Keyboard input

Code | Animation

Count: 3

Reset

Show grid

Show Toolbox | Workspace: | Version History | Show Text

```
1  var count=0;
2  textSize(50);
3  function draw() {
4    background("white");
5    if (mouseWentDown("rightButton")){
6      count++;
7    } else if (mouseWentDown("leftButton")) {
8      count--;
9    }
10   text("Count: "+count, 100, 200);
11 }
12
```

Show Debug Commands | Debug Sprites: Off | Clear | Watch

1. Initialize loop stopping variables

The code above the draw loop is run
once before the loop runs.

1. Initialize loop stopping variables

The code above the draw loop is run
once before the loop runs.

2. Test loop stopping condition

The draw loop code begins and will only
stop when the user presses reset.

1. Initialize loop stopping variables

The code above the draw loop is run
once before the loop runs.

2. Test loop stopping condition

The draw loop code begins and will only
stop when the user presses reset.

3. Steps to repeat

The code that will be repeated in
each iteration of the draw loop is
put inside the function draw() { }

1. Initialize loop stopping variables

The code above the draw loop is run
once before the loop runs.

2. Test loop stopping condition

The draw loop code begins and will only
stop when the user presses reset.

3. Steps to repeat

The code that will be repeated in
each iteration of the draw loop is
put inside the function draw() { }

4. Progress to the loop stopping condition.

After a certain length of time, the draw loop runs again.

1. Initialize loop stopping variables

The code above the draw loop is run once before the loop runs.

```
var count=0;
textSize(50);
```

2. Test loop stopping condition

The draw loop code begins and will only stop when the user presses reset.

3. Steps to repeat

The code that will be repeated in each iteration of the draw loop is put inside the function draw() { }

4. Progress to the loop stopping condition.

After a certain length of time, the draw loop runs again.

1. Initialize loop stopping variables

The code above the draw loop is run once before the loop runs.

```
var count=0;
textSize(50);
```

2. Test loop stopping condition

The draw loop code begins and will only stop when the user presses reset.

Was 

Pressed?

3. Steps to repeat

The code that will be repeated in each iteration of the draw loop is put inside the function draw() { }

4. Progress to the loop stopping condition.

After a certain length of time, the draw loop runs again.

1. Initialize loop stopping variables

The code above the draw loop is run once before the loop runs.

```
var count=0;
textSize(50);
```

2. Test loop stopping condition

The draw loop code begins and will only stop when the user presses reset.

3. Steps to repeat

The code that will be repeated in each iteration of the draw loop is put inside the function draw() { }

```
function draw() {
  background("white");
  if (mouseWentDown("rightButton")){
    count++;
  } else if (mouseWentDown("leftButton")) {
    count--;
  }
  text("Count: "+count, 100, 200);
}
```

4. Progress to the loop stopping condition.

After a certain length of time, the draw loop runs again.

1. Initialize loop stopping variables

The code above the draw loop is run once before the loop runs.

```
var count=0;
textSize(50);
```

2. Test loop stopping condition

The draw loop code begins and will only stop when the user presses reset.

