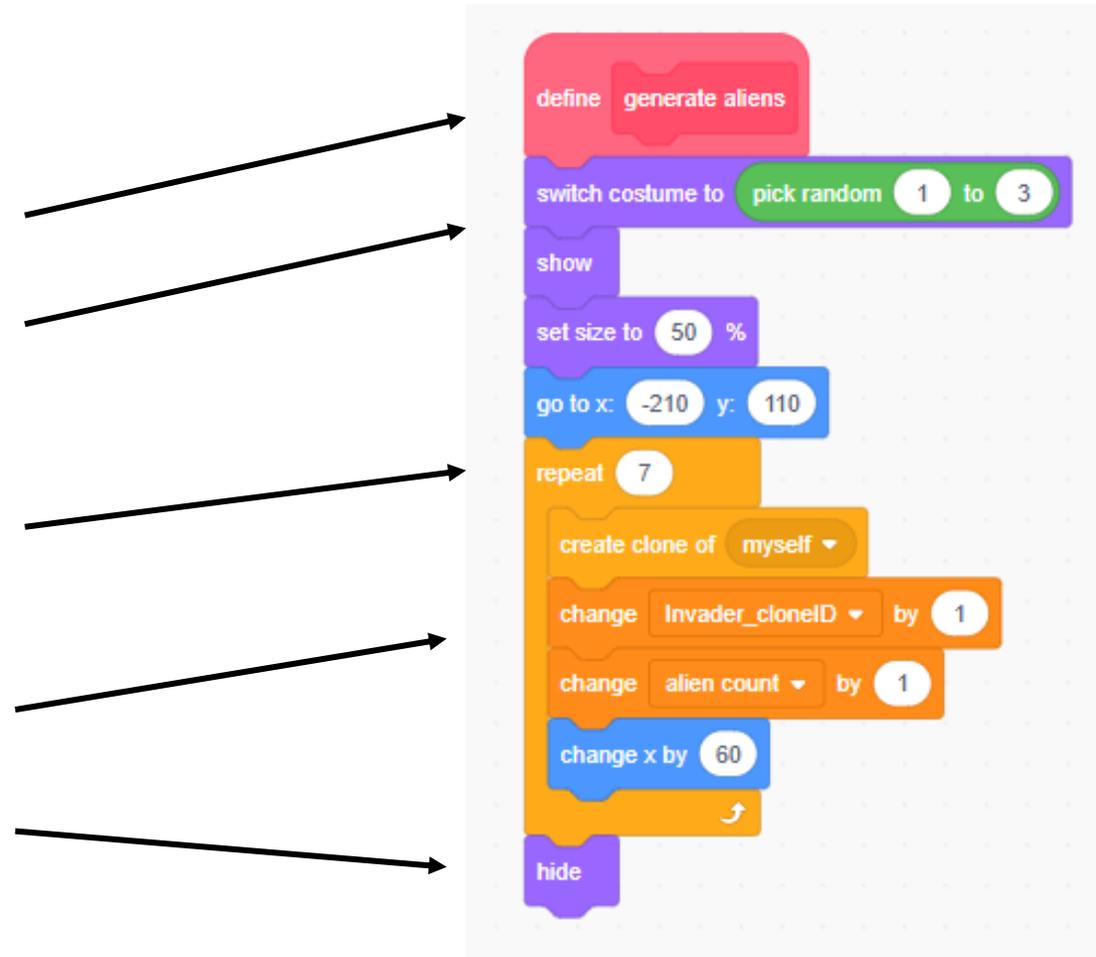
We need to generate lots of Invaders, so we'll use a code block.

I use a random operator to choose one of the three available Invader costumes (look back to see that the costume names are numbers so this works....1, 2 and 3)

We'll generate a row of 7 invaders each time. We create a clone of the invader sprite then move to the right and repeat until we have 7.

We use two variables here. Invader\_CloneID allows us to make individual clones do things. Alien Count is used to track how many invaders there are on the screen. We'll see how these are used later.

When we've generated the row of clone invaders, we hide the Invader sprite.



```

define generate aliens
  switch costume to pick random 1 to 3
  show
  set size to 50 %
  go to x: -210 y: 110
  repeat 7
    create clone of myself
    change Invader_cloneID by 1
    change alien count by 1
    change x by 60
  hide
  
```

# Scratch Space Invaders

## Coding sprites: Invaders

Now we have to add code so our Invader clones know what to do. First we'll code their movements.

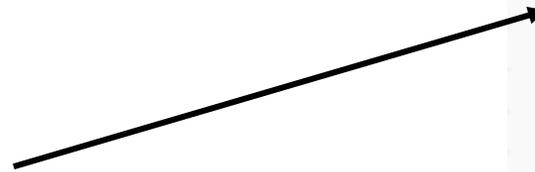
In a forever loop, we'll have them move to the right, then move down.

Next, they move back to the left, and move down. This is done with repeat loops. We can adjust the movement and wait values to speed the clones up or to slow them down until we're happy with how they behave.

