

# Space Invaders Game

Instructions

# Set up the Environment

- Delete Scratchy Sprite (right click, delete)
- Create backdrop
  - Select Backdrop from tabs
  - Edit it
    - Make it completely black
    - If you want, add some stars (small white dots) to the backdrop
- Create new sprite for ship
  - Import “ship” from the USB drive
- Create missile sprite
  - Import “arrow” from the USB drive

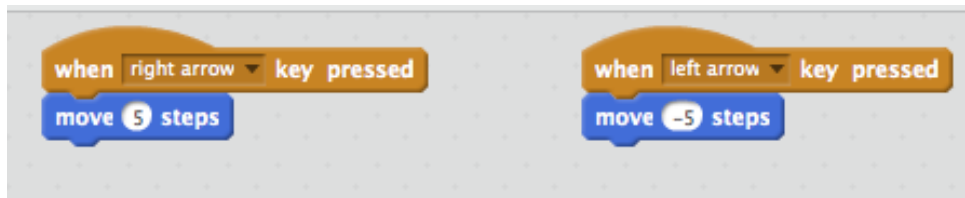
# Concepts

- Space Invaders has a lot of different objects on the screen
  - The aliens
  - The ship that shoots
  - The missile that the ship shoots
- Each item has their own set of events
  - It's kind of like each has a mind of their own!
- When you want to make an object do something, make sure you are creating events and actions in that specific sprite

# Game Play – Events for your space ship

- Right Arrow
  - Move 5 steps to the right
- Left Arrow
  - Move 5 steps to the left

# Space Ship Code



# Game Play – Events for your arrow

- When Space Bar Clicked you want the arrow to shoot up the screen
- How do you do this?
  - Create space key hit event for this sprite
  - Repeat 40 times
    - Change Y by 10
  - This moves the arrow up the screen in increments of 10
- BUT
  - You want the arrow to come out of the space ship so use a secret command
    - Go to Ship!
- You could speed it up or slow it down
  - How?

# Arrow Code



# Creating an Alien

- Import Sprite Invader1 from the USB drive
- Place on top left of screen



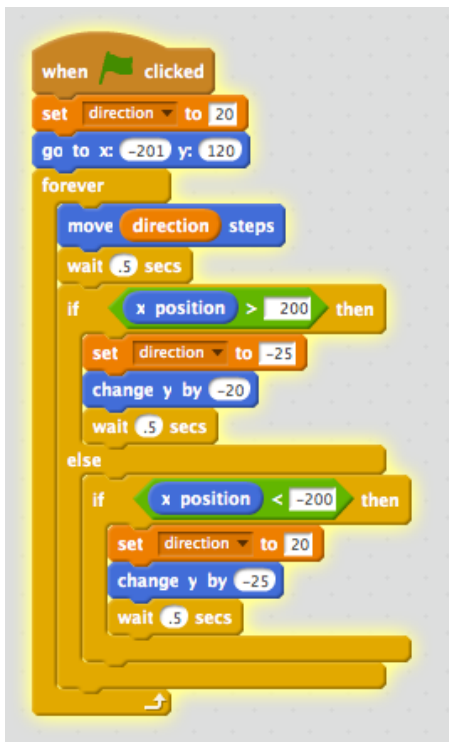
# Moving the Alien

- Aliens move back and forth on their own so
  - You need to *programmatically* determine their direction
  - Create a variable called **direction**
- Start the Alien at position -201, 120
- Set **direction** = 10 (that's 10 steps to the right)
- In a forever loop
  - If the x position > 200, move the Alien down 25 spaces and change the direction to -10 (that's 10 steps to the left)
  - Likewise, if the x position < 200, move the Alien down 25 spaces and change the direction to 10 (that's 10 steps to the right)

# Moving the Alien

- When you run this, the Alien goes across the screen in a very smooth way
- This will make shooting him hard so
- Let's add a pause after he moves
  - Command is wait
  - Do it for .5 seconds
- You might want to also wait when you move the alien down a row

# Moving the Alien - Code



# Shooting the Alien – SCORE!

- When you shoot the arrow at the Alien, you should score a point
- The Alien should also die and disappear from the screen
- Steps
  - Create a variable called *score* and *initialize* it to 0 when the game starts
  - In the Alien sprite, create a script that is executed when the flag is clicked (yes, you now have 2 flag clicked events)
    - Check to see if the arrow and Alien are touching and if they are
      - *Increment* the score by 1
      - Hide the Alien sprite

