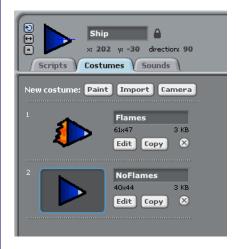
Home Chorus Dance Orchestra Robotics Technology

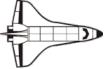
Asteriod Directions

Steps for Creating the SpaceShip:

1. Draw a SpaceShip. Make sure you point the ship to the right. Make two costumes, one with flames and one without.



Option: You may use one of these images for your Space Ship:



Space Shuttle



Space Station



Lunar Module

2. Create the following Variables: (For This Sprite Only)

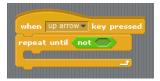
DirectionChange NewX NewY Thrust



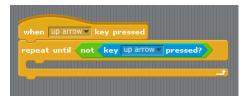
- 3. We need to create three event Scripts ("Up Arrow", "Right Arrow", and "Left Arrow")
- 4. Drag a "when Space Key pressed" and a "repeat until" into the Ship Script Area and change to "Up Arrow."



5. Put a green operator "not " into the repeat until.



6. Go to sensing and put a "key up arrow pressed" into the "not " $\mbox{block}.$



7. Put a "next costume" in the repeat until.

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume
```

8. Put three "change DirectionChange" blocks into the repeat until.

Change the variables to "Thrust," "NewX," "NewY."

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume
change Thrust by 10
change NewX by 10
change NewY by 10
```

9. Put a value of "0.1" in the change Thrust Block

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume

change Thrust by 0.1

change NewX by 10

change NewY by 10
```

10. Drag two "multiplication" operators into the script area.

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume
change Thrust by 0.1
change NewY by 10
change NewY by 10
```

11. Put the Variable "Thrust" into the multiplication blocks.

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume
change Thrust by (0.1)
change NewX by 10

Thrust *

Thrust *
```

12. Put "sqrt " block into each multiplication block.

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume

change Thrust by 0.1

change NewX by 10

change NewY by 10

Thrust * sqrt of 10

Thrust * sqrt of 10
```

13. Change one of the "sqrt " blocks to "sin"

```
Thrust * Sin v of 10
```

14. Change the other "sqrt" block to "cos"

```
Thrust * COS ▼ of 10
```

15. Put the Thrust * sin block into the change NewX block

```
when up arrow key pressed?

repeat until not key up arrow pressed?

next costume
change Thrust by (),1
change NewX by Thrust * Sin of 10
change NewY by 10
```

16. Put the Thrust * cos block into the change NewY block

```
when up arrow key pressed?

repeat until not key up arrow pressed?

next costume

change Thrust by 0.1

change NewX by Thrust * Sin of 10

change NewY by Thrust * Cos of 10
```

17. Go to motion blocks and put a blue direction block into the \sin and \cos blocks.

18. Put a "switch to costume NoFlames" underneath the repeat until.

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume
change Thrust by 0.1

change NewX by Thrust * SIN of direction
change NewY by Thrust * COS of direction

switch to costume NoFlames
```

19. Put a "set Thrust to 0" under the "switch costume."

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume
change Thrust by 0.1

change NewX by Thrust * Sin of direction

change NewY by Thrust * Cos of direction

switch to costume NoFlames

set Thrust to 0
```

- 20. Now that we have finished the "Up Arrow" Scripts we need to do the "right" and "left" arrow scripts.
- 21. Put the following Blocks for "right arrow"

```
when right arrow | key pressed

repeat until | not | key right arrow | pressed?

change | DirectionChange | by | 0.2

if | | abs | of | DirectionChange | < .5

set | DirectionChange | to | 0
```

22. Put the following Blocks for "left arrow"

```
when left arrow key pressed

repeat until not key left arrow pressed?

change DirectionChange by -0.2

if abs of DirectionChange < 5

set DirectionChange to 0
```

- 23. Up, Right, and Left represent the User controls for the Ship. However, the ship will not move until we put the "Action" code for the computer to take the commands ,and data and translate it to movement.
- $24. \ \mbox{Use}$ the following Green Flag script to move the ship around the screen:

```
when clicked

forever

go to x: x position + NewX y: y position + NewY

turn DirectionChange degrees
```

