

Code Club Adventures

MakeCode Arcade

Level 3 Blocks

Invasion of the Alien Ducks



Alien ducks are attacking the Earth and you are humanity's only hope with your super massive cannon! The cannon is controlled by the left and right buttons to move along the bottom of the screen. The super massive cannon can be fired with the A button. Because it is so powerful, each time you shoot the cannon you lose a life. Shooting a fruit of fortitude gives you strength and increases your lives. Humanity needs you commander, good luck!



Contents

Introduction	3
Step 1 - Humanities saviour (the cannon)	4
Step 2 - Enter the alien ducks and fruit of fortitude	8
Step 3 - Launching shells from the cannon	12
Step 4 - Shells hit alien ducks	15
Step 5 - Shells hit fruits of fortitude	16
Step 6 - Creating a function to descend sprites	17
Step 7 - Descending the fruits of fortitude	31
Step 8 - Descending the alien ducks	34
Appendix A - Explanations and Extensions	38

About the Author

Daniel Bloy is a self confessed geek with a passion for technology. Daniel fondly remembers his first escapades in coding as a young 10 year old on his beloved Commodore 64. Daniel has been a professional software developer since 1998 in Cambridge, England. Daniel loves to inspire a new generation of coders and has been running code clubs since 2016.

Acknowledgements

The technical reviewer for this book was my daughter Amelia, who never faltered in providing honest and often ruthless feedback. Watching you work through this book brought me joy!

The style used to present the material in this book has been developed through my code clubs where my students find many inventive ways to interpret my lessons and instructions. Those same students taught me that reading instructions is strictly optional. You are all inspirational.

Code Club Adventures, MakeCode Arcade, Level 3 Blocks, Invasion of the Alien Ducks

Copyright © 2023 by Daniel Bloy

Some rights reserved. This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. To view a copy of this license, visit <https://creativecommons.org/licenses/by-nc-sa/4.0/>

This book will take you through the steps to build a game using Microsoft MakeCode Arcade.

Who is this book for?

This book is suitable for anyone from 8 to 80 years old who wants to learn how to make computer games. This is a level 3 book and therefore uses both basic and advanced blocks to build a complete game but leaves scope for the reader to extend themselves through ideas provided in Appendix A. Some previous experience with blocks based programming would be advantageous but is not essential. It will typically take between 1 and 2 hours to work through the book and an additional 1 to 2 hours to complete the extension exercises.

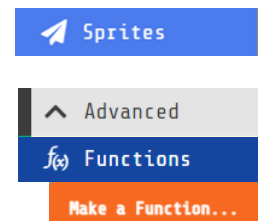
What is required to use this book?

The only items you need to use this book is an Internet browser and an Internet connection. The Microsoft MakeCode Arcade platform is free to use and can be used with an iPad or other tablet but a desktop computer such as a PC, Raspberry Pi or Apple Mac is recommended. You can download your creations to a Microsoft MakeCode Arcade compatible console but this is not necessary as the games can be played in the browser. The code in this book has been tested with the Google Chrome, Microsoft Edge and Mozilla Firefox browsers.

How to use this book?

The Microsoft MakeCode Arcade platform is available at <https://arcade.makecode.com>.

As you follow the instructions, you will see icons to the right hand side of the instructions like those shown here. These icons indicate the section of the palette where the new blocks to add can be found. If there are stacked icons, it indicates the subsection where the new blocks can be found. In the example to the right, select the “Advanced” section, then the “Functions” subsection and finally the “Make a Function...” option.



As you work through the instructions, be creative and use your imagination to improvise and extend your creations. Don't be afraid to experiment and innovate.

Appendix A contains extension ideas along with explanations about how the code works.

Where to find out more?

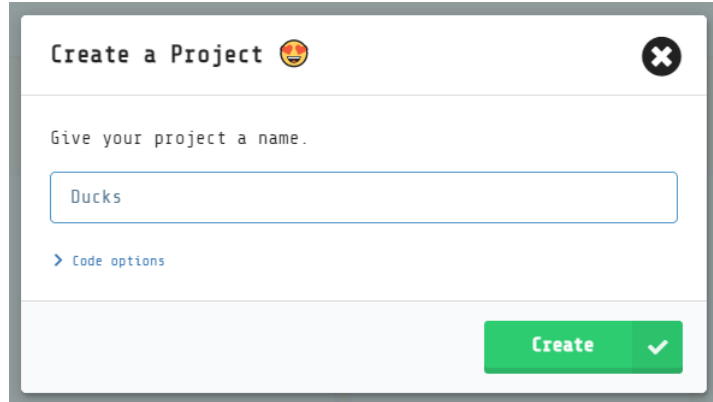
Errata and additional materials are available from the Code Club Adventures website which can be found at www.codeclubadventures.com.

Feedback

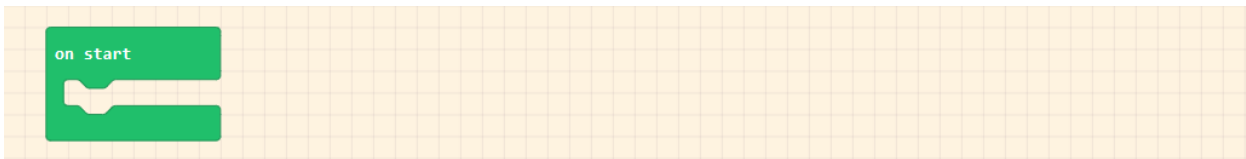
I would love to hear your comments and feedback about this book, suggestions for new ideas you would like to see or simply just about your coding adventures. You can contact me online at daniel.bloy@codeclubadventures.com.

Step 1 - Humanities saviour (the cannon)

<https://arcade.makecode.com>



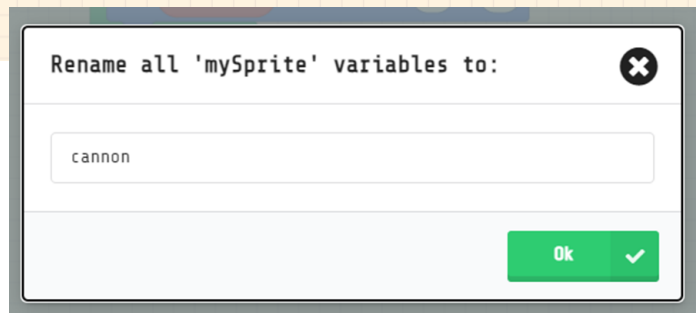
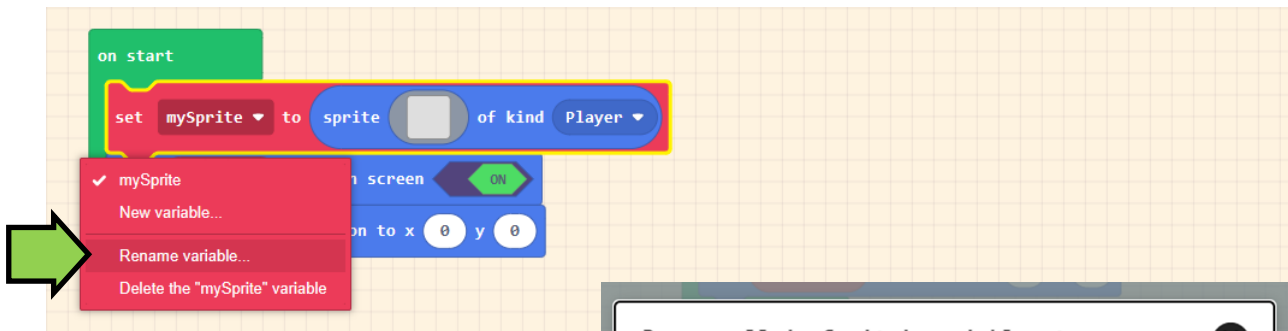
1