# Score! - Code

```
when green flag clicked
  set direction to 20
  set Score to 0
  go to x: -201 y: 120
  forever
    move direction steps
    wait .5 secs
    if x position > 200 then
      set direction to -25
      change y by -20
      wait .5 secs
    else
      if x position < -200 then
        set direction to 20
        change y by -25
        wait .5 secs
  
```

The image displays two Scratch code snippets. The left snippet is a 'when green flag clicked' event handler. It sets the character's direction to 20, initializes a 'Score' variable to 0, and moves the character to coordinates (-201, 120). A 'forever' loop follows, where the character moves 'direction' steps every 0.5 seconds. An 'if' statement checks the 'x position'. If it is greater than 200, the direction is set to -25, the y-position is decreased by 20, and it waits 0.5 seconds. Otherwise, another 'if' statement checks if the 'x position' is less than -200. If true, the direction is set to 20, the y-position is decreased by 25, and it waits 0.5 seconds. The right snippet is also a 'when green flag clicked' event handler. It contains a 'forever' loop with an 'if' statement checking 'touching arrow?'. If true, the character is hidden and the 'Score' variable is increased by 1.

# Add More Aliens!

- Duplicate or Clone the Aliens
- But
  - You will need to move the new Aliens to a new starting position
    - What X position (recommend they are 50 spaces apart and 25 spaces on the Y axis)
- Create a few rows
- What happens?

# Personalize Your Game

- Want to stop here and perfect your game play?
- Do the aliens move the way you want them to move?
  - If not, change it now
- Does the missile move fast (or slow) enough?
  - If not, change it now
- Do you have enough aliens?
- Too close together?
- Too far apart?
- Make the game your own!

# Taking Your Game to the Next Level

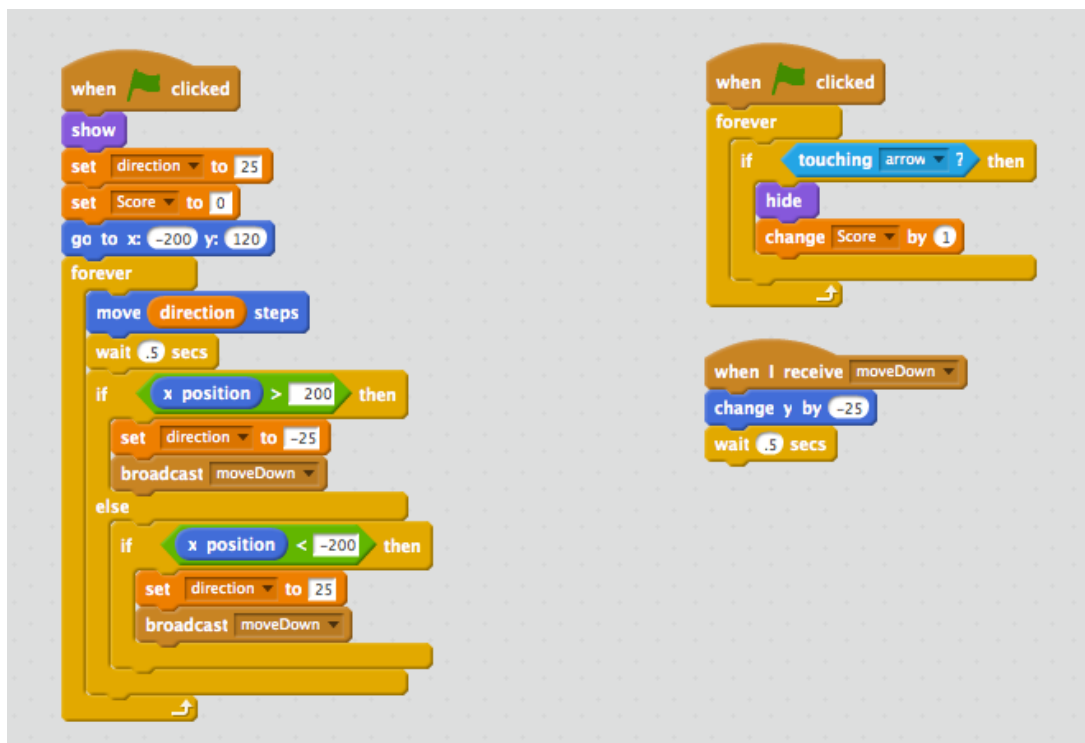
Advanced Concepts



# Making Everyone Do the Same Thing at the Same Time...Broadcast

- When you put multiple rows of Aliens on your screen, they **centipede** down the screen. We need our Aliens to all move at the same time!
- Do this using a **Broadcast** message
- Create a Broadcast called `moveDown`
- Instead of moving down and changing direction, simply call `moveDown`

# The moveDown Event



# But Something is Wrong...

- The Aliens are going crazy on the screen!
- Why?
  - Aliens do not change direction until they read the end of the screen
  - So you still get crazy behavior
- How do you fix this?