In this way, the Invader clones will gradually move down the screen towards the Defender. We test to see if they have reached the Defender's part of the screen, and if they have, the game is over! We broadcast the game over message and stop the script if this happens.

```

when I start as a clone
  forever
    repeat 10
      change x by 5
      wait 0.1 seconds
    change y by -5
    repeat 10
      change x by -5
      wait 0.1 seconds
    change y by -5
    if y position < -100 then
      broadcast Game Over
      stop this script
  
```



# Scratch Space Invaders

## Coding sprites: Invaders

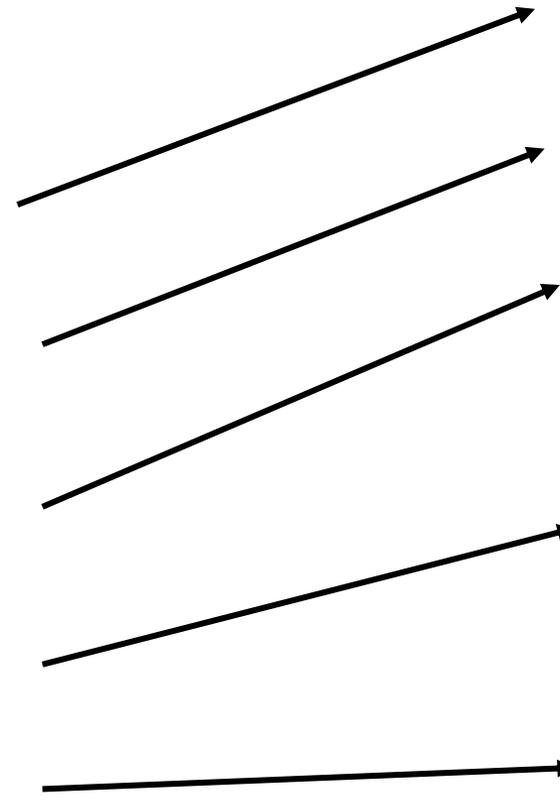
What happens if an Invader clone is hit by a Defender missile? We can use a separate 'When I start as a clone' block for this.

In a forever loop, we use a sensing block in an if block to test if there is a hit.

If there is, we update our score and alien count variables. Score goes up, alien count goes down!

We can then use the additional costumes to create an explosion animation if we want.

Finally, we delete the clone so that it disappears.



```

when I start as a clone
  forever
    if touching Defender Missile ? then
      change Score by 1
      change alien count by -1
      switch costume to explosion1
      wait 0.1 seconds
      switch costume to explosion2
      wait 0.1 seconds
      switch costume to explosion3
      delete this clone
  