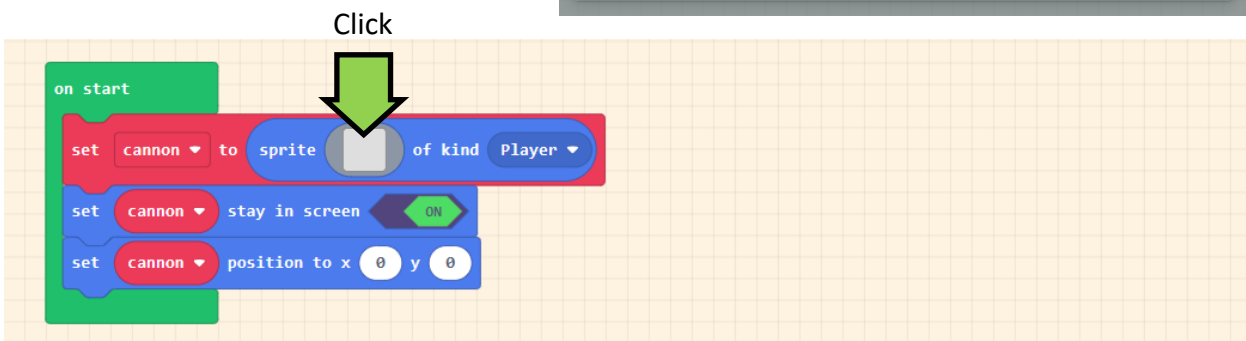
2



3

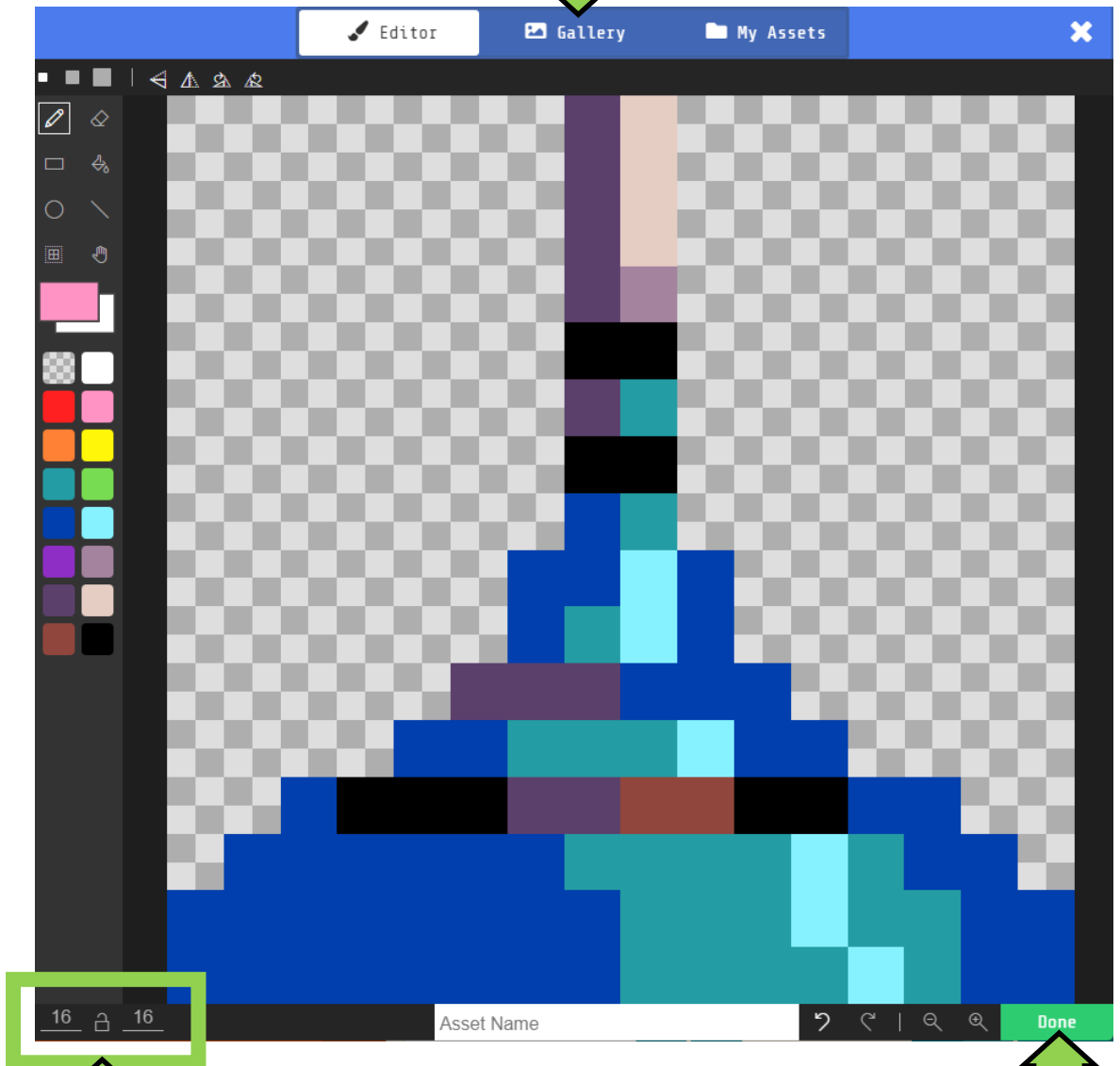


4



5

Select an image from the gallery or draw your own cannon



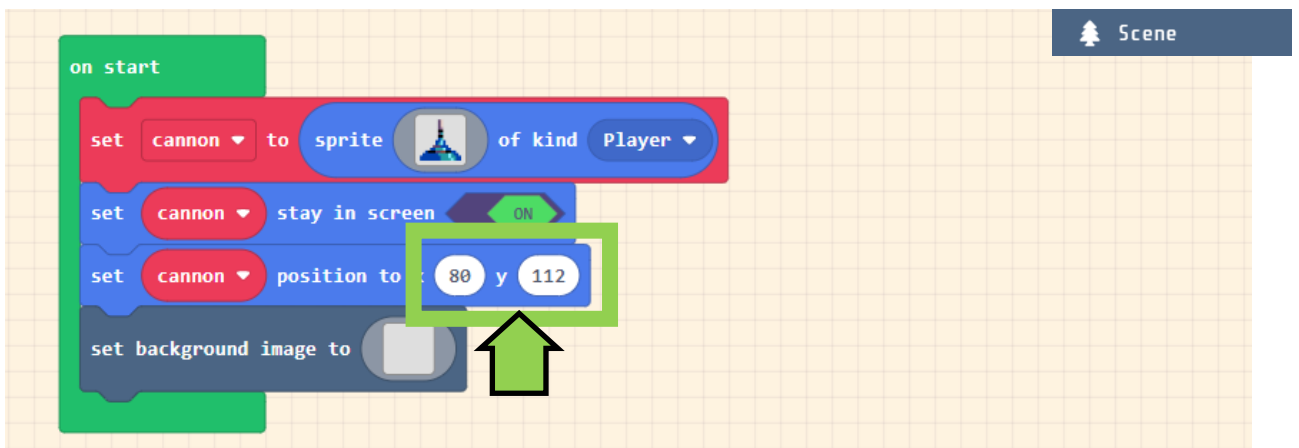
6



Size should be 16 x 16 pixels



7



```
on start
  set cannon to sprite of kind Player
  set cannon stay in screen ON
  set cannon position to x 80 y 112
  set background image to
  move mySprite with buttons +
```

Click to expand

8

```
on start
  set cannon to sprite of kind Player
  set cannon stay in screen ON
  set cannon position to x 80 y 112
  set background image to
  move mySprite with buttons vx 100 vy 0 -
```

9

```
on start
  set cannon to sprite of kind Player
  set cannon stay in screen ON
  set cannon position to x 80 y 112
  set background image to
  move cannon with buttons vx 100 vy 0 -
```

Variable selection menu:

- ✓ cannon
- mySprite
- New variable...
- Rename variable...
- Delete the "cannon" variable

10

```
on start
  set cannon to sprite of kind Player
  set cannon stay in screen ON
  set cannon position to x 80 y 112
  set background image to
  move cannon with buttons vx 100 vy 0 -
```

Click and be creative by drawing your own backdrop.

11

```
on start
  set cannon to sprite of kind Player
  set cannon stay in screen ON
  set cannon position to x 80 y 112
  set background image to
  move cannon with buttons vx 100 vy 0
  set score to 0
  set life to 20
```

12

```
on start
  set cannon to sprite of kind Player
  set cannon stay in screen ON
  set cannon position to x 80 y 112
  set background image to
  move cannon with buttons vx 100 vy 0
  set score to 0
  set life to 20
```

13

Step Done

Your game should now look like this and you will be able to move your cannon left and right. Try it out.