# Fixing the Aliens - Code

The image displays two columns of Scratch code blocks. The left column shows the initialization and movement logic for an alien. It starts with a 'when clicked' event that triggers a 'show' block, followed by setting 'direction' to 25 and 'Score' to 0. The alien then moves to x: -200, y: 120. A 'forever' loop follows, where the alien moves 'direction' steps, waits for 0.5 seconds, and checks its x position. If the x position is greater than 200, it broadcasts 'moveDown'. If the x position is less than -200, it also broadcasts 'moveDown'. The right column shows the response to the 'moveDown' broadcast. It starts with a 'when clicked' event that enters a 'forever' loop. Inside this loop, it checks if the alien is 'touching arrow'. If true, it hides the alien and increases the 'Score' by 1. Below this, a 'when I receive moveDown' event triggers an 'if' statement. If the 'direction' is 25, it sets 'direction' to -25; otherwise, it sets 'direction' to 25. Finally, it changes the y position by -25 and waits for 0.5 seconds.

```
when clicked
  show
  set direction to 25
  set Score to 0
  go to x: -200 y: 120
  forever
    move direction steps
    wait .5 secs
    if x position > 200 then
      broadcast moveDown
    else
      if x position < -200 then
        broadcast moveDown

when clicked
  forever
    if touching arrow ? then
      hide
      change Score by 1

when I receive moveDown
  if direction = 25 then
    set direction to -25
  else
    set direction to 25
  change y by -25
  wait .5 secs
```

## Problem 2 – Magic Bullets!

- If you have multiple rows of Aliens, one bullet seems to kill more than one Alien. Let's make the game harder.
- Once you hit an Alien, the bullet needs to stop
- Create a Broadcast called hitAlien and call it when an Alien is hit
  - In the Arrow scripts, add a variable called stopFire. When this is set to 0, let the bullet continue. When it is set to 1, the bullet will stop.
  - When we receive the hitAlien Broadcast, set stopFire to 1
  - In the space key event
    - Show the Arrow
    - Set stopFire to 0
    - In the repeat, check if stopFire is 1. If it is, hide the bullet and stop this script.

# Fixing the Magic Bullet - Code

```
when green flag clicked
  forever loop
    if touching arrow? then
      broadcast hitAlien
      hide
      change Score by 1
```

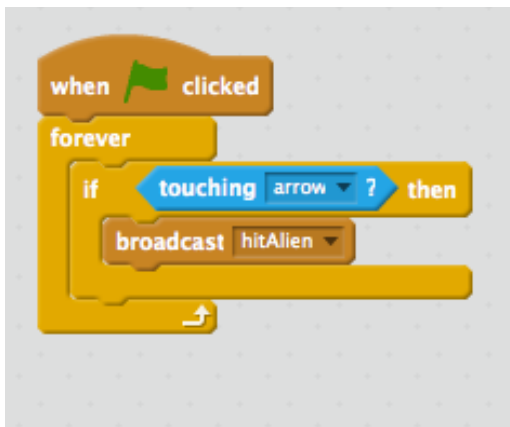
```
when space key pressed
  show
  go to ship
  set stopFire to 0
  repeat 40
    if stopFire = 1 then
      hide
      stop this script
  change y by 10
```

```
when I receive hitAlien
  set stopFire to 1
```