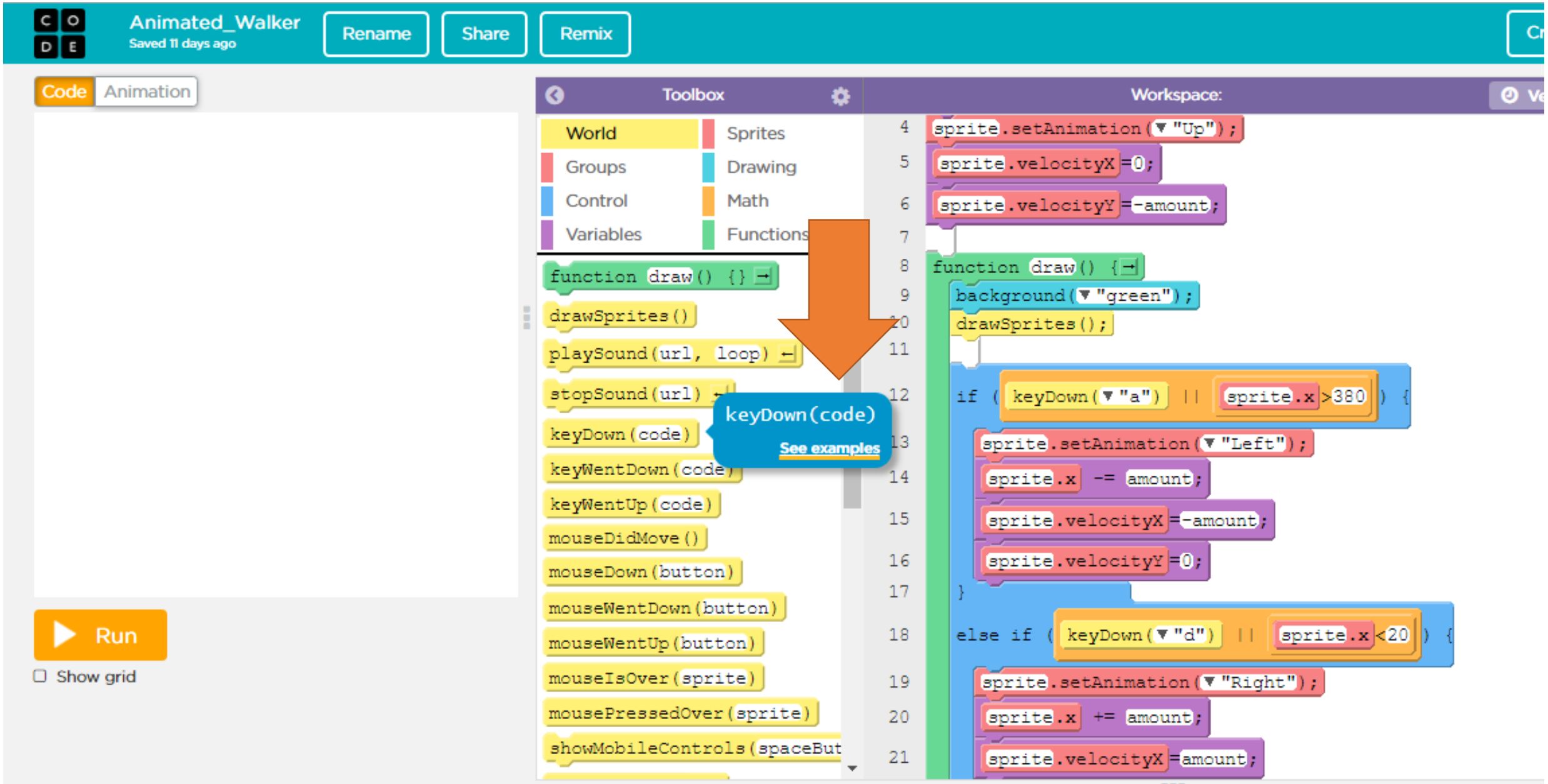
3. Steps to repeat

The code that will be repeated in each iteration of the draw loop is put inside the function draw() { }

```
function draw() {
  background("white");
  if (mouseWentDown("rightButton")){
    count++;
  } else if (mouseWentDown("leftButton")) {
    count--;
  }
  text("Count: "+count, 100, 200);
}
```

4. Progress to the loop stopping condition.

After a certain length of time, the draw loop runs again.