Step 2 - Enter the alien ducks and fruit of fortitude

on game update every 500 ms

Game



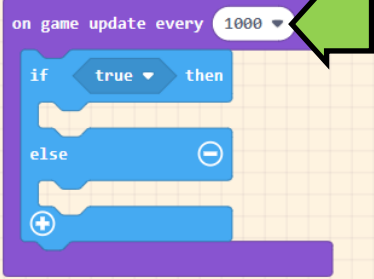
1

on game update every 1000 ms

Logic

if true then

else



2


on game update every 1000 ms

Logic

if true then

0 = 0

else



3


on game update every 1000 ms

Math

if 0 = 0 then

pick random 0 to 10

else



4

on game update every 1000 ms

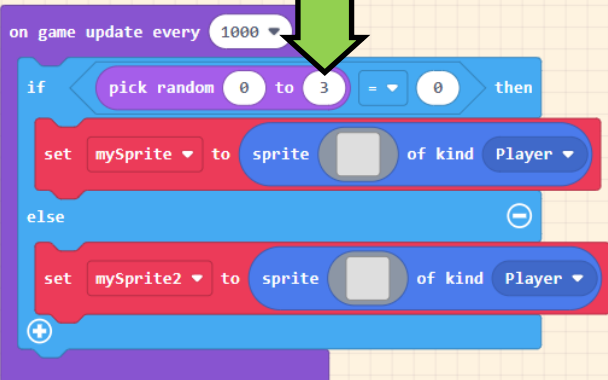
Sprites

if pick random 0 to 3 = 0 then

set mySprite to sprite of kind Player

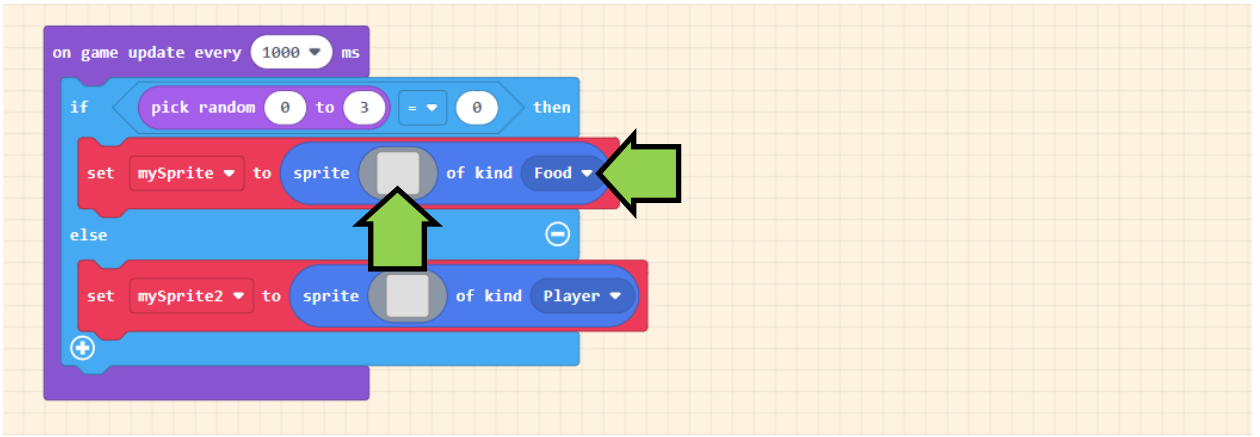
else

set mySprite2 to sprite of kind Player



5

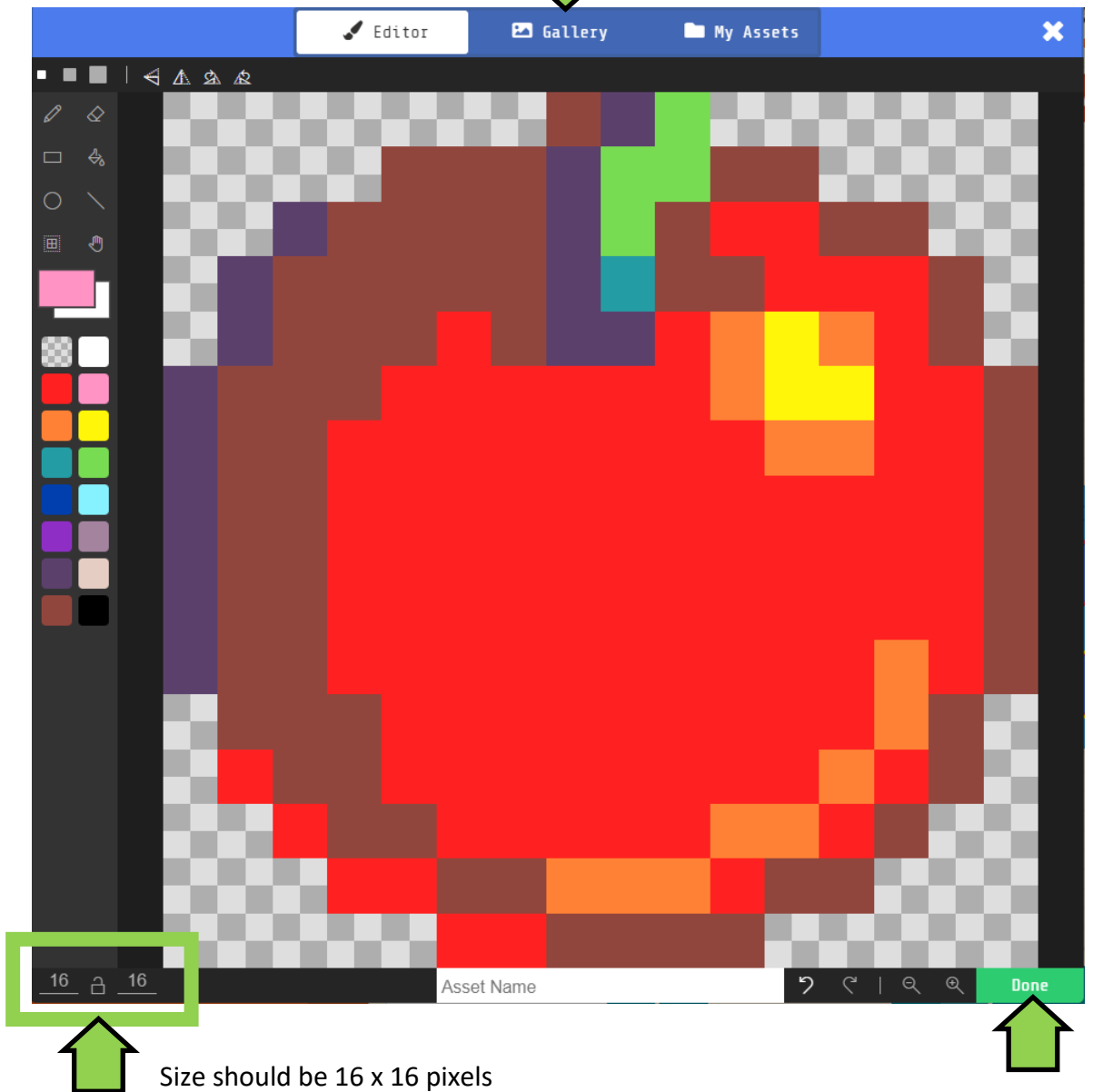
6

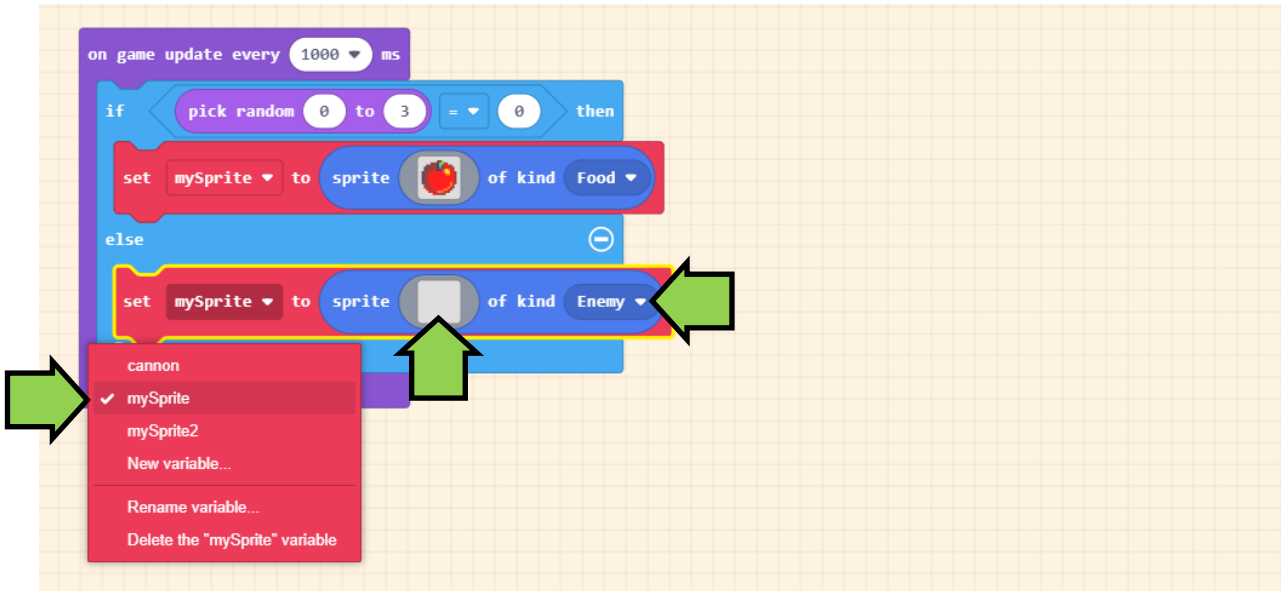


Select a fruit image from the gallery or draw your own



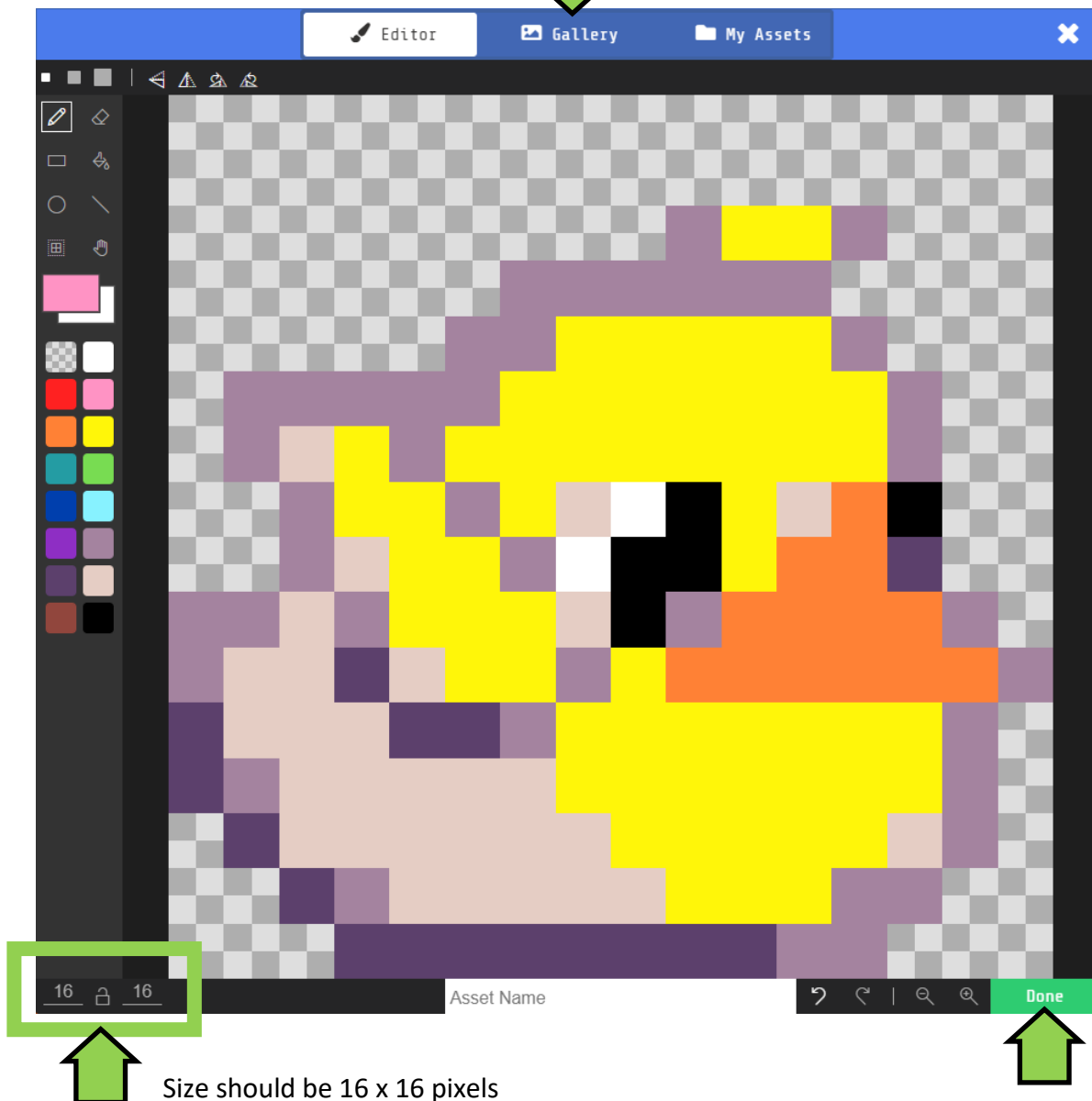
7





8

Select an image from the gallery or draw your own alien duck



9

10

```
on game update every 1000 ms
  if <pick random 0 to 3 = 0> then
    set mySprite to sprite of kind Food
  else
    set mySprite to sprite of kind Enemy
  set mySprite stay in screen ON
  set mySprite auto destroy OFF
  set mySprite position to x 0 y 0
  set mySprite velocity to vx 50 vy 50
```