```

# Scratch Space Invaders

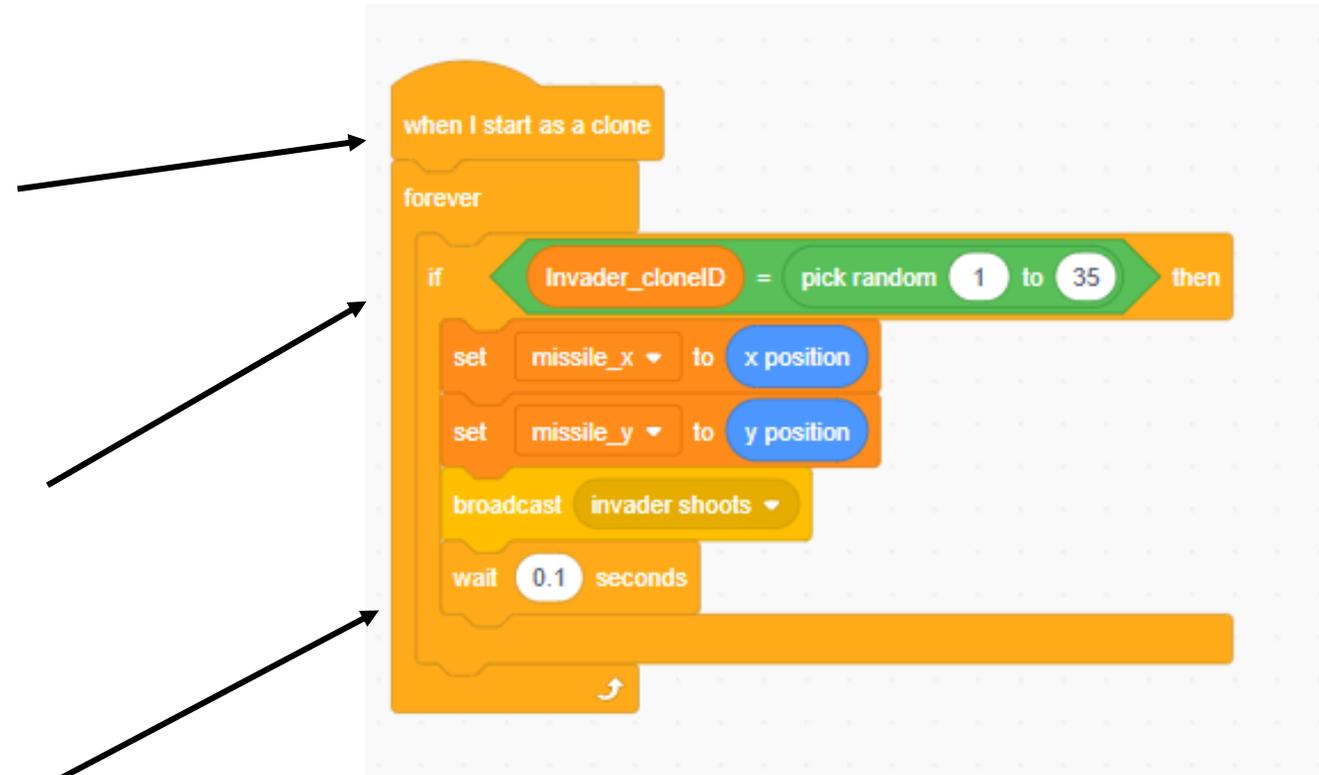
## Coding sprites: Invaders

There's one other thing the Invaders can do....shoot missiles! We use a another separate 'When I start as a clone' block for this.

Remember that we gave each clone an Invader\_CloneID when they were created. We can use this to make the clones shoot missiles in a random way.

Every time we pass through the forever loop (and remember each clone is doing this all the time!) we shoot a missile if the random number selected matches the clone's Invader\_CloneID. If this happens, we broadcast a message. I have five waves of 7 Invaders per wave, so I'll pick a random number between 1 and 35 so all Invaders can shoot missiles.

The wait block is used to slow down or speed up the rate at which missiles will be shot to make the game play well.



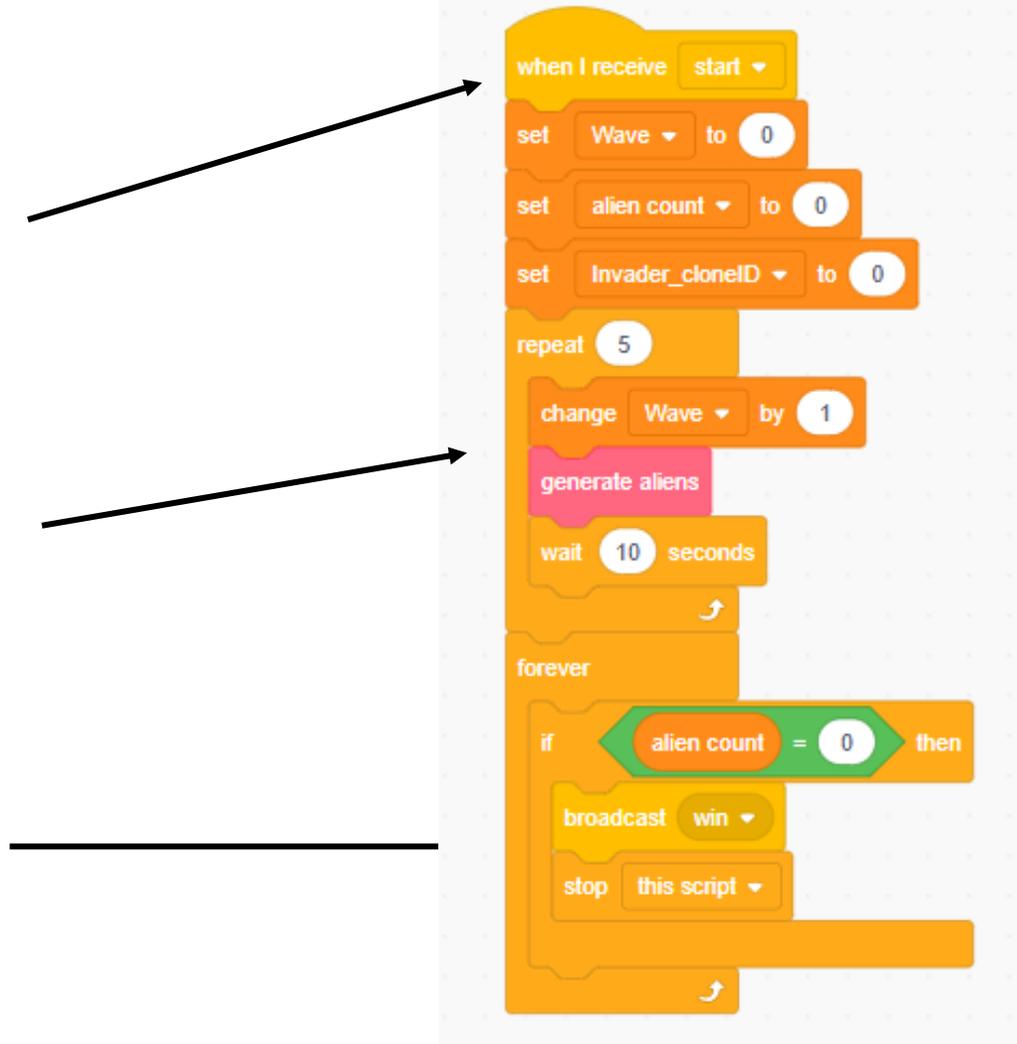
# Scratch Space Invaders

## Coding sprites: Invaders

Now we need the code to pull all of the Invader actions together. We hide the Invader sprite to begin with. When we receive the start broadcast, we set all of the variables to their initial values.

Next, we call the Generate Aliens block once for each wave of Invaders that we want. I have 5 waves of Invaders here. After each wave is generated we give them time to move down the screen before generating the next wave.

