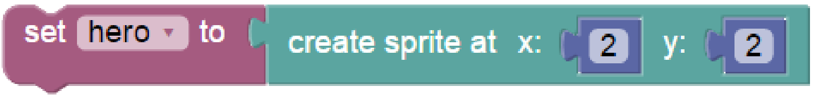
hero

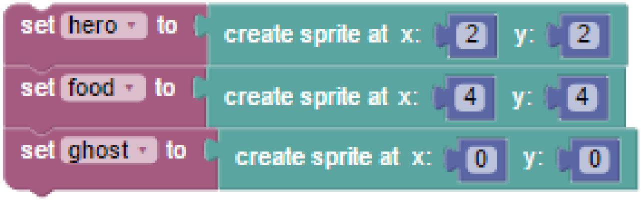
This tutorial will teach you how to create a hero game to capture food while dodging the ghost; the game was inspired by the classic arcade game Pac Man.

First we need to create a function create sprite at the x, y coordinates and that set the variable called hero on the BBC micro:bit the first time we play.



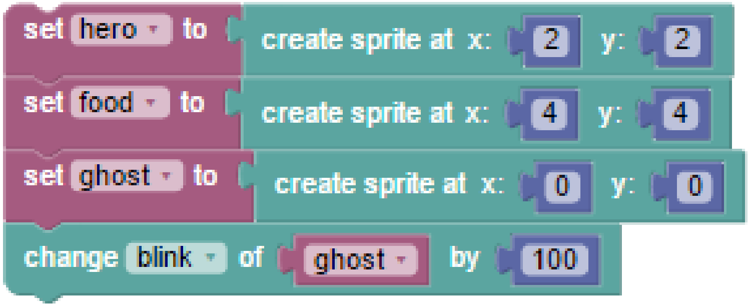
```
set hero to create sprite at x: 2 y: 2
```

Let's set up where we want our ghost will be. Create starting positions of the hero, ghost, and food in each individual quadrant. We will ensure that the ghost, food, and hero is set apart. Divide the BBC micro:bit into 4 quadrants. We don't want hero to be too close to the ghost so let's make the ghost starting positions in the 3 other quadrants. Hero is in the one quadrant. To do this, we need to keep track of the middle point in each quadrant. Create sprite at the x, y coordinates for the hero, food, and ghost.



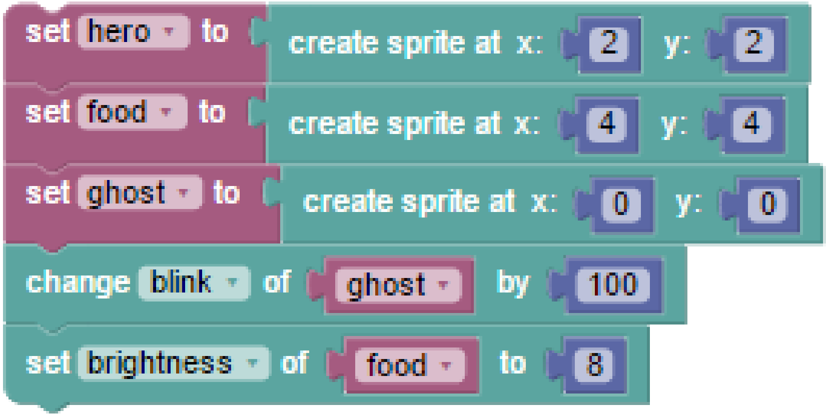
```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
```

Create a function called change blink that will plot a blinking point so the player can tell between a ghost and hero.