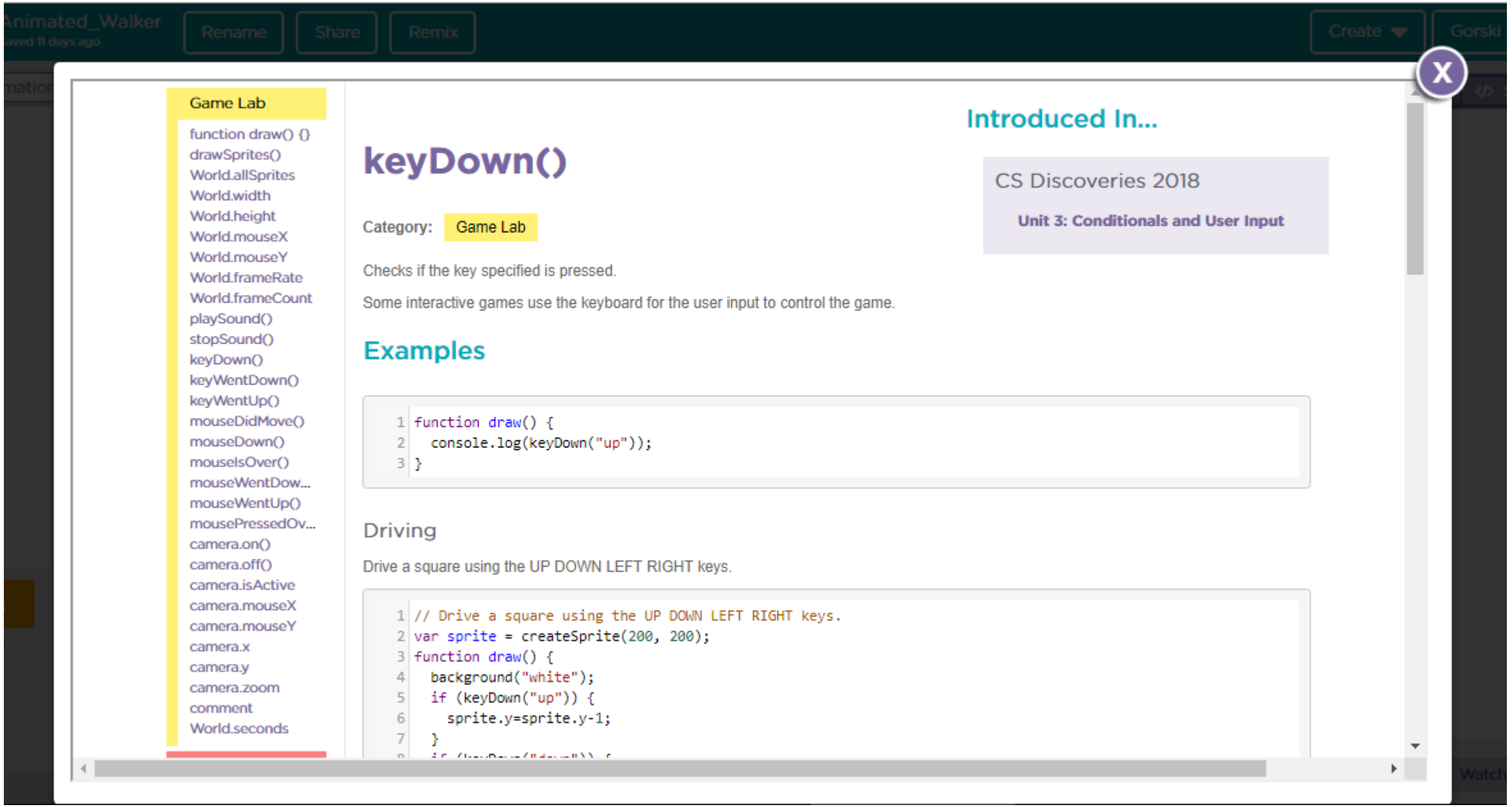

Animated_Walker
Saved 11 days ago
Rename
Share
Remix
Code
Animation
Run
Show grid
Toolbox
World
Sprites
Groups
Drawing
Control
Math
Variables
Functions
function draw() {}
drawSprites()
playSound(url, loop)
stopSound(url)
keyDown(code)
keyWentDown(code)
keyWentUp(code)
mouseDidMove()
mouseDown(button)
mouseWentDown(button)
mouseWentUp(button)
mouseIsOver(sprite)
mousePressedOver(sprite)
showMobileControls(spaceBut
keyDown(code)
See examples
Workspace:
sprite.setAnimation("Up");
sprite.velocityX=0;
sprite.velocityY=-amount;
function draw() {
background("green");
drawSprites();
if (keyDown("a") || sprite.x>380) {
sprite.setAnimation("Left");
sprite.x -= amount;
sprite.velocityX=-amount;
sprite.velocityY=0;
}
else if (keyDown("d") || sprite.x<20) {
sprite.setAnimation("Right");
sprite.x += amount;
sprite.velocityX=amount;