We use the alien\_count variable to test if all of the Invaders have been destroyed. If this happens, the Defender wins the game!



The code for the Invader sprite is as follows:

```

when clicked
  hide

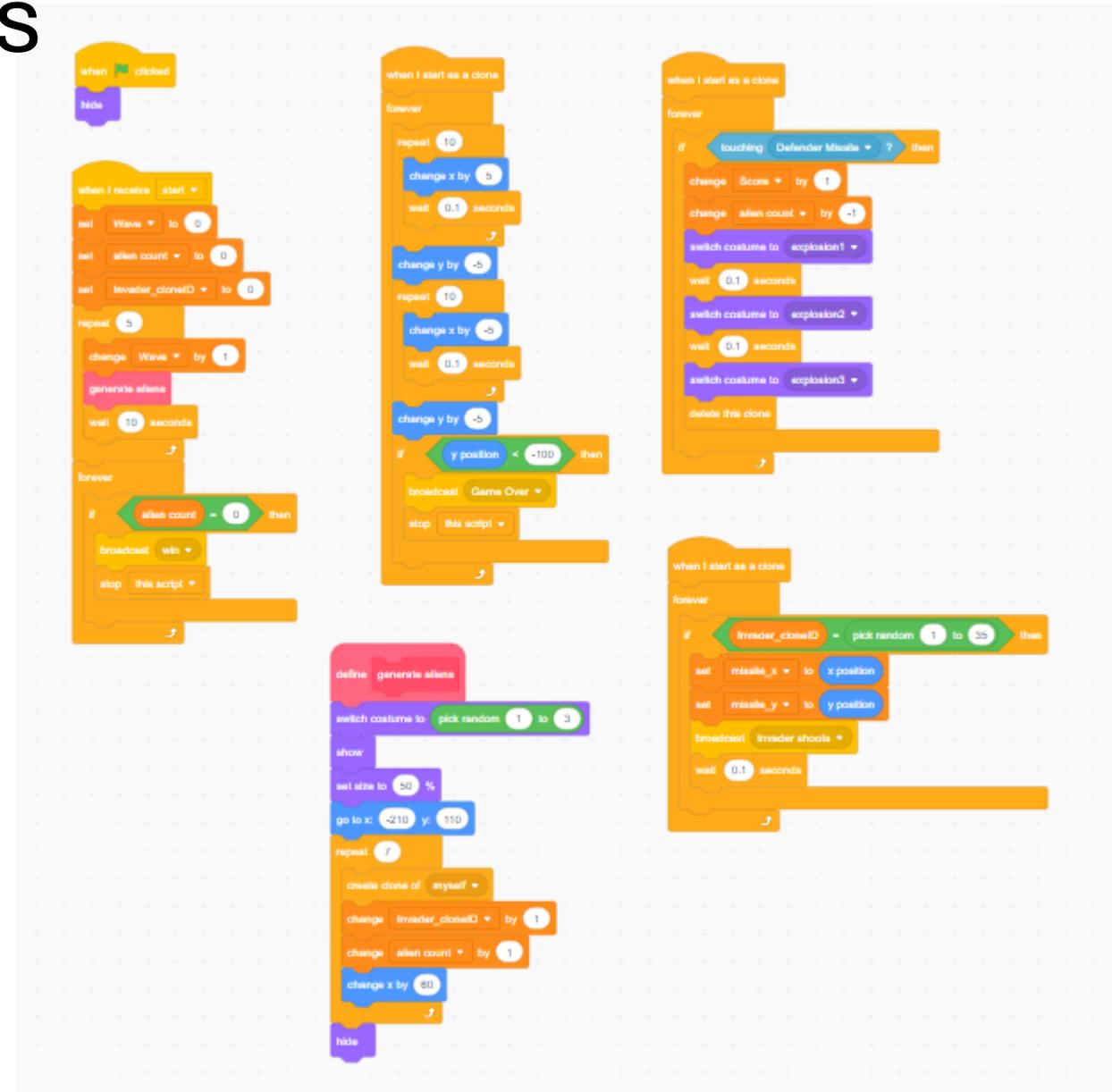
when I receive start
  set Wave to 0
  set alien count to 0
  set Invader_cloneID to 0
  repeat 5
    change Wave by 1
    generate aliens
    wait 10 seconds
  forever
    if alien count = 0 then
      broadcast win
      stop this script
  
```

Arrows in the image point from the text to the corresponding code blocks: the first arrow points to the 'when I receive start' block, the second arrow points to the 'generate aliens' block, and the third arrow points to the 'if alien count = 0' block.