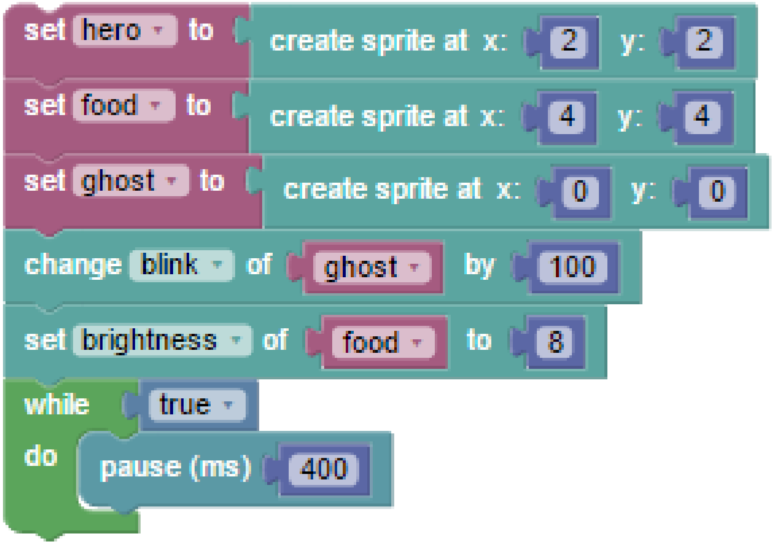
```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
change blink of ghost by 100
```

We want to identify the food so the player moves towards the food. We need to set the brightness of the variable food to 8. The brightness of the LED screen is expressed as a number between 0 and 255.



```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
change blink of ghost by 100
set brightness of food to 8
```

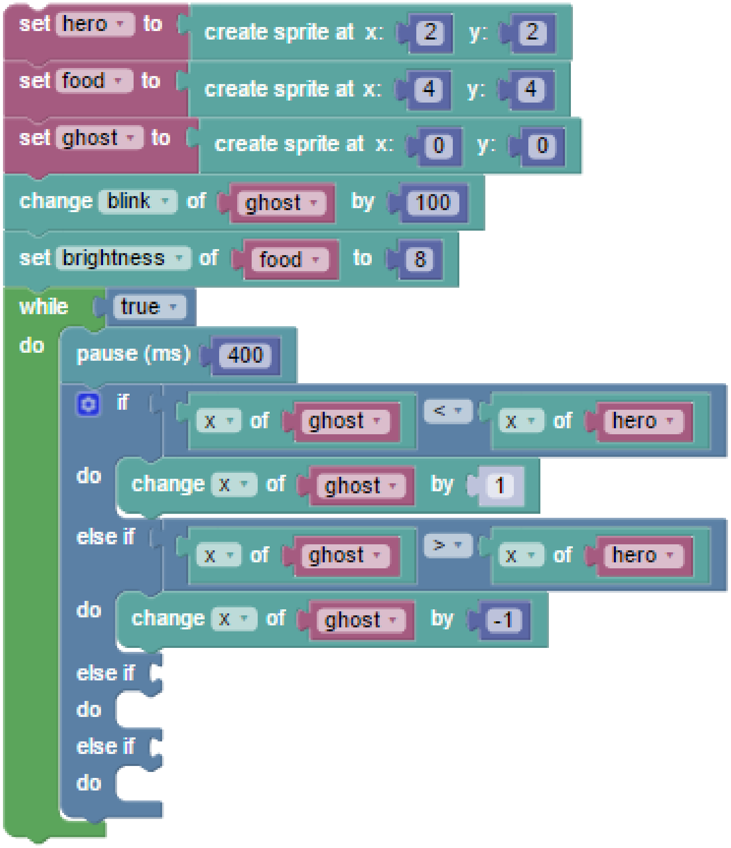
We want to include a block from the Loops drawer called While. Then set the While loop to true. This code will be important for repeating code of the game logic of the game. The game will continue to run using While loop while the Boolean condition is true. Finally, include a pause of 400 milliseconds before the logic of the game begins.



```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
change blink of ghost by 100
set brightness of food to 8
while true
do
    pause (ms) 400
```