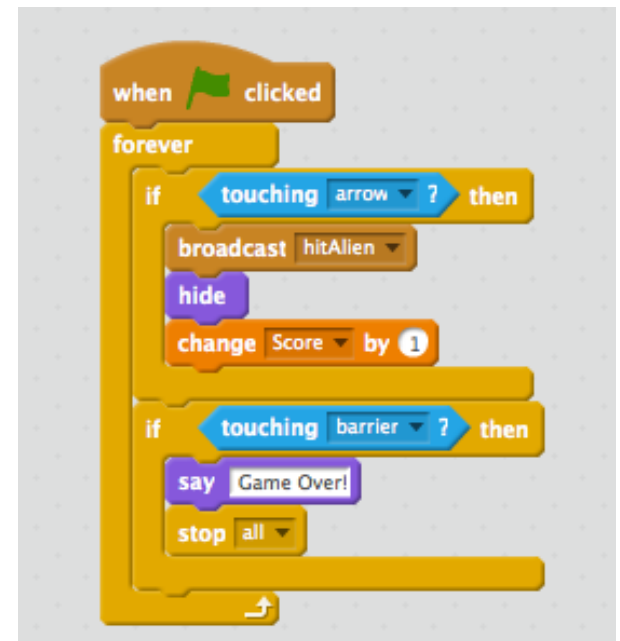
# Adding Barriers

- Let's add some barriers so we can hide from Alien fire!
- A barrier will protect us from Alien fire (will absorb the missile) but will also absorb fire from the ship
- Use Sprite barrier from the USB drive
  - Place on screen
- Adjust arrow to sense the barriers
  - If it is touching the arrow, stop the bullet, just like you did when you hit an Alien!

# Barriers - Code



```
when clicked clicked
forever
  if touching arrow ? then
    broadcast hitAlien
```



```
when clicked clicked
forever
  if touching arrow ? then
    broadcast hitAlien
    hide
    change Score by 1
  if touching barrier ? then
    say Game Over!
    stop all
```



# What if the Alien Could Shoot?

- Let's make the Aliens drop bombs
  - If a bomb hits the ship, the game is over
- How to do this
  - Import sprite alienFire from the USB drive
  - The alienFire sprite will receive two broadcasts that you will create
    - alienFire (when an Alien fires a missile – this happens every time it moves)
    - stopAlienFire (when the missile hits something)
  - The Barrier sprite will need to broadcast stopAlienFire when it touches the bomb
  - The Ship sprite needs to broadcast stopAlienFire when it touches the bomb and it needs to end the game

# Code Samples



```
when I receive alienFire
  go to invader1
  set stopAlienFire to 0
  show
  repeat 70
    if stopAlienFire = 1 then
      hide
      stop this script
    change y by -10
```

when I receive stopAlienFire  
set stopAlienFire to 1



```
when clicked
  forever
    if touching arrow ? then
      broadcast hitAlien
    if touching alienFire ? then
      broadcast stopAlienFire
```



```
when clicked
  show
  forever
    if touching alienFire ? then
      hide
      stop all
```

# More Code Samples



```
when clicked
show
set direction to 25
set Score to 0
go to x: -200 y: 120
forever
  move direction steps
  wait 0.5 secs
  if x position > 200 then
    broadcast moveDown
  else
    if x position < -200 then
      broadcast moveDown
  broadcast alienFire
```

The image shows a Scratch code block starting with a 'when clicked' event. It includes a 'show' block, followed by 'set direction to 25', 'set Score to 0', and 'go to x: -200 y: 120'. A 'forever' loop contains 'move direction steps', 'wait 0.5 secs', and an 'if' statement. The 'if' statement has two branches: one for 'x position > 200' and another for 'x position < -200', both leading to a 'broadcast moveDown' block. Finally, a 'broadcast alienFire' block is placed at the end of the loop.

# Ghost Aliens!

- We made it so one alien is shooting, but
  - What happens if that alien is shot?
    - He still shoots! He's a ghost shooter!
- How do we fix this?
  - Create a variable called `alienVisible` and set it to 1
  - When the alien is shot (and hidden) change this to 0
  - Before you fire, check `alienVisible` and ONLY shoot if it equals 1

# The Code

```
when clicked
show
set direction to 25
set Score to 0
set alienVisible to 1
go to x: -200 y: 120
forever
  move direction steps
  wait 0.5 secs
  if x position > 200 then
    broadcast moveDown
  else
    if x position < -200 then
      broadcast moveDown
  if alienVisible = 1 then
    broadcast alienFire
```

```
when clicked
forever
  if touching arrow ? then
    set alienVisible to 0
    change Score by 1
    broadcast hitAlien
    hide
  if touching barrier ? then
    say Game Over!
    stop all
```

# Personalize It

- Which aliens should be shooting at the ship?
  - Change this around if you want

# Other Things to Try

- There are more graphics on the USB drive
  - Want to change the alien's appearance as they move across the screen?
- Add sounds
- Give your ship multiple lives and end the game once all lives have been used
- Add more barriers