# Scratch Space Invaders

## Coding sprites: Invader

The complete code for the Invaders looks like this.



```

when clicked
  hide

when I receive 'start'
  set Wave to 0
  set alien count to 0
  set invader_cloneID to 0
  repeat 5
    change Wave by 1
  generate aliens
  wait 10 seconds
  forever
    if alien count = 0 then
      broadcast 'win'
      stop this script

when I start as a clone
  forever
    repeat 10
      change x by 5
      wait 0.1 seconds
    change y by -5
    repeat 10
      change x by -5
      wait 0.1 seconds
    change y by -5
    if y position < -100 then
      broadcast 'Game Over'
      stop this script

define generate aliens
  switch costume to 'pick random 1 to 3'
  show
  set size to 50 %
  go to x: -210 y: 110
  repeat 7
    create clone of myself
    change invader_cloneID by 1
    change alien count by 1
    change x by 80
  hide

when I start as a clone
  forever
    if invader_cloneID = pick random 1 to 35 then
      set missile_x to x position
      set missile_y to y position
      broadcast 'invader shoots'
      wait 0.1 seconds
    if touching 'Defender Missile' then
      change Score by 1
      change alien count by -1
      switch costume to 'explosion1'
      wait 0.1 seconds
      switch costume to 'explosion2'
      wait 0.1 seconds
      switch costume to 'explosion3'
      delete this clone
  
```

# Scratch Space Invaders

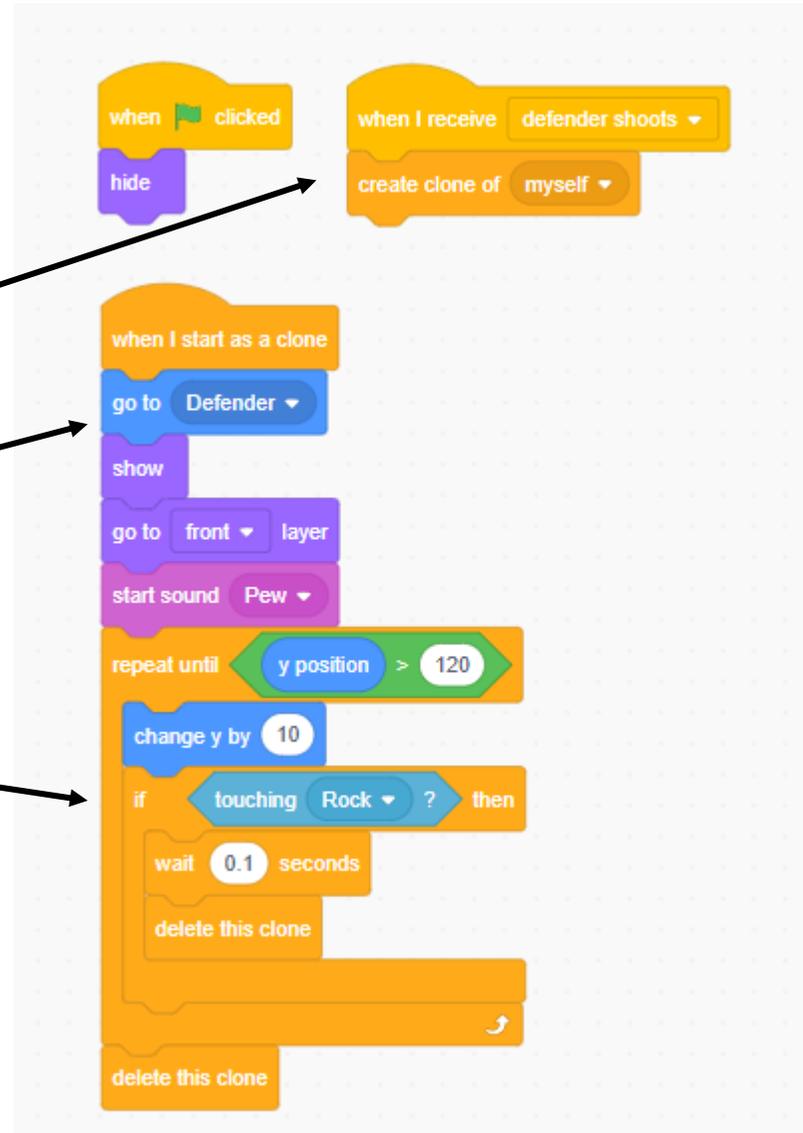
## Coding sprites: Defender Missile

We use clones for the Defender missile too so we can have many on the screen at the same time. A clone is created when the broadcast message is received.

Each clone starts at the Defender's current position, then simply moves up the screen until it reaches the top ( $y > 120$ ) using a repeat until block.

If the Defender shoots and the missile hits a fortification, the missile clone stops. The wait command is used to make sure the fortification sprites can work correctly as we will see.

Use sound effects to make your game more exciting!



```

when clicked
  hide
  when I receive defender shoots
    create clone of myself

when I start as a clone
  go to Defender
  show
  go to front layer
  start sound Pew

repeat until y position > 120
  change y by 10
  if touching Rock? then
    wait 0.1 seconds
    delete this clone
delete this clone
  
```

# Scratch Space Invaders

## Coding sprites: Invader Missile

The invader missile works in a very similar way, using clones that respond to a broadcast message.

This time, the missile starts at the (x,y) position of the Invader clone that generated the message. This is done using variables for the x and y positions of the clone when the missile is fired that can then be used by the Invader missile sprite.

If the Invader missile hits a fortification, the missile clone stops. The wait command is used to make sure the fortification sprites can work correctly as we will see.