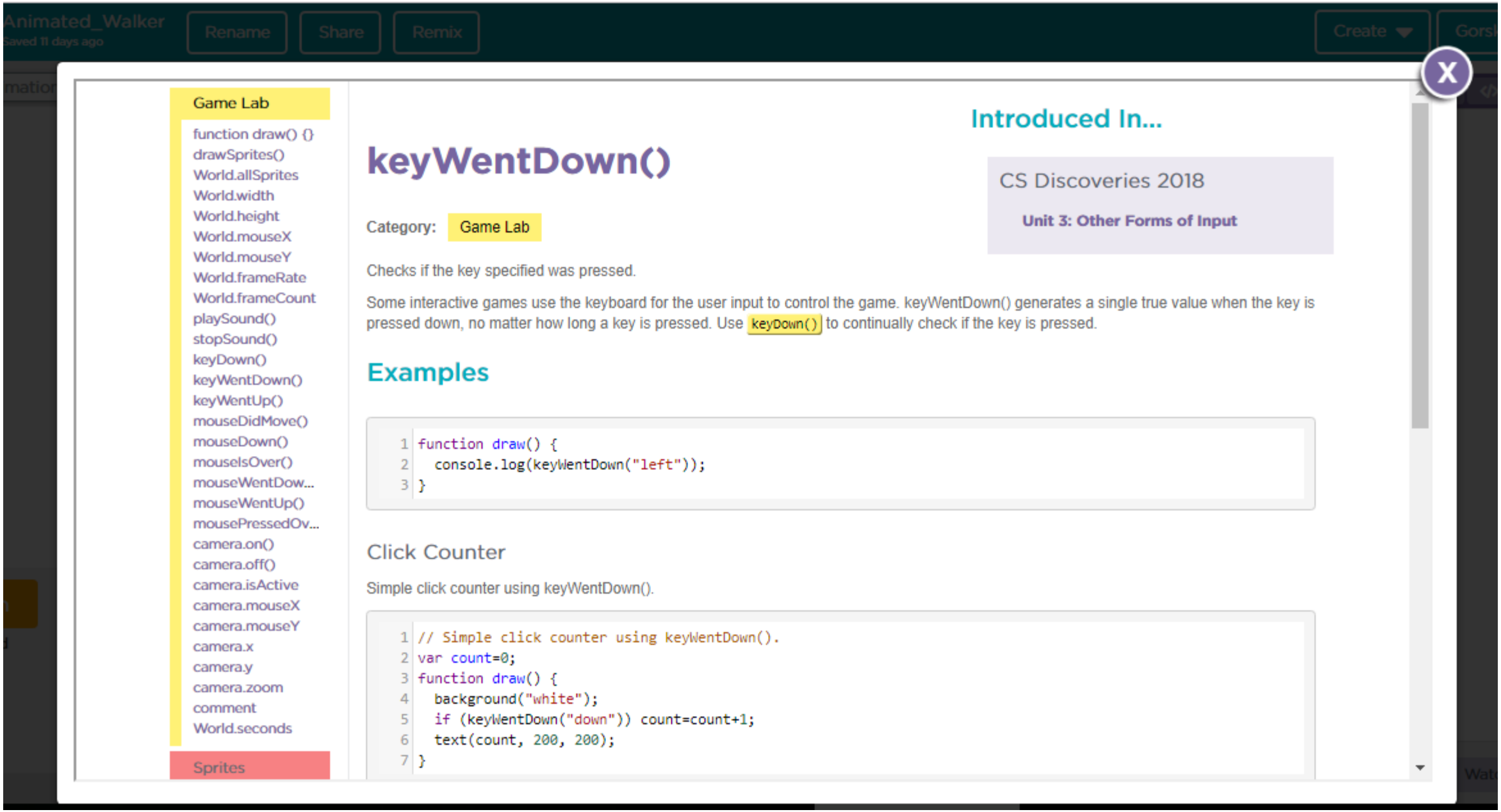
```
mouseDown(button)
```

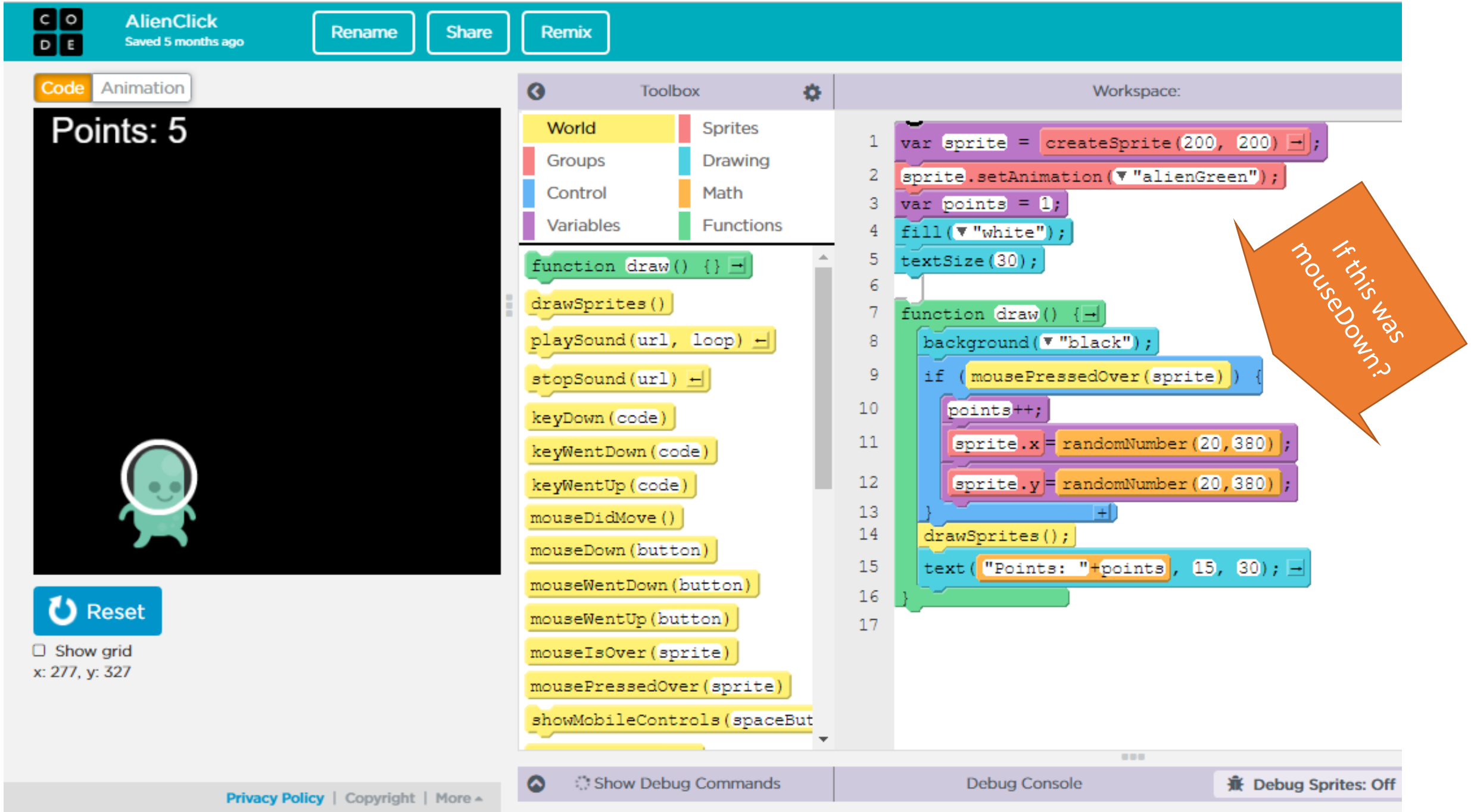
All the time the mouse is down.
Might run several times.
Don’t use if you’ve got a score.

```
mouseWentDown(button)
```