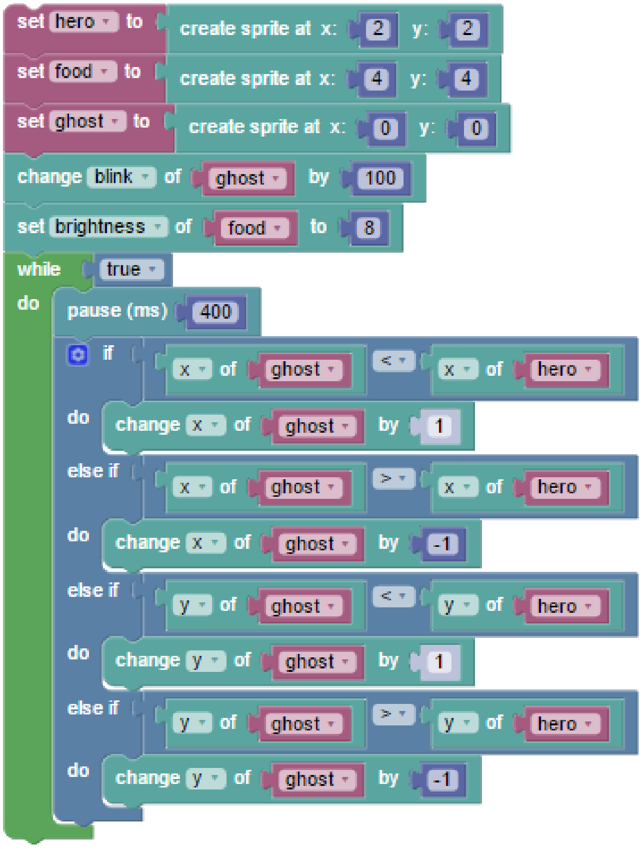
Let's create a function that will take care of keep the ghost pursuing the hero. We will need to a conditional statement that checks the position of the ghost and hero. The first condition will check if the horizontal coordinates of the ghost is less than the horizontal coordinates of the hero. We create a function from the Game drawer that will check the coordinates of the hero and the ghost. Finally, change the x-direction of the ghost by 1.

Then create another function that will take care of keep the ghost pursuing the hero. We will need to a conditional statement that checks the position of the ghost and hero. The second condition will check if the horizontal coordinates of the ghost is greater than the x-direction of hero. We create a function from the Game drawer that will check the x-direction of hero and ghost. Finally, change the x-direction of the ghost by -1.



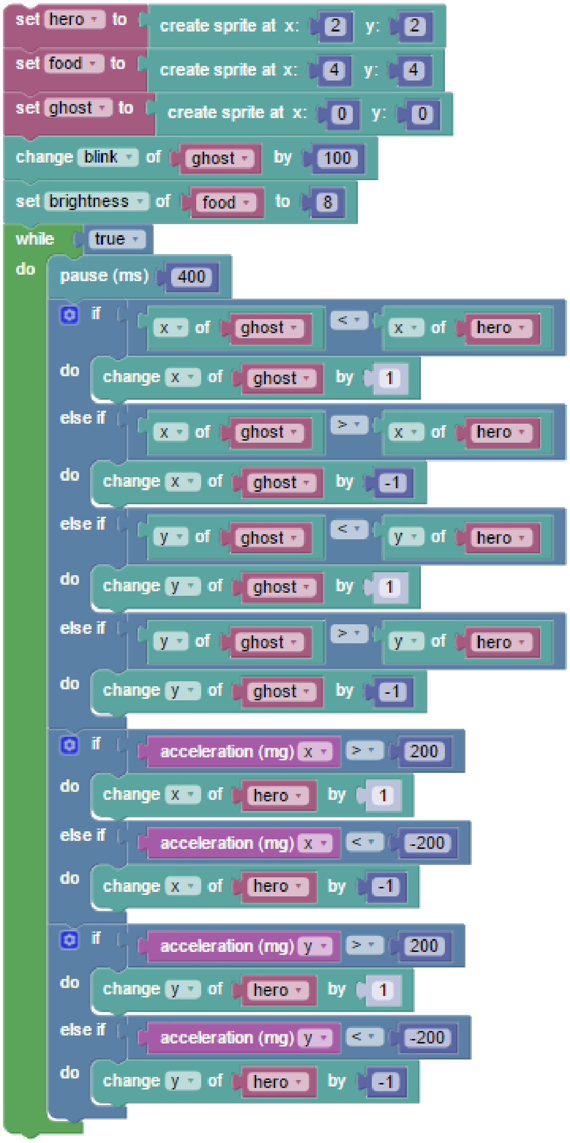
```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
change blink of ghost by 100
set brightness of food to 8
while true
do
    pause (ms) 400
    if x of ghost < x of hero
    do
        change x of ghost by 1
    else if x of ghost > x of hero
    do
        change x of ghost by -1
    else if
    do
    else if
    do
```