Use sound effects to make your game more exciting!

```

when clicked
  hide

when I receive invader shoots
  create clone of myself

when I start as a clone
  go to x: missile_x y: missile_y - 5
  show
  go to front layer
  start sound Zoop
  repeat until y position < -160
    change y by -5
  if touching Rock? then
    wait 0.1 seconds
    delete this clone
  delete this clone
  
```

# Scratch Space Invaders

## Coding sprites: UFO

The UFO is hidden initially, then appears at a random time after the game has started.

When it appears, it moves in a random way across the top of the screen so it is hard to hit. We use random operators to decide how far and how fast the UFO moves.

```

when clicked
  go to x: -240 y: 130
  switch costume to ufo
  hide

when I receive start
  wait pick random 30 to 120 seconds
  start sound Space Noise
  show
  repeat pick random 3 to 8
    glide 1 secs to x: pick random -240 to 240 y: 130
  hide
  
```

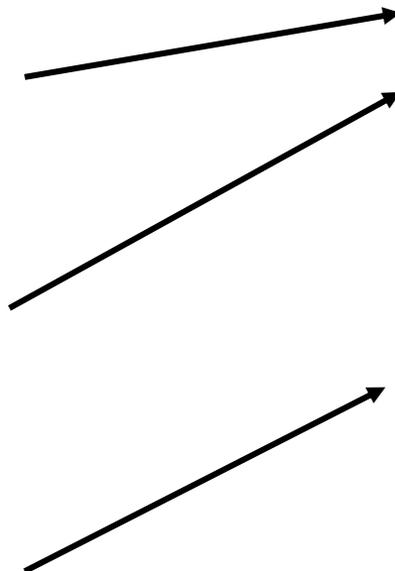
# Scratch Space Invaders

## Coding sprites: UFO

We have to test whether the UFO is hit by a Defender missile. We use a separate block of code for this with a forever loop, an if block and a sensing block.

It is hard to hit the UFO so you get a big score if you do!

When the UFO is hit by a Defender missile, we use an explosion animation in the same way as we did for the Invader sprite then hide the UFO sprite for the rest of the game.



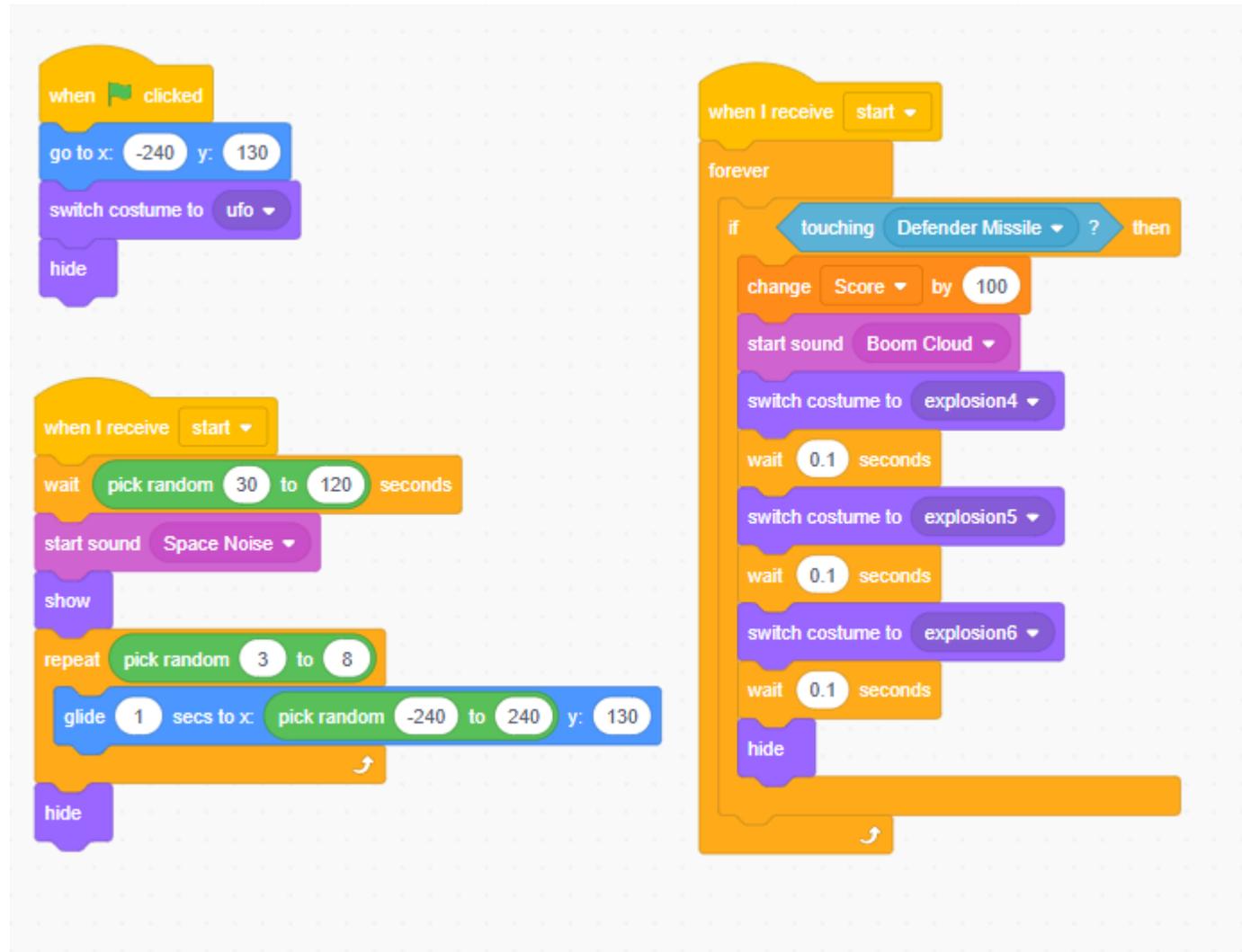
```

when I receive start
  forever loop
    if touching Defender Missile then
      change Score by 100
      start sound Boom Cloud
      switch costume to explosion4
      wait 0.1 seconds
      switch costume to explosion5
      wait 0.1 seconds
      switch costume to explosion6
      wait 0.1 seconds
      hide
  