11

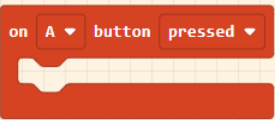
```
on game update every 1000 ms
  if <pick random 0 to 3 = 0> then
    set mySprite to sprite of kind Food
  else
    set mySprite to sprite of kind Enemy
  set mySprite stay in screen OFF
  set mySprite auto destroy ON
  set mySprite position to x 16 y 30
  set mySprite velocity to vx 50 vy 0
```

12

```
on game update every 1000 ms
  if <pick random 0 to 3 = 0> then
    set mySprite to sprite of kind Food
  else
    set mySprite to sprite of kind Enemy
  set mySprite stay in screen OFF
  set mySprite auto destroy ON
  set mySprite position to x 16 y 30
  set mySprite velocity to vx 50 vy 0
```

Step 3 - Launching shells from the cannon

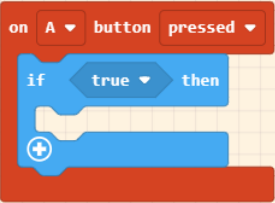
1



Controller

The Scratch script starts with a red 'on A button pressed' block. The 'Controller' menu is open on the right.

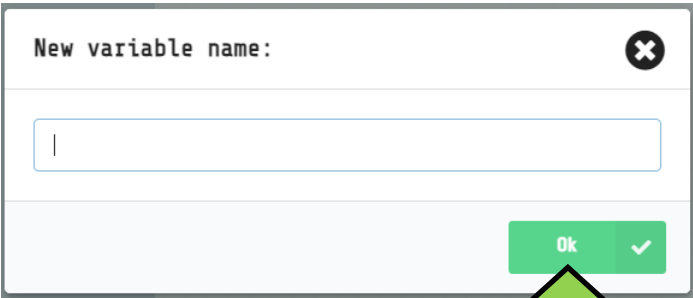
2



Logic

The Scratch script now includes an 'if true then' block nested under the 'on A button pressed' block. The 'Logic' menu is open on the right.

3

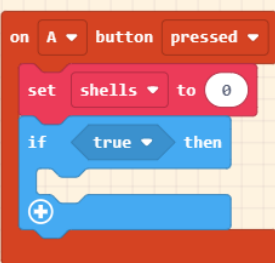


Variables

Make a Variable...

The 'New variable name' dialog box is open, showing an empty text input field and an 'Ok' button with a checkmark. A green arrow points to the 'Ok' button. The 'Variables' menu is open on the right.

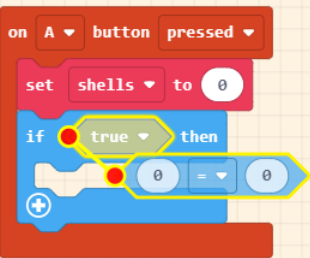
4



Variables

The Scratch script now includes a 'set shells to 0' block nested under the 'on A button pressed' block. The 'Variables' menu is open on the right.

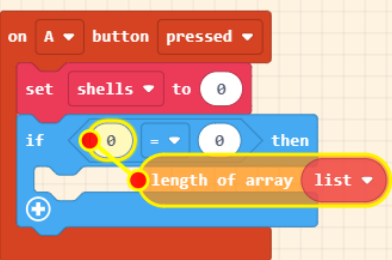
5



Logic

The Scratch script now includes a subtraction block ('0 - 0') nested under the 'if true then' block. The 'Logic' menu is open on the right.

6



Advanced

Arrays

The Scratch script now includes a 'length of array list' block nested under the 'if 0 = 0 then' block. The 'Advanced' and 'Arrays' menus are open on the right.

7

```

on A button pressed
  set shells to array of sprites of kind Player
  if length of array list = 0 then
  
```

8

```

on A button pressed
  set shells to array of sprites of kind Projectile
  if length of array shells = 0 then
  
```

9

```

on A button pressed
  set shells to array of sprites of kind Projectile
  if length of array shells = 0 then
    set projectile to projectile from mySprite with vx 50 vy 50
  
```

10

```

on A button pressed
  set shells to array of sprites of kind Projectile
  if length of array shells = 0 then
    set projectile to projectile from cannon with vx 0 vy -200
    change life by -1
  
```

11

```

on A button pressed
  set shells to array of sprites of kind Projectile
  if length of array shells = 0 then
    set projectile to projectile from cannon with vx 0 vy -200
    change life by -1
  
```

Select an image from the gallery or draw your own cannon blast

Flip vertical

16 x 16

Asset Name

Done

Size should be 16 x 16 pixels

12

```
on A button pressed
  set shells to array of sprites of kind Projectile
  if length of array shells = 0 then
    set projectile to projectile from cannon with vx 0 vy -200
    change life by -1
```

Step Done

13

Step 4 - Shells hit alien ducks

1

```
on sprite of kind Player overlaps otherSprite of kind Player
```

Sprites

2

```
on sprite of kind Projectile overlaps otherSprite of kind Enemy  
destroy mySprite  
destroy mySprite
```

Sprites

3

```
on sprite of kind Projectile overlaps otherSprite of kind Enemy  
destroy  
destroy
```

Drag mySprite away to remove

4

```
on sprite of kind Projectile overlaps otherSprite of kind Enemy  
destroy with spray effect for 500 ms  
destroy
```

Click to expand

5

```
on sprite of kind Projectile overlaps otherSprite of kind Enemy  
destroy sprite with fountain effect for 200 ms  
destroy with spray effect for 500 ms
```

Drag from sprite to destroy

6

```
on sprite of kind Projectile overlaps otherSprite of kind Enemy  
destroy sprite otherSprite effect for 200 ms  
destroy with disintegrate effect for 100 ms  
change score by 1
```

Drag from otherSprite to destroy

Info

7

```
on sprite of kind Projectile overlaps otherSprite of kind Enemy  
destroy sprite with fountain effect for 200 ms  
destroy otherSprite with disintegrate effect for 200 ms  
change score by 1
```

Step Done

Step 5 - Shells hit fruits of fortitude

1 on **sprite** of kind **Player** overlaps **otherSprite** of kind **Player**

2 on **sprite** of kind **Projectile** overlaps **otherSprite** of kind **Food**
destroy **mySprite**
destroy **mySprite**

3 on **sprite** of kind **Projectile** overlaps **otherSprite** of kind **Food**
destroy **mySprite**
destroy **mySprite**
Drag mySprite away to remove

4 on **sprite** of kind **Projectile** overlaps **otherSprite** of kind **Food**
destroy with **spray** effect for **500** ms
destroy **mySprite**
Click to expand

5 on **sprite** of kind **Projectile** overlaps **otherSprite** of kind **Food**
destroy **sprite** with **fountain** effect for **200** ms
destroy **mySprite** with **spray** effect for **500** ms
change life by **-1**
Drag from sprite to destroy

6 on **sprite** of kind **Projectile** overlaps **otherSprite** of kind **Food**
destroy **sprite** with **fountain** effect for **200** ms
destroy **otherSprite** with **disintegrate** effect for **100** ms
change life by **5**
Drag from otherSprite to destroy

7 on **sprite** of kind **Projectile** overlaps **otherSprite** of kind **Food**
destroy **sprite** with **fountain** effect for **200** ms
destroy **otherSprite** with **disintegrate** effect for **100** ms
change life by **5**
Step Done

Step 6 - Creating a function to descend sprites

1

Edit Function

Advanced

Functions

Make a Function...

Add a parameter

Text Boolean Number Array Sprite Image

function doSomething

Done

2

Edit Function

Add a parameter

Text Boolean Number Array Sprite Image

function doSomething mySprite

Click

Done

3

Edit Function

Add a parameter

Text Boolean Number Array Sprite Image

function doSomething mySprite myImage myImage2

Click x 2

Done

4

Edit Function

Add a parameter

Text Boolean Number Array Sprite Image

function dropSprite sprite left right

Done

```
function dropSprite sprite left right
```

5

```
function dropSprite sprite left right
  if true then
    if true then
      +
      +
```

Logic

6

```
function dropSprite sprite left right
  if true then
    if true then
      0 < 0
      +
      +
```

Logic

7

```
function dropSprite sprite left right
  if 0 < 0 then
    if true then
      mySprite x
      +
      +
```

Sprites

8

```
function dropSprite sprite left right
  if mySprite x < 0 then
    if true then
      or
```

Logic

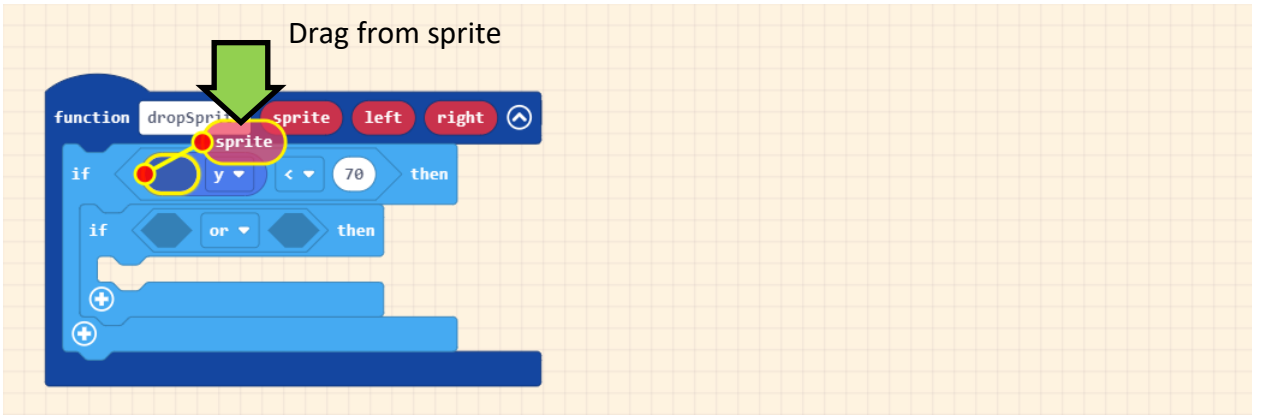
9

Drag away to remove

```
function dropSprite sprite left right
  if y < 70 then
    if
```