Once per click.
Only runs once.
Great if you’ve got a score.


Animated_Walker
Rename
Share
Remix
Create
Gorski
Game Lab
function draw() {}
drawSprites()
World.allSprites
World.width
World.height
World.mouseX
World.mouseY
World.frameRate
World.frameCount
playSound()
stopSound()
keyDown()
keyWentDown()
keyWentUp()
mouseDidMove()
mouseDown()
mouseIsOver()
mouseWentDow...
mouseWentUp()
mousePressedOv...
camera.on()
camera.off()
camera.isActive
camera.mouseX
camera.mouseY
camera.x
camera.y
camera.zoom
comment
World.seconds
keyDown()
Category: Game Lab
Checks if the key specified is pressed.
Some interactive games use the keyboard for the user input to control the game.
Examples
function draw() {
console.log(keyDown("up"));
}
Driving
Drive a square using the UP DOWN LEFT RIGHT keys.
// Drive a square using the UP DOWN LEFT RIGHT keys.
var sprite = createSprite(200, 200);
function draw() {
background("white");
if (keyDown("up")) {
sprite.y=sprite.y-1;
}
Introduced In...
CS Discoveries 2018
Unit 3: Conditionals and User Input
X


Animated_Walker
Saved 11 days ago
Rename
Share
Remix
Create
Game Lab
function draw() {}
drawSprites()
World.allSprites
World.width
World.height
World.mouseX
World.mouseY
World.frameRate
World.frameCount
playSound()
stopSound()
keyDown()
keyWentDown()
keyWentUp()
mouseDidMove()
mouseDown()
mouseIsOver()
mouseWentDow...
mouseWentUp()
mousePressedOv...
camera.on()
camera.off()
camera.isActive
camera.mouseX
camera.mouseY
camera.x
camera.y
camera.zoom
comment
World.seconds
Sprites
keyWentDown()
Introduced In...
CS Discoveries 2018
Unit 3: Other Forms of Input
Category: Game Lab
Checks if the key specified was pressed.
Some interactive games use the keyboard for the user input to control the game. keyWentDown() generates a single true value when the key is pressed down, no matter how long a key is pressed. Use keyDown() to continually check if the key is pressed.
Examples
function draw() {
console.log(keyWentDown("left"));
}
Click Counter
Simple click counter using keyWentDown().
// Simple click counter using keyWentDown().
var count=0;
function draw() {
background("white");
if (keyWentDown("down")) count=count+1;
text(count, 200, 200);
}


AlienClick
Saved 5 months ago
Rename
Share
Remix
Code
Animation
Points: 5
Reset
Show grid
x: 277, y: 327
Privacy Policy | Copyright | More
Toolbox
World
Sprites
Groups
Drawing
Control
Math
Variables
Functions
function draw() {}
drawSprites()
playSound(url, loop)
stopSound(url)
keyDown(code)
keyWentDown(code)
keyWentUp(code)
mouseDidMove()
mouseDown(button)
mouseWentDown(button)
mouseWentUp(button)
mouseIsOver(sprite)
mousePressedOver(sprite)
showMobileControls(spaceBut
Workspace:
1 var sprite = createSprite(200, 200);
2 sprite.setAnimation("alienGreen");
3 var points = 1;
4 fill("white");
5 textSize(30);
6
7 function draw() {
8 background("black");
9 if (mousePressedOver(sprite)) {
10 points++;
11 sprite.x = randomNumber(20,380);
12 sprite.y = randomNumber(20,380);
13 }
14 drawSprites();
15 text("Points: "+points, 15, 30);
16 }
17
If this was mouseDown?
Show Debug Commands
Debug Console
Debug Sprites: Off