25. Add this Green Flag script to reset the variables and allow the ship to "Pass Through" the edges of the screen:

```
when clicked

switch to costume NoFlames

set Thrust to []

set NewY to []

set NewY to []

set DirectionChange to []

forever

if x position > 235

set x to -234

if y position > 175

set y to -174

if y position < -175

set y to 174
```

26. Click the Green flag and test your ship!

Steps for Creating the "Ray"

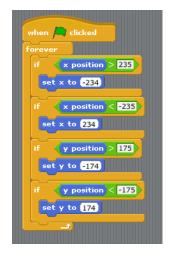
- 27. We want the "Ray" to travel to a Target and then stop traveling when it hits a target. We also want the Ray to go back to the Ship and Hide when not being fired.
- 28. Click "Create New Sprite" and draw a "Ray"
- 29. Create the following Variables for "Ray"



30. Put the following blocks for a Green Flag event for "Ray" (This allows the Ray to travel a set distance and check its status (fire or not fired)



31. Put these scripts to allow the Ray to "Pass Through" the edges:



32. Use these Scripts for the "Space Key" event:

```
when Space key pressed
go to Ship point in direction direction of Ship show
set Fire to 1
set FireCount to 0
```

33. We will use a "When I Receive" to allow the ray to know when it has hit a target $% \left(1\right) =\left(1\right) +\left(1\right$



Scripts for Landing Pad Game

34. Add the Variable "Gravity" to the Ship. Make sure it is "for this sprite only."



35. Find the Green Flag Script for the Ship that allows it to "Pass Through" the edge of the screen. Remove the if statements containing the "y" values.

```
when clicked

switch to costume NoFlames

set Thrust to 0

set NewX to 0

set NewY to 0

set DirectionChange to 0

forever

if x position > 235

set x to -234

if x position < -235

set x to 234
```

36. The Gravity value will pull the ship towards the "ground." So we will change the Green Flag script that controls the ship's position by subtracting "Gravity" from the y position and

increasing "Gravity" by 0.05 each cycle.

```
when clicked

set Gravity to 0

forever

go to x: x position + NewX y: y position + NewY - Gravity

change Gravity by 0.05

turn  DirectionChange degrees
```

37. Add a new Green Flag Script that sets all the values to 0 when the ship's y is less than -175 (at the bottom of the screen) or if touching the red of the landing pad.

38. At the start of the game we want the ship to find a new starting position and orientation. So, add the following Green Flag script.

```
when clicked

go to x: pick random 200 to 200 y: 137

point in direction pick random 20 to 20
```

39. We now need to modify the "up arrow" event to compare "Gravity" to "NewY." We will check to see if the two values are within 2. If they are - we will reset both to 0. This keeps Gravity and NewY from getting too large. Modify the "Up arrow" Script in this way:

```
when up arrow key pressed

repeat until not key up arrow pressed?

next costume

change Thrust by O.1

change NewY by Thrust * Sin of direction

change NewY by Thrust * COS of direction

switch to costume NoFlames set Thrust to O

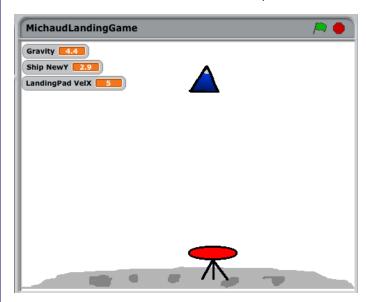
if abs of Gravity NewY < 2

set Gravity to O

set NewY to O
```

40. Create a New Sprite to be the Landing Pad. I chose to make

the platform red and have the ship check to see if it is touching the red for a proper landing. Position the Landing Pad at the bottom of the screen. I also drew a "moon" surface sprite.



41. Add these scripts to the Landing Pad so the Pad changes position each time the game starts.

```
when dicked
glide 1 secs to x: pick random 150 to 150 y: 140
```

42. Here are some Scripts for using a We-Do Tilt controller and a gimbal.

43. Save and test! Try adding more challenges like having the landing pad move about so the ship has to touch a moving target.

Email Mr.Michaud

Home | Chorus | Orchestra | Robotics | Technology | Nebo Elementary Home | Paulding County Schools

Copyright ©2011 Mr. Michaud.