10

11



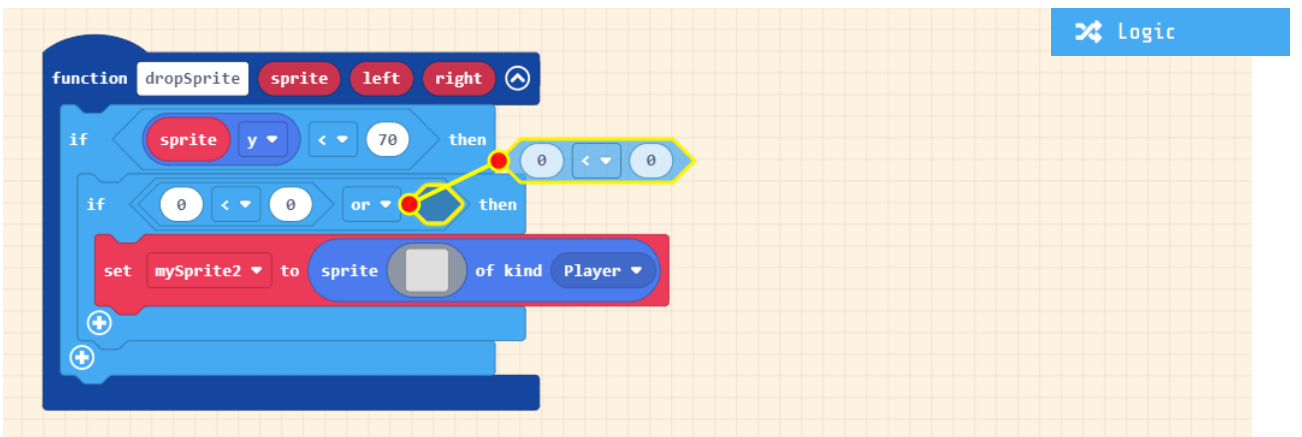
12



13



14



```

function dropSprite sprite left right
  if sprite y < 70 then
    if 0 < 0 or 0 > 160 then
      set mySprite2 to sprite of kind Player
  
```

15

Sprites

```

function dropSprite sprite left right
  if sprite y < 70 then
    if mySprite x > 160 then
      set mySprite2 to sprite of kind Player
  
```

16

Sprites

Drag away to remove

```

function dropSprite sprite left right
  if sprite y < 70 then
    if right < 0 or mySprite x > 160 then
      set mySprite2 to sprite of kind Player
  
```

17

Drag from sprite

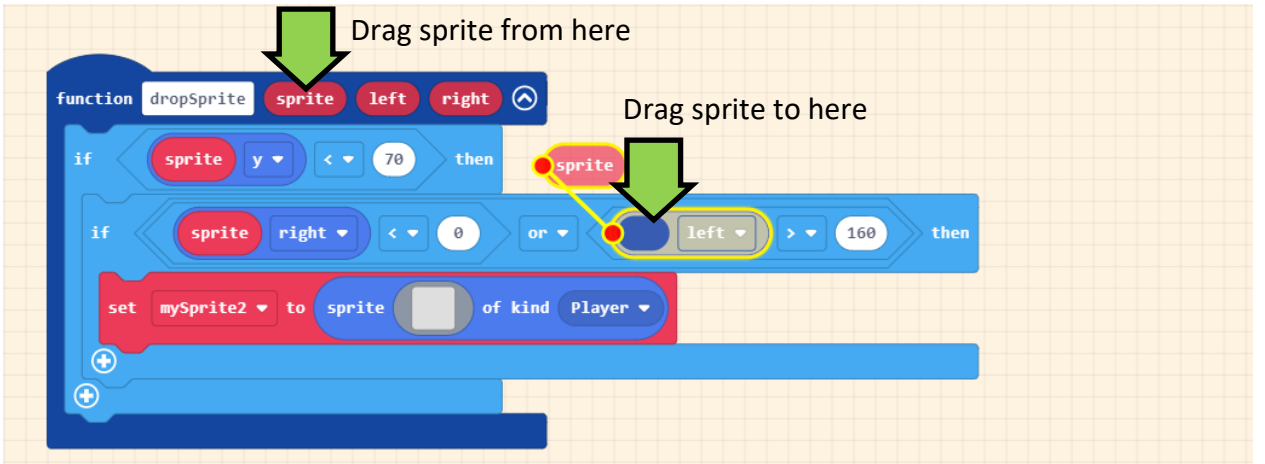
Drag away to remove

```

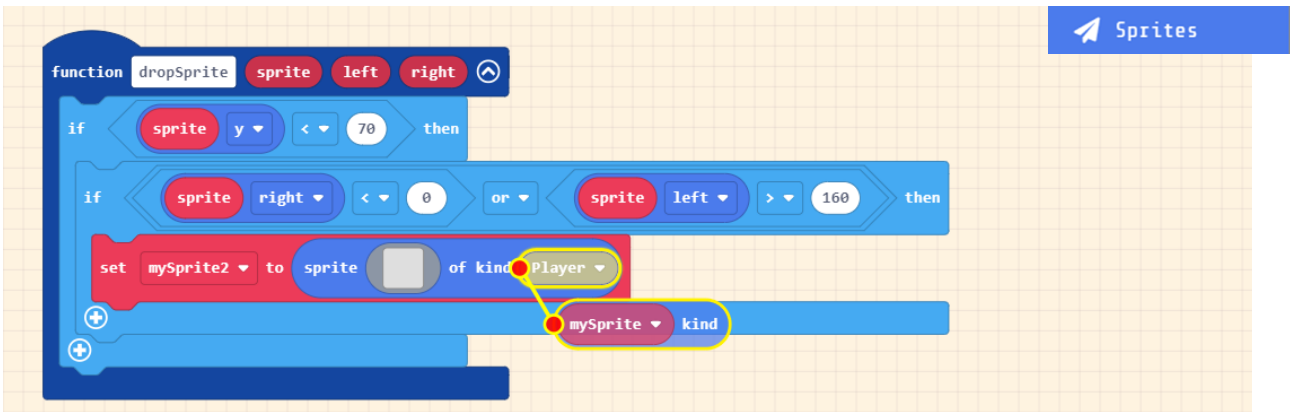
function dropSprite sprite left right
  if sprite < 70 then
    if right < 0 or left > 160 then
      set mySprite2 to sprite of kind Player
  
```

18

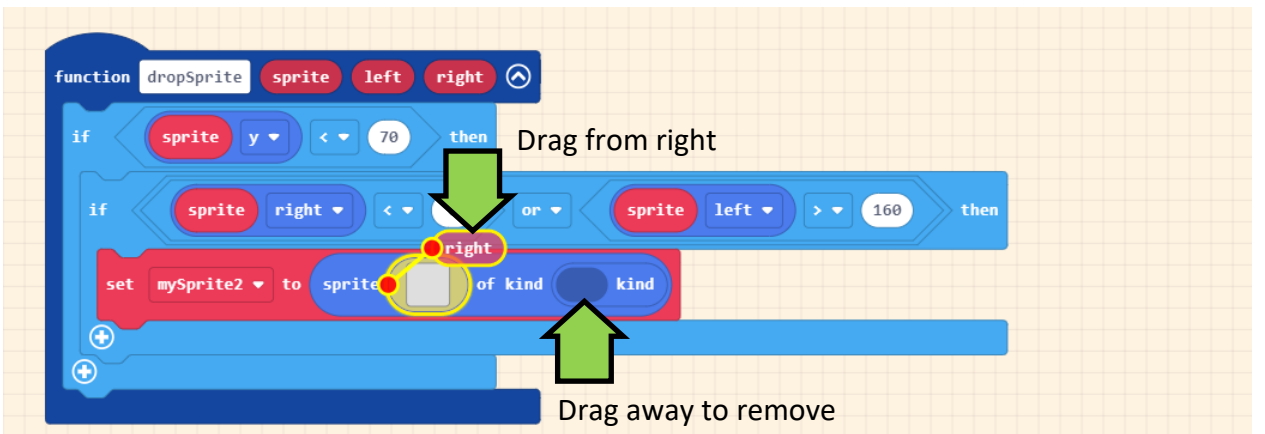
19



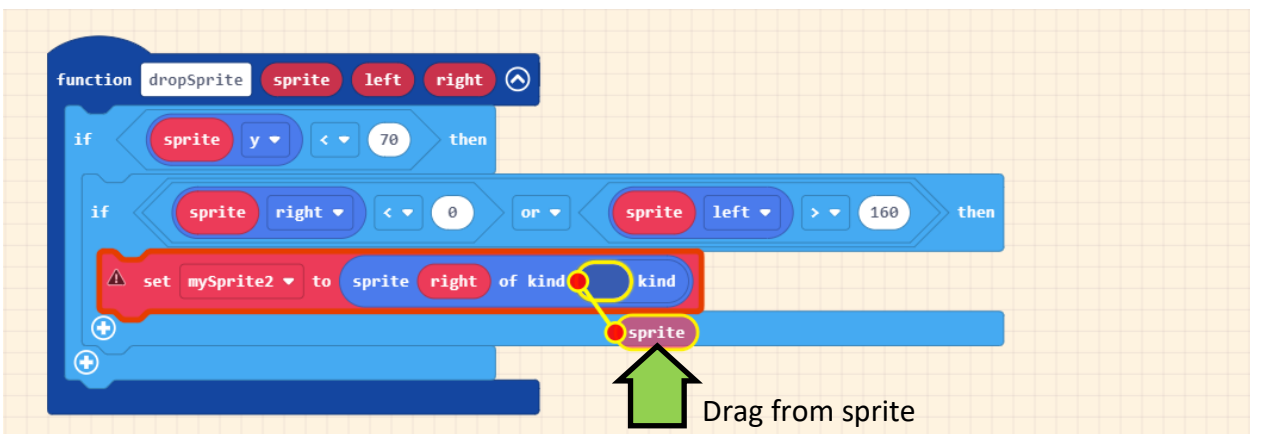
20



21



22



```
function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if true then
        +
        +
        +
```