1. Initialize loop stopping variables

```
var sprite = createSprite(200, 200);
sprite.setAnimation("alienGreen");
var points = 1;
fill("white");
textSize(30);
```

Run

2. Test loop stopping condition

Was Reset Pressed?

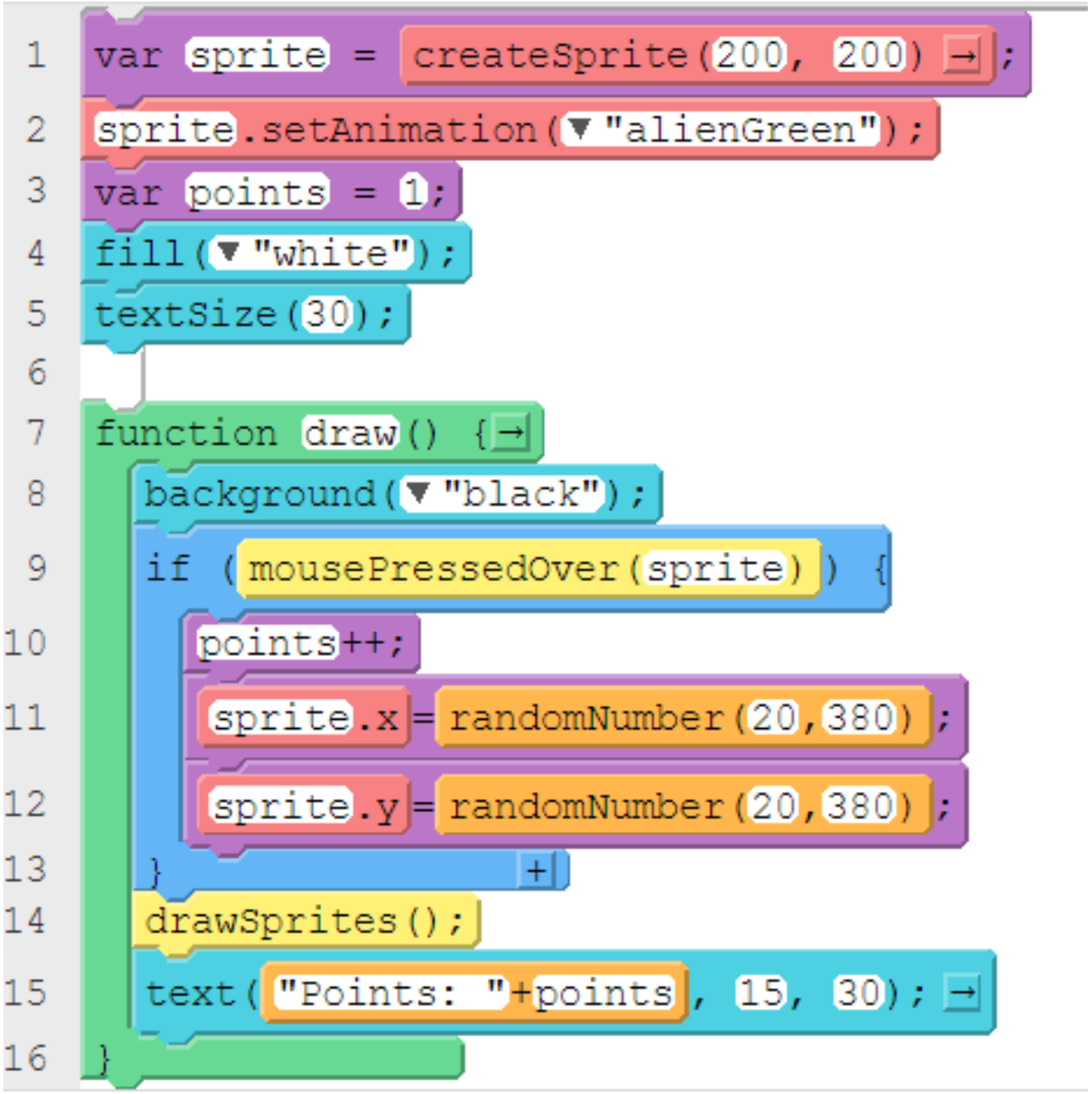
3. Steps to repeat

```
function draw() {
  background("black");
  if (mousePressedOver(sprite)) {
    points++;
    sprite.x = randomNumber(20,380);
    sprite.y = randomNumber(20,380);
  }
  drawSprites();
  text("Points: "+points, 15, 30);
}
```