Let's create the third function and forth function that continues the same logic in the y-direction of pacman and ghost. We create a function from the Game drawer that will check the y-direction of pacman and ghost. Finally, change the y-direction of the ghost to continue following pacman.



```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
change blink of ghost by 100
set brightness of food to 8
while true
do
    pause (ms) 400
    if x of ghost < x of hero
    do
        change x of ghost by 1
    else if x of ghost > x of hero
    do
        change x of ghost by -1
    else if y of ghost < y of hero
    do
        change y of ghost by 1
    else if y of ghost > y of hero
    do
        change y of ghost by -1
```

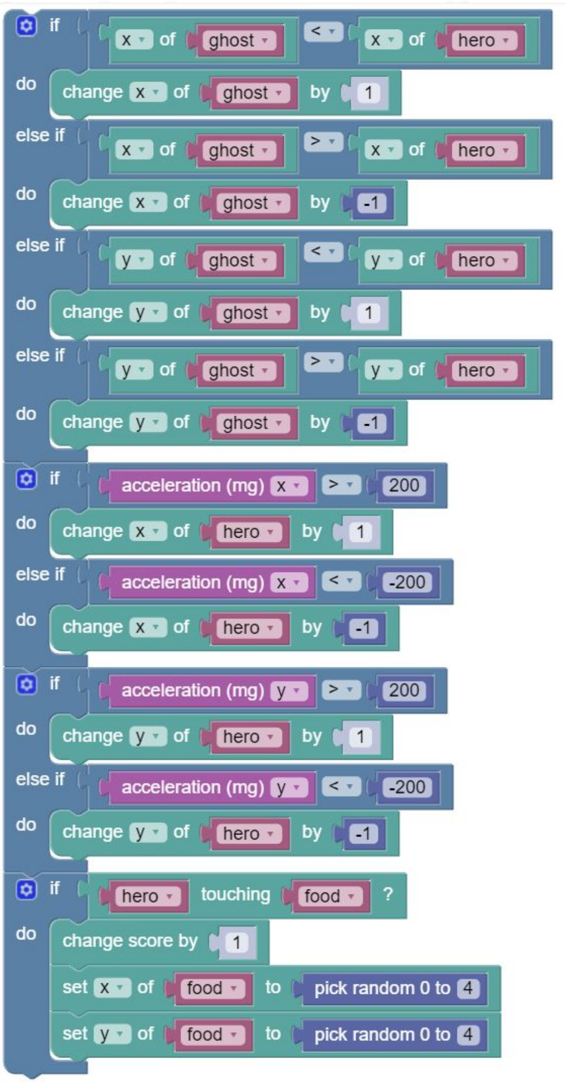
Let's enable pacman to move in the x-direction and move in the y-direction with acceleration using the BBC micor:bit sensor



```
set hero to create sprite at x: 2 y: 2
set food to create sprite at x: 4 y: 4
set ghost to create sprite at x: 0 y: 0
change blink of ghost by 100
set brightness of food to 8
while true
do
    pause (ms) 400
    if x of ghost < x of hero
    do
        change x of ghost by 1
    else if x of ghost > x of hero
    do
        change x of ghost by -1
    else if y of ghost < y of hero
    do
        change y of ghost by 1
    else if y of ghost > y of hero
    do
        change y of ghost by -1
    if acceleration (mg) x > 200
    do
        change x of hero by 1
    else if acceleration (mg) x < -200
    do
        change x of hero by -1
    if acceleration (mg) y > 200
    do
        change y of hero by 1
    else if acceleration (mg) y < -200
    do
        change y of hero by -1
```