23

```
function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if true then
        0 < 0
        +
        +
        +
```

24

```
function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if 0 < 0 then
        mySprite x
        +
        +
        +
```

25

```
function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if mySprite x > 0 then
        +
        +
        +
```

26

```

function dropSprite sprite left right
  if < sprite y < 70 > then
    if < sprite right < 0 > or < sprite left > 160 > then
      set mySprite2 to sprite right of kind sprite kind
      if < mySprite x > 0 > then
        set mySprite image to
  +
  +
  +

```

27

```

function dropSprite sprite left
  if < sprite y < 70 > then
    if < sprite right < 0 > or < sprite left > 160 > then
      set mySprite2 to sprite right of kind sprite kind
      if < mySprite vx (velocity x) > 0 > then
        set mySprite2 image to left
  +
  +
  +

```

Drag left from here

Drag left to here

28

```

function dropSprite sprite left right
  if < sprite y < 70 > then
    if < sprite right < 0 > or < sprite left > 160 > then
      set mySprite2 to sprite right of kind sprite kind
      if < mySprite vx (velocity x) > 0 > then
        set mySprite2 image to left
  +
  +
  +
  set mySprite stay in screen ON
  set mySprite auto destroy OFF
  set mySprite position to x 0 y 0
  set mySprite velocity to vx 50 vy 50

```

Drag away to remove

29

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite ← Drag sprite to here
        set mySprite2 image to left
      +
    +
  set mySprite stay in screen ON
  set mySprite auto destroy OFF
  set mySprite position to x 0 y 0
  set mySprite velocity to vx 50 vy 50
  +

```

Drag sprite from here

Drag sprite to here

30

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
    +
  set mySprite stay in screen OFF
  set mySprite auto destroy ON
  set mySprite2 position to x 0 y 0
  set mySprite2 velocity to vx 50 vy 50
  +

```

31


```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      set mySprite2 stay in screen OFF
      set mySprite2 destroy ON
      set mySprite2 position to x 0 y 0
      set mySprite2 velocity to vx 50 vy 50
  
```

32

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain 0 between 0 and 0 y 0
      set mySprite2 velocity to vx 50 vy 50
  
```

33

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
      +
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain 0 between 0 and 0 y 0
      set mySprite2 velocity to vx 0 x 0 vy 50
  +
  +

```

34

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
      +
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain mySprite x between 0 and 0 y 0
      set mySprite2 velocity to vx 0 x 0 vy 50
  +
  +

```

35

```
function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
      +
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain x between 0 and 160 y 0
      set mySprite2 velocity to vx mySprite x x 0 vy 50
```

Drag away to remove

36

```
function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
      +
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain x between 0 and 160 y 0 + 0
      set mySprite2 velocity to vx mySprite vx (velocity x) x 0 vy 50
```

Drag sprite from here

Drag sprite to here

37

```

function dropSprite sprite left right
  if < sprite y < 70 then
    if << sprite right < 0 or << sprite left > 160 >> then
      set mySprite2 to sprite right of kind sprite kind
      if < sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
    +
    set mySprite2 stay in screen OFF
    set mySprite2 auto destroy ON
    set mySprite2 position to x constrain sprite x between 0 and 160 y 0 + 25
    set mySprite2 velocity to vx vx (velocity x) x -2 vy 0
  +

```

38

Drag away to remove

```

function dropSprite sprite left right
  if < sprite y < 70 then
    if << sprite right < 0 or << sprite left > 160 >> then
      set mySprite2 to sprite right of kind sprite kind
      if < sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
    +
    set mySprite2 stay in screen OFF
    set mySprite2 auto destroy ON
    set mySprite2 position to x constrain sprite x between 0 and 160 y 0 + 25
    set mySprite2 velocity to vx vx (velocity x) x -2 vy 0
  +

```

39

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain sprite x between 0 and 160 y 0 + 25
      set mySprite2 velocity to vx sprite vx (velocity x) x -2 vy 0

```

40

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain sprite x between 0 and 160 y y + 25
      set mySprite2 velocity to vx sprite vx (velocity x) x -2 vy 0

```

41

Drag sprite from here

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
      +
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain sprite x between 0 and 160 y sprite y + 25
      set mySprite2 velocity to vx sprite vx (velocity x) x -2 vy 0
    +
  +

```

Drag sprite to here

42

```

function dropSprite sprite left right
  if sprite y < 70 then
    if sprite right < 0 or sprite left > 160 then
      set mySprite2 to sprite right of kind sprite kind
      if sprite vx (velocity x) > 0 then
        set mySprite2 image to left
      +
      +
      set mySprite2 stay in screen OFF
      set mySprite2 auto destroy ON
      set mySprite2 position to x constrain sprite x between 0 and 160 y sprite y + 25
      set mySprite2 velocity to vx sprite vx (velocity x) x -2 vy 0
    +
  +

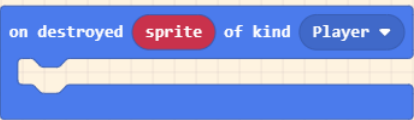
```

Step Done

43

Step 7 - Descending the fruits of fortitude


1



on destroyed **sprite** of kind **Player**

Sprites

2



on destroyed **sprite** of kind **Player**

call dropSprite **mySprite** **myImage** **myImage**

Advanced
Functions

3

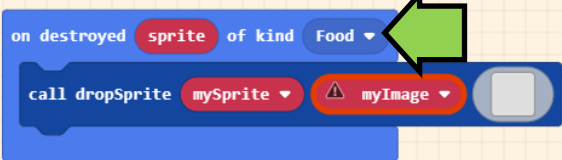


on destroyed **sprite** of kind **Player**

call dropSprite **mySprite** **myImage** **myImage**

Advanced
Images

4



on destroyed **sprite** of kind **Food**

call dropSprite **mySprite** **myImage**

5



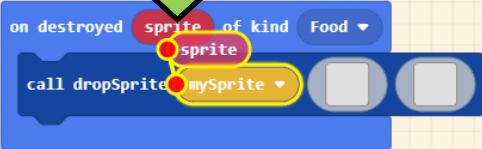
on destroyed **sprite** of kind **Food**