```

# Scratch Space Invaders

## Coding sprites: UFO

The complete code for the UFO looks like this.



```

when clicked
  go to x: -240 y: 130
  switch costume to ufo
  hide

when I receive start
  wait pick random 30 to 120 seconds
  start sound Space Noise
  show
  repeat pick random 3 to 8
    glide 1 secs to x: pick random -240 to 240 y: 130
  hide

when I receive start
  forever
    if touching Defender Missile ? then
      change Score by 100
      start sound Boom Cloud
      switch costume to explosion4
      wait 0.1 seconds
      switch costume to explosion5
      wait 0.1 seconds
      switch costume to explosion6
      wait 0.1 seconds
      hide
  
```

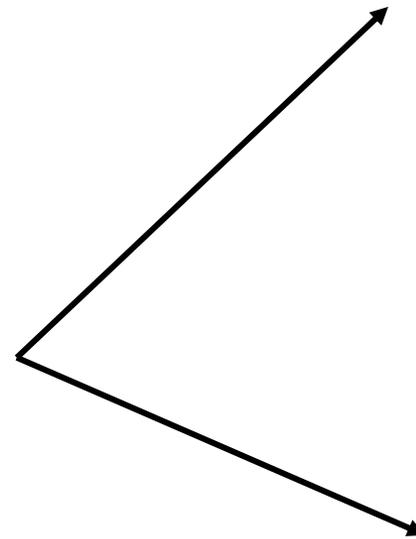
# Scratch Space Invaders

## Coding sprites: Rocks

The rock sprite is the base for the fortifications we built. We already added the code to build the fortifications. We need to add some more code to the rock sprite now that we have our Defender and Invader missiles.

We use if blocks and sensing blocks inside forever loops to test for each rock sprite clone being hit by a missile.

If any clone is hit by a missile, it disappears. This gives the effect we want of the fortification being gradually destroyed.



```

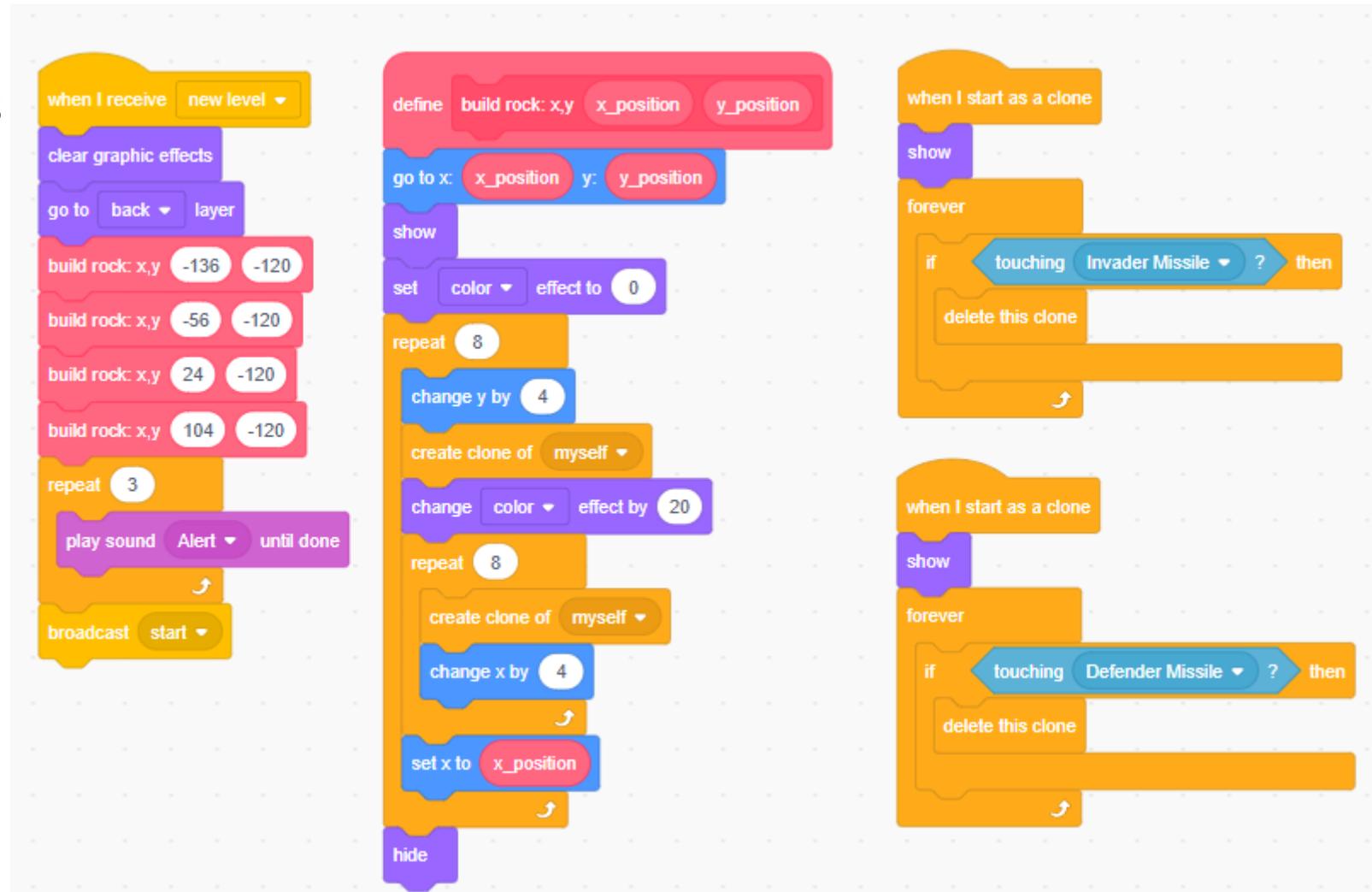
when I start as a clone
  show
  forever
    if touching Invader Missile ? then
      delete this clone
  forever

when I start as a clone
  show
  forever
    if touching Defender Missile ? then
      delete this clone
  forever
  