Do not disconnect the blocks for the conditional statements. We are focusing on this section of the code and are not showing the entire code

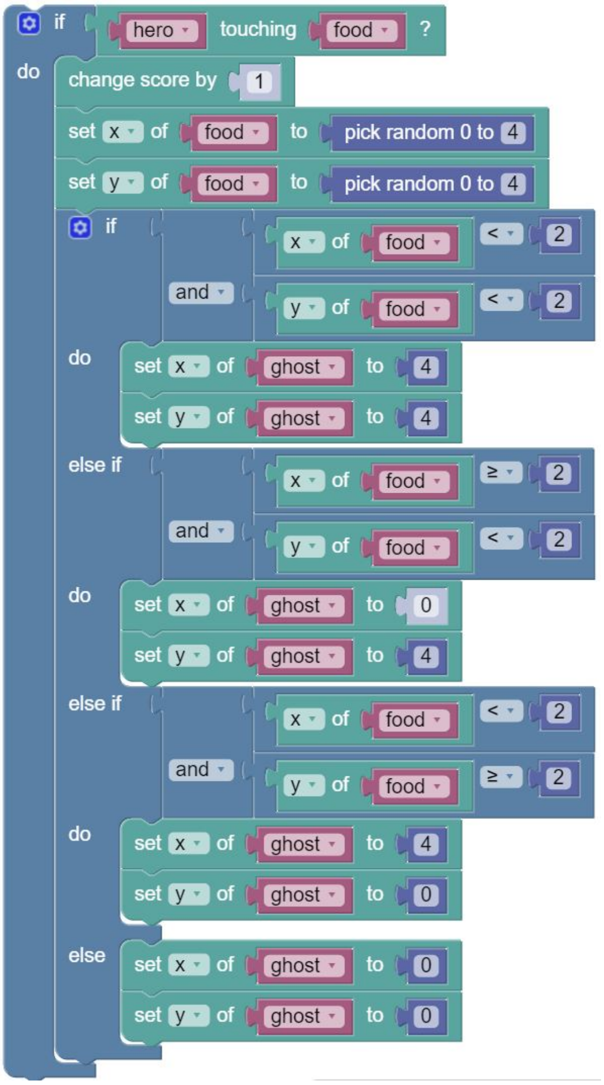
Let's setup the logic for the food. If hero is touching "food", increase the score of the game by 1 and set x -direction of food randomly randomly from 0 to 4 and set y -direction of food randomly from 0 to 4.



```
if x of ghost < x of hero
do  change x of ghost by 1
else if x of ghost > x of hero
do  change x of ghost by -1
else if y of ghost < y of hero
do  change y of ghost by 1
else if y of ghost > y of hero
do  change y of ghost by -1
if acceleration (mg) x > 200
do  change x of hero by 1
else if acceleration (mg) x < -200
do  change x of hero by -1
if acceleration (mg) y > 200
do  change y of hero by 1
else if acceleration (mg) y < -200
do  change y of hero by -1
if hero touching food ?
do  change score by 1
    set x of food to pick random 0 to 4
    set y of food to pick random 0 to 4
```

Do not disconnect the blocks from the conditional statements. We are focusing on this section of the code and are not showing the entire code