call dropSprite **mySprite** **myImage**

Advanced
Images

6


Drag sprite from here



on destroyed **sprite** of kind **Food**

call dropSprite **mySprite**

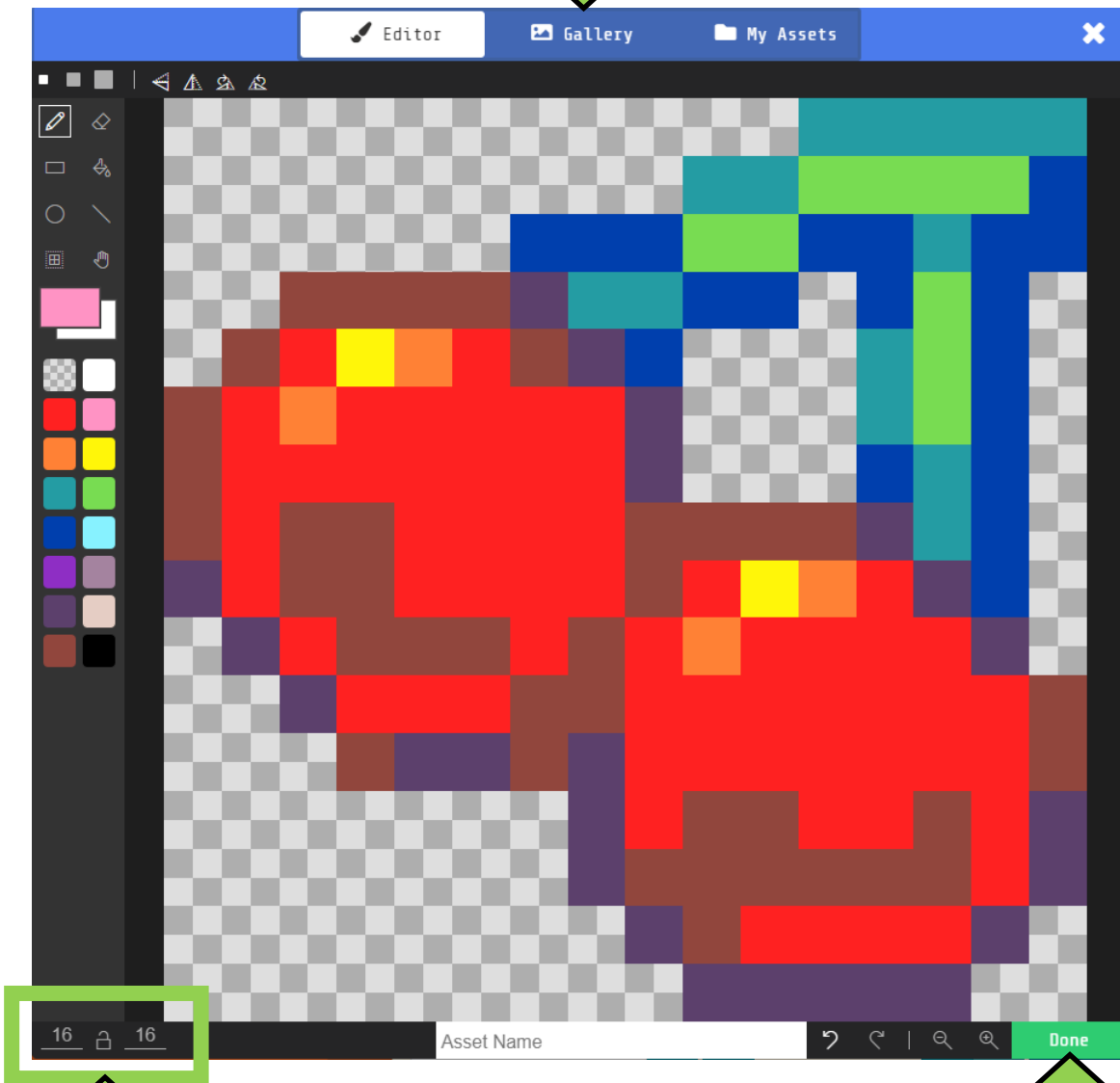
7



on destroyed **sprite** of kind **Food**

call dropSprite **sprite**

Select a fruit image from the gallery or draw your own



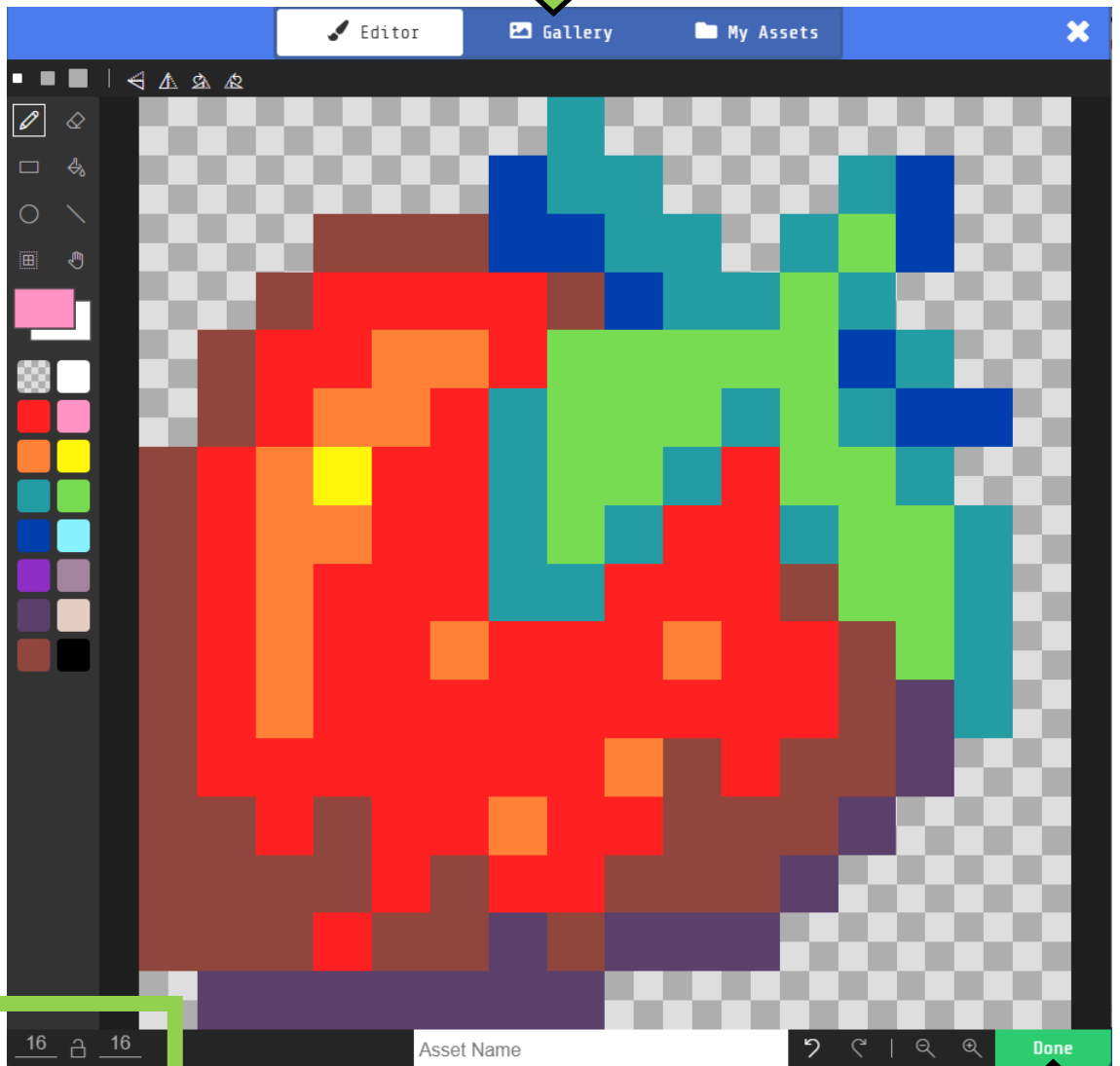
8

Size should be 16 x 16 pixels



9

Select a fruit image from the gallery or draw your own



10



Size should be 16 x 16 pixels



11

```
on destroyed sprite of kind Food
  call dropSprite sprite
```

Step Done

Step 8 - Descending the alien ducks

1

2

3

4

5

6

7

8

Drag sprite from here

Sprites

Advanced

Functions

Advanced

Images

Advanced

Images

Logic

Logic

9

```
on destroyed sprite of kind Enemy
  call dropSprite sprite
  if 0 < 0 then
    mySprite x
```

10

```
on destroyed sprite of kind Enemy
  call dropSprite sprite
  if mySprite x 0 then
```

Flip horizontal

Select an image from the gallery

11

The screenshot shows the Sprites editor interface. At the top, there are tabs for 'Editor', 'Gallery', and 'My Assets'. Below the tabs is a toolbar with various drawing tools. The main canvas displays a pixel art character with a yellow body, orange and black limbs, and purple and pink accents. The character is set against a grey and white checkerboard background. At the bottom of the editor, there is a status bar with a lock icon, the dimensions '16 x 16', an 'Asset Name' field, and a 'Done' button. Green arrows point to the 'Flip horizontal' text above the canvas, the 'Done' button, and the '16 x 16' dimensions field.

Size should be 16 x 16 pixels

```

on destroyed sprite of kind Enemy
  call dropSprite sprite
  if mySprite x > 70 then
    change score by 1
    change life by -1
  
```

12

```

on destroyed sprite of kind Enemy
  call dropSprite sprite
  if mySprite y > 70 then
    change score by -5
    change life by -1
  
```

13

```

on destroyed sprite of kind Enemy
  call dropSprite sprite
  if y > 70 then
    change score by -5
    change life by -1
  
```

Drag away to remove

14

```

on destroyed sprite of kind Enemy
  call dropSprite sprite
  if sprite y > 70 then
    change score by -5
    change life by -1
  
```

Drag sprite from here

15

```

on destroyed sprite of kind Enemy
  call dropSprite sprite
  if sprite y > 70 then
    change score by -5
    change life by -1
  
```

16

Select an image from the gallery



17

Size should be 16 x 16 pixels

18

Step Done

Appendix A - Explanations and Extensions

Step 1 - Humanities saviour (the cannon)

1. Navigate to MakeCode Arcade, click "New Project", enter the name "Ducks" and click "Create".
2. A new game will be created with the "on start" block present.
3. Add 3 blocks from the "Sprites" palette.
4. Rename "mySprite" to "cannon"; it's good practice to give variables meaningful names.
5. Click the sprite image to bring up the image editor.
6. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
7. Add a block from the "Scene" palette. Change the default starting position for the cannon sprite. This will be the centre of the bottom of the screen.
8. Add a block from the "Controller" palette. Click the "+" icon to expand the settings.
9. Change the "vy" value to zero. This stops the sprite from being moved vertically.
10. Select the cannon variable. The cannon can now be moved horizontally across the bottom of the screen.
11. Click the background image to bring up the image editor. Draw your own image and click "Done". The image should be 160 x 120 pixels, the same as the screen resolution.
12. Add 2 blocks from the "Info" palette. Change the starting lives to 20. This represents the number of shells that cannon starts with.
13. The completed code. Try it out. You will be able to move the cannon along the bottom of the screen.

Extension ideas

- Animate the cannon as it moves left or right using the blocks in the "Advanced -> Animation" palette.
- Add a musical background tune to your game using the blocks from the "Music" palette.

Step 2 - Enter the alien ducks and fruit of fortitude

This step creates sprites that come in at the top of the screen from the left-hand side. On average, 1 in 4 sprites will be a fruit and 3 in 4 sprites will be alien ducks.

1. Add the block from the "Game" palette.
2. Add the block from the "Logic" palette. Change how often the code runs.
3. Add the block from the "Logic" palette.
4. Add the block from the "Math" palette. This block will control how frequently the first condition executes.
5. Add the 2 blocks from the "Sprites" palette. Change the frequency with which the first condition executes. This will be executed 1 in 4 times on average.
6. In the first condition, change the sprite type to "Food" and click the image to bring up the image editor.
7. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
8. In the second condition, change the sprite variable to "mySprite", the sprite type to "Enemy" and click the image to bring up the image editor.
9. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
10. Add the 4 blocks from the "Sprites" palette. These control properties of the new sprite, including its starting position and speed.
11. Change the sprite properties so that the sprite starts in the upper left of the screen, only moves horizontally and automatically destroys when it leaves the screen.
12. The completed code. Try it out. You will get a single line of sprites coming from the left-hand side of the screen and disappearing off the right-hand side.

Extension ideas:

- Animate the sprites as they move right using the blocks in the "Animation" palette under "Advanced".
- Add a sound effect each time a sprite appears using the blocks from the "Music" palette.
- Adjust the frequency with which food sprites are created to 2 in 4 times.
- Adjust the rate at which new sprites are created from once per 1,000ms to once per 500ms. Try 2,000ms.
- Adjust the speed with which the sprites move. Try higher and lower numbers.