4. Progress to the loop stopping condition.

The Counter Pattern

Can be used to keep score.



```
1  var sprite = createSprite(200, 200);
2  sprite.setAnimation("alienGreen");
3  var points = 1;
4  fill("white");
5  textSize(30);
6
7  function draw() {
8    background("black");
9    if (mousePressedOver(sprite)) {
10     points++;
11     sprite.x = randomNumber(20,380);
12     sprite.y = randomNumber(20,380);
13   }
14   drawSprites();
15   text("Points: "+points, 15, 30);
16 }
```

The Counter Pattern

Can be used to keep score.

```
1  var sprite = createSprite(200, 200);
2  sprite.setAnimation("alienGreen");
3  var points = 1;
4  fill("white");
5  textSize(30);
6
7  function draw() {
8    background("black");
9    if (mousePressedOver(sprite)) {
10     points++;
11     sprite.x = randomNumber(20,380);
12     sprite.y = randomNumber(20,380);
13   }
14   drawSprites();
15   text("Points: "+points, 15, 30);
16 }
```

Start the variable at 1

The Counter Pattern

Can be used to keep score.

```
1  var sprite = createSprite(200, 200);
2  sprite.setAnimation("alienGreen");
3  var points = 1;
4  fill("white");
5  textSize(30);
6
7  function draw() {
8    background("black");
9    if (mousePressedOver(sprite)) {
10     points++;
11     sprite.x = randomNumber(20,380);
12     sprite.y = randomNumber(20,380);
13   }
14   drawSprites();
15   text("Points: "+points, 15, 30);
16 }
```

Start the variable at 1

Add one

The Counter Pattern

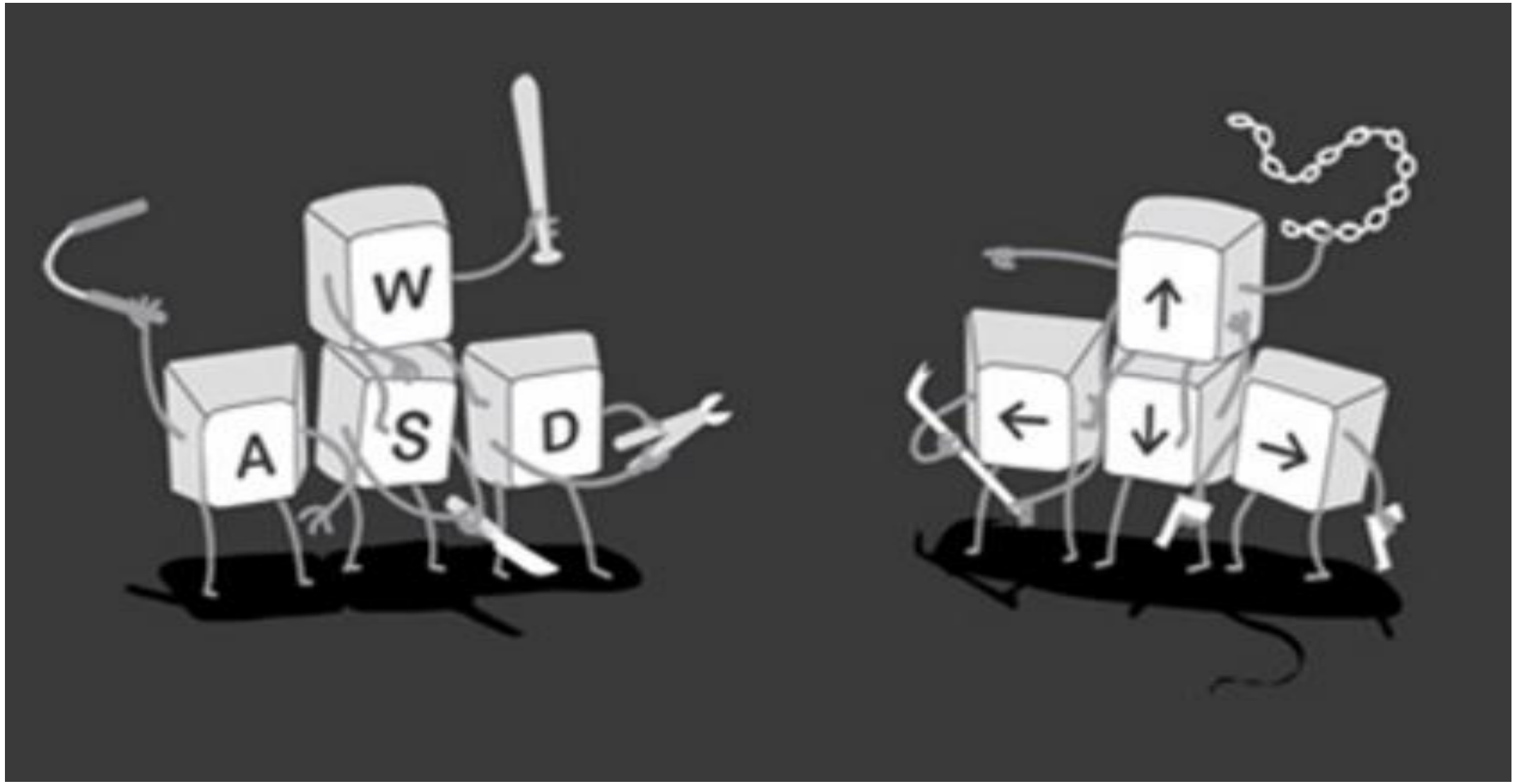
Can be used to keep score.

```
1  var sprite = createSprite(200, 200);
2  sprite.setAnimation("alienGreen");
3  var points = 1;
4  fill("white");
5  textSize(30);
6
7  function draw() {
8    background("black");
9    if (mousePressedOver(sprite)) {
10     points++;
11     sprite.x = randomNumber(20,380);
12     sprite.y = randomNumber(20,380);
13   }
14   drawSprites();
15   text("Points: "+points, 15, 30);
16 }
```