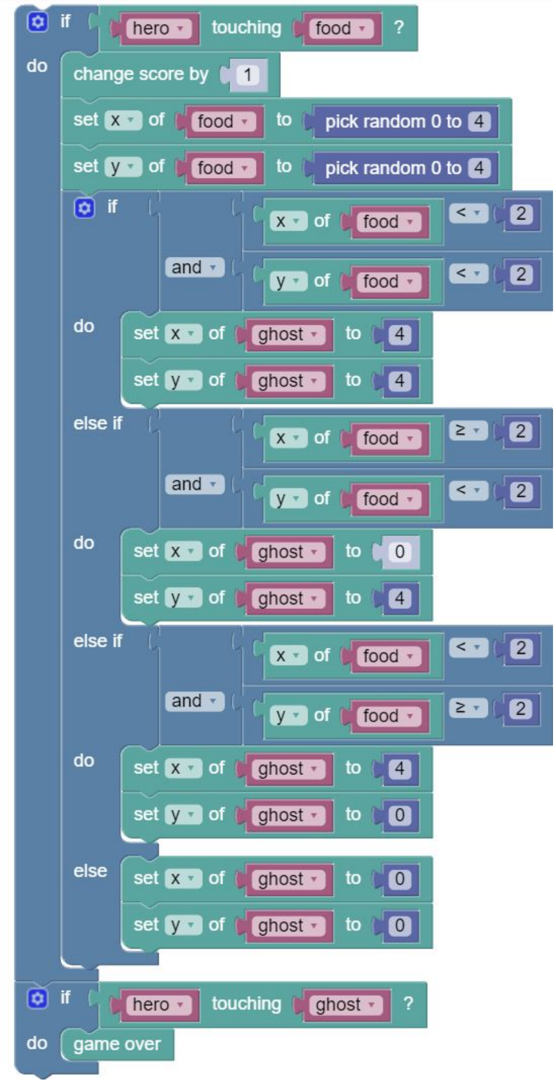
Let's setup the logic for the food and the ghost to be in different quadrants.



```
if hero touching food ?
do
  change score by 1
  set x of food to pick random 0 to 4
  set y of food to pick random 0 to 4
  if x of food < 2
     and y of food < 2
  do
    set x of ghost to 4
    set y of ghost to 4
  else if x of food ≥ 2
     and y of food < 2
  do
    set x of ghost to 0
    set y of ghost to 4
  else if x of food < 2
     and y of food ≥ 2
  do
    set x of ghost to 4
    set y of ghost to 0
  else
    set x of ghost to 0
    set y of ghost to 0
```

Do not disconnect the blocks for the conditional statements. We are focusing on this section of the code and are not showing the entire code

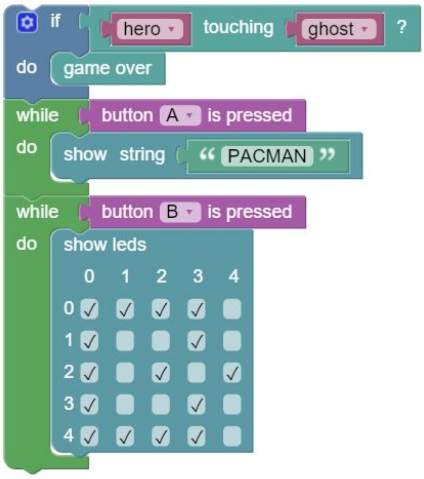
The game over component is now upon the game. If the hero is `touching` the ghost, let's display `game over`



```
if hero touching food ?
do  change score by 1
    set x of food to pick random 0 to 4
    set y of food to pick random 0 to 4
    if  x of food < 2
        and
        y of food < 2
    do  set x of ghost to 4
        set y of ghost to 4
    else if  x of food ≥ 2
             and
             y of food < 2
    do  set x of ghost to 0
        set y of ghost to 4
    else if  x of food < 2
             and
             y of food ≥ 2
    do  set x of ghost to 4
        set y of ghost to 0
    else
        set x of ghost to 0
        set y of ghost to 0
if hero touching ghost ?
do  game over
```

Do not disconnect the conditional statements blocks from the remaining blocks. We are focusing on this section of the code and are not showing the entire code

The game can also use the buttons. Let's create a `while` loop for button A is pressed. If button A is pressed, the micro:bit will display the word Hero. Let's use a `while` loop for button B is pressed. If button B is pressed, the micro:bit will display image of the hero.



```
if hero touching ghost ?
do  game over
while button A is pressed
do  show string " PACMAN "
while button B is pressed
do  show leds
       0 1 2 3 4
    0  ✓ ✓ ✓ ✓ □
    1  ✓ □ □ ✓ □
    2  ✓ □ ✓ □ ✓
    3  ✓ □ □ ✓ □
    4  ✓ ✓ ✓ ✓ □
```

You can review the final code for **hero (https://www.microbit.co.uk/numraj)**