```

# Scratch Space Invaders

## Coding sprites: Rocks

The complete code for the rock sprite looks like this.



```

when I receive new level
  clear graphic effects
  go to back layer
  build rock: x,y -136 -120
  build rock: x,y -56 -120
  build rock: x,y 24 -120
  build rock: x,y 104 -120
  repeat 3
    play sound Alert until done
  broadcast start

define build rock: x,y x_position y_position
  go to x: x_position y: y_position
  show
  set color effect to 0
  repeat 8
    change y by 4
    create clone of myself
    change color effect by 20
  repeat 8
    create clone of myself
    change x by 4
    set x to x_position
  hide

when I start as a clone
  show
  forever
    if touching Invader Missile ? then
      delete this clone

when I start as a clone
  show
  forever
    if touching Defender Missile ? then
      delete this clone
  
```

# Scratch Space Invaders

Game Features

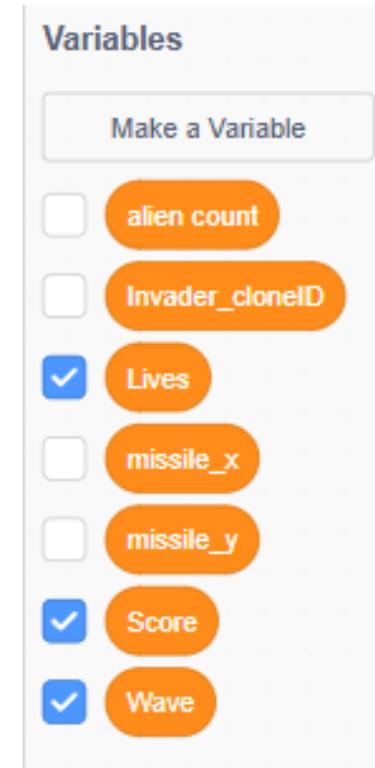
# Scratch Space Invaders

## Game Features

We have used several variables to help control how the game works.

You can use the variables control to show the **Lives**, **Score** and **Wave** variables on the screen using the check box in the variable list.

The other variables don't need to be shown as they are used to control the game. However, when you are trying to get the game to work, it can be useful to show these variables so you can see what's happening.



# Scratch Space Invaders

## Game Features

Some of the game feature variables are set in the code for the sprites that we have created. Some are set up in code for the backdrop.

I use the backdrop code to send the first message to create a new game and then to switch backdrops when the player wins or loses.



```

when clicked clicked
  clear graphic effects
  switch backdrop to Stars
  set Score to 0
  set Lives to 5
  broadcast new level

when I receive Game Over
  switch backdrop to GAME OVER
  play sound Trap Beat until done
  stop all

when I receive win
  switch backdrop to INVASION DEFEATED
  play sound Dance Space until done
  stop all
  
```

# Scratch Space Invaders

## Testing and feedback

Once you have finished, share your game with another Scratch coder and ask them to test it.

When you are testing someone else's game, look for ideas to improve the game:

- How do the sprites look?
- Do the defences generate properly?
- Is the game playable?
- What features and effects could you add?

When you have feedback, think about whether you agree with it and use it to make your game even better!

# Scratch Space Invaders

Well done!

Hopefully you now have a working game that you can share with friends.

There's always ways to make your game more interesting. You can add new features, experiment with different Invader behaviour, add sounds and effects, a player guide...

There are also different and maybe better ways to code the game. These instructions give you a start but you can think of better ways to set up the game or code your sprites.

Have fun!