Start the variable at 1

Add one

Show on screen


W
A
S
D
↑
←
↓
→


F1
F2
F3
F4
Q
E
R
T
Y
F
G
H
Z
X
C
V
B
Thereifixedit.com

Keyboard & Mouse Input

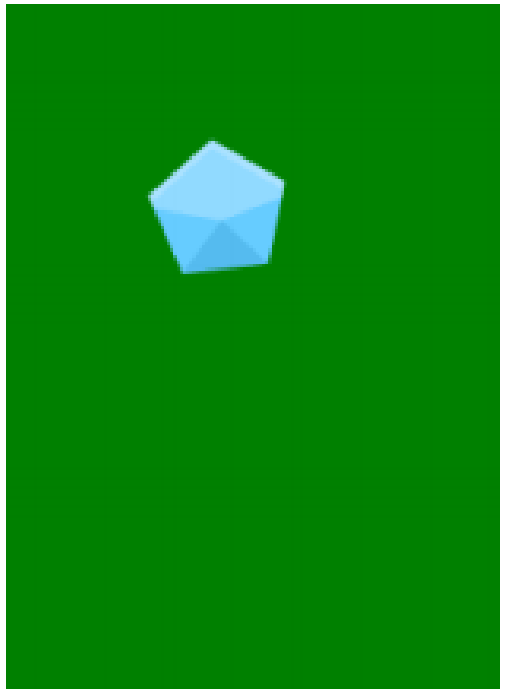
3.8 C

KeyDown	(a)	Continually checks if the key specified is pressed. Pressing a key once and holding it can result in multiple actions.
KeyWentDown	(b)	Checks if the key specified was pressed. It generates a single value when the key is pressed down, no matter how long a key is pressed.
KeyWentUp	(c)	Checks if the key specified was released. It generates a single value when the key is released, no matter how long a key is pressed.
MouseDown	(d)	Continually checks if the mouse button specified was pressed.
MouseWentDown	(e)	Checks if the mouse button specified was pressed. It generates a single value when the mouse is pressed down, no matter how long the mouse button is pressed.
MousePressedOver	(f)	Checks if the mouse is pressed over the sprite specified.

1. This code has an alien jumping around the screen. It keeps score for each time the user clicks on the alien. Fill in the code to make the program run.


```
1  var sprite = createSprite(200, 200);
2  sprite.setAnimation("alienGreen");
3  var points = 1;
4  fill("white");
5  textSize(30);
6
7  function draw() {
8    background(black);
9    if (mousePressedOver(sprite)) {
10     points++;
11     sprite.x = randomNumber(20,380);
12     sprite.y = randomNumber(20,380);
13   }
14   drawSprites();
15   text("Points: "+points, 15, 30);
16 }
```

2. This code moves around the diamond when the button Q is pressed. Fill in the blanks.



```
1  var sprite = createSprite(100, 100);
2  sprite.setAnimation("diamond");
3
4  function draw() {
5    background("green");
6    drawSprites();
7    if (keyDown("Q")) {
8      sprite.x = randomNumber(10, 380);
9      sprite.y = randomNumber(10, 380);
10   }
11 }
```