Step 3 - Launching shells from the cannon

This step adds the ability to shoot from the cannon using the A button. Each time the player shoots, a life is lost. Only a single projectile can be shot at once, thus limiting the rate of fire.

1. Add the block from the "Controller" palette.
2. Add the block from the "Logic" palette.
3. Create a new variable using "Make a Variable" from the "Variables" palette.
4. Add the block from the "Variables" palette.
5. Add the block from the "Logic" palette.
6. Add the block from the "Arrays" palette under "Advanced".
7. Add the block from the "Sprites" palette.
8. Change the sprite kind to "Projectile". The shells variable now holds an array (a list) all sprites of kind Projectile. In the condition, change "list" to "shells". The condition will now only execute when the number of sprites of kind Projectile equals zero; thus, limiting the rate of fire.
9. Add the block from the "Sprites" palette.
10. Add the block from the "Info" palette. Change the projectile to come from the "cannon" and adjust its movement to only move vertically.
11. Click the projectile image to bring up the image editor.
12. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
13. The completed code. Try it out. You will be able to shoot but it will not yet destroy any sprites.

Extension ideas:

- Animate the shells as they move upwards using the blocks in the "Animation" palette under "Advanced".
- Add a sound effect each time the shell is launched using the blocks from the "Music" palette.
- Increase the number of shells that can be on screen at once to 2 (hint you need change the condition to use < and adjust the number).

Step 4 - Shells hit alien ducks

This step adds the shells destroying the alien ducks. Points are awarded for each alien duck destroyed.

1. Add the block from the "Sprites" palette.
2. Add the 2 blocks from the "Sprites" palette. Change the kinds to "Projectile" and "Enemy".
3. Remove the "mySprite" variables from the blocks by dragging them away.
4. Click the "+" icons to expand the settings.
5. Drag "sprite" from the top line to the first "Destroy" block. Change the effect to "Fountain" and the duration to 200ms.
6. Add the block from the "Info" palette. Drag "otherSprite" from the top line to the second "Destroy" block. Change the effect to "Disintegrate" and the duration to 100ms.
7. The completed code. Try it out. You will be able to shoot the alien ducks but not the fruit.

Extension ideas:

- Add a sound effect each time an alien duck is destroyed using the blocks from the "Music" palette.
- Experiment with different effects and durations in the "destroy" blocks to find your favourites.
- Change the number of points awarded for shooting an alien duck from 1 to 5.

Step 5 - Shells hit fruits of fortitude

This step adds the shells destroying the fruits of fortitude. Additional lives are awarded for each fruit of fortitude destroyed.

1. Add the block from the "Sprites" palette.
2. Add the 2 blocks from the "Sprites" palette. Change the kinds to "Projectile" and "Food".
3. Remove the "mySprite" variables from the blocks by dragging them away.
4. Click the "+" icons to expand the settings.
5. Add the block from the "Info" palette. Drag "sprite" from the top line to the first "Destroy" block. Change the effect to "Fountain" and the duration to 200ms.
6. Drag "otherSprite" from the top line to the second "Destroy" block. Change the effect to "Disintegrate" and the duration to 100ms. Change the number of additional lives awarded to 5.
7. The completed code. Try it out. You will now be able to shoot the alien ducks and the fruits of fortitude.

Extension ideas:

- Add a sound effect each time a fruit of fortitude is destroyed using the blocks from the "Music" palette.
- Experiment with different effects and durations in the "destroy" blocks to find your favourites.
- Change the number of lives awarded for shooting a fruit of fortitude from 5 to 10.

Step 6 - Creating a function to descend sprites

This step creates a function that will be used to descend sprites from one row to another and flip the direction the sprite is travelling in as well as adjust the speed. The effect we are looking for is a sprite which goes off the screen when travelling to the right, comes back moving left and slightly lower down. The new function will take three parameters, the sprite to descend, an image to use when moving left and an image to use to moving right.

1. Select "Make a Function..." from the "Functions" palette under "Advanced".
2. Click "Sprite" to add the "mySprite" parameter.
3. Click "Image" twice to add the "myImage" and "myImage2" parameters.
4. Rename the function, rename the parameters, and click "Done".
5. The newly created function block.
6. Add the 2 blocks from the "Logic" palette.
7. Add the block from the "Logic" palette.
8. Add the block from the "Sprite" palette.
9. Add the block from the "Logic" palette.
10. Remove the "mySprite" variable from the block by dragging it away. Change the sprite property from "x" to "y". Change the value in the condition to "70".
11. Drag "sprite" from the top line to the first condition block. The condition will only pass for sprites whose vertical position is less than 70 pixels high. (0,0) is the top left corner and y values increase as they go down. Therefore, this condition only affects sprites in the upper parts of the screen.
12. Add the block from the "Sprite" palette.
13. Add the block from the "Logic" palette.
14. Add the block from the "Logic" palette.
15. Change the condition to ">" and the value to "160".
16. Add the block from the "Sprite" palette.
17. Add the block from the "Sprite" palette. Remove the "mySprite" variable from the block by dragging it away. Change the sprite property from "x" to "right".
18. Drag "sprite" from the top line to the second condition block. Remove the "mySprite" variable from the block by dragging it away. Change the sprite property from "x" to "left".
19. Drag "sprite" from the top line to the first condition block. The x co-ordinates of the screen run from 0 (leftmost) to 159 (rightmost). The condition will only pass for sprites that are offscreen.
20. Add the block from the "Sprite" palette.
21. Drag "right" from the top line to the sprite block. Remove the "mySprite" variable from the block by dragging it away.
22. Drag "sprite" from the top line to the sprite block.
23. Add the block from the "Logic" palette.
24. Add the block from the "Logic" palette.
25. Add the block from the "Sprite" palette.
26. Change the condition to ">".
27. Add the block from the "Sprite" palette.
28. Drag "left" from the top line to the sprite block. Change the sprite property from "x" to "vx (velocity x)". Change "mySprite" to "mySprite2". This block will execute when the sprite has a positive velocity and is therefore going right. Thus, we want the new sprite to be travelling left.
29. Add the 4 blocks from the "Sprite" palette. Remove the "mySprite" variable from the block by dragging it away.
30. Drag "sprite" from the top line to the third condition block.
31. Change "stay in screen" to "off" and "auto destroy" to "on". Change "mySprite" to "mySprite2" in the last 2 sprite blocks.
32. Add the block from the "Math" palette. Change "mySprite" to "mySprite2" in the other 2 sprite blocks.
33. Add the block from the "Math" palette.
34. Add the block from the "Sprite" palette.
35. Add the block from the "Sprite" palette.
36. Add the block from the "Math" palette. Remove the "mySprite" variable from the block by dragging it away. Change the second value in the constrain block to "160".
37. Drag "sprite" from the top line to the third condition block. Change the sprite property from "x" to "vx (velocity x)".
38. Remove the "mySprite" variable from the block by dragging it away. Change values of the last two blocks to "25", "-2" and "0".
39. Drag "sprite" from the top line to the last sprite block.
40. Add the block from the "Sprite" palette.
41. Remove the "mySprite" variable from the block by dragging it away. Change the sprite property from "x" to "y".

42. Drag "sprite" from the top line to the sprite block.
43. The completed code. It is not called by any other code yet, so you try it out and implement the extension ideas after you have completed the next step.

Extension ideas:

- Add a sound effect each time a sprite is created using the blocks from the "Music" palette.
- Experiment with the speeds of the lower levels. This can be done by changing the value "-2" in the last sprite block to either smaller numbers or bigger numbers. Try -3, -1.5 and -0.7.

Step 7 - Descending the fruits of fortitude

This step makes use of the dropSprite function created in the previous step to descend the fruits of fortitude.

1. Add the block from the "Sprite" palette.
2. Add the block from the "Functions" palette under "Advanced".
3. Add the block from the "Images" palette under "Advanced".
4. Change kind from "Player" to "Food".
5. Add the block from the "Images" palette under "Advanced".
6. Drag "sprite" from the top line to the dropSprite block.
7. Click the image to bring up the image editor.
8. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
9. Click the image to bring up the image editor.
10. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
11. The completed code. Try it out. The fruits of fortitude will now descend but the alien duck's wont.

Extension ideas:

- Add a sound effect each time a sprite dropped using the blocks from the "Music" palette.
- If you have animated your sprites in the earlier steps, these animations will not be present in the sprites on the lower rows. How could you add them back in?

Step 8 - Descending the alien ducks

This step makes use of the dropSprite function to descend the alien ducks. Each time a duck escapes from the bottom row, you lose points and a life.

1. Add the block from the "Sprite" palette.
2. Add the block from the "Functions" palette under "Advanced".
3. Add the block from the "Images" palette under "Advanced".
4. Change kind from "Player" to "Enemy".
5. Add the block from the "Images" palette under "Advanced".
6. Drag "sprite" from the top line to the dropSprite block.
7. Add the block from the "Logic" palette.
8. Add the block from the "Logic" palette.
9. Add the block from the "Sprite" palette.
10. Click the image to bring up the image editor.
11. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
12. Add the 2 blocks from the "Info" palette. Change the condition to ">". Change the value in the condition to 70.
13. Change the sprite property from "x" to "y". Change the score from "1" to "-5".
14. Remove the "mySprite" variable from the block by dragging it away.
15. Drag "sprite" from the top line to the sprite block. This code will only execute if the sprite has left the lowest row.
16. Click the image to bring up the image editor.
17. Select an image from the gallery or draw your own. Size should be of 16 x 16 pixels. Click "Done" when finished.
18. The completed code. Try it out. The alien ducks will now descend along with the fruits of fortitude.

Extension ideas:

- Add a sound effect each time a sprite dropped using the blocks from the "Music" palette.
- If you have animated your sprites in the earlier steps, these animations will not be present in the sprites on the lower rows. How could you add them back in?