nerd

CODE JOURNAL

Mrs. Brandeberry

JavaScript

Student Practices

Problem Solving

Use a structured problem solving process to help address new problems

View challenges as solvable problems

Break down larger problems into smaller components

Persistence

Value and expect mistakes as a natural and productive part of problem solving

Continue working towards solutions in spite of setbacks

Iterate and continue to improve partial solutions

Creativity

Incorporate your own interests or ideas into your work

Experiment with new ideas and consider multiple possible approaches

Extend or build upon the ideas and projects of others

Collaboration

Work with others to develop solutions that incorporate all contributors

Mediate disagreements and help teammates agree on a common solution

Actively contribute to the success of group projects

Communication

Structure your work so that it can be easily understood by others

Consider the perspective and background of your audience when presenting your work

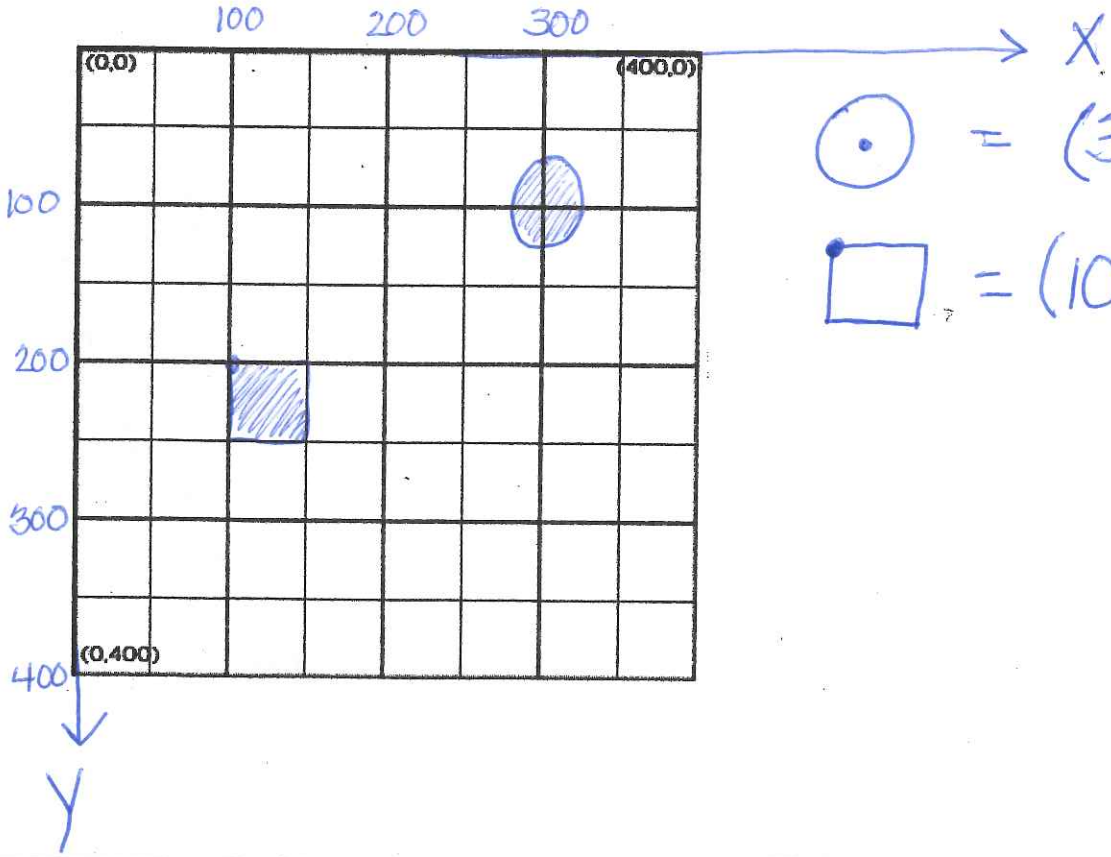
Provide and accept constructive feedback in order to improve your work

Table of Contents

Date	Title	Page

Lesson 2 – Plotting Shapes (computer grid)

(X, Y)



○ = (300, 100)

□ = (100, 200)

Lesson 3- Drawing in Game Lab

Vocabulary:

- Bug - Part of a program that does not work correctly.
- Debugging - Finding and fixing problems in an algorithm or program.
- Program - An algorithm that has been coded into something that can be run by a machine.

New Code:

- `fill(color)`
- `ellipse(x, y, w, h)`
- `rect(x, y, w, h)`

RGB- Red, Green, Blue

brown Rgb(105,63,35)

New Code

- rgb(red, green, blue)

1. Google RGB calculator to make any color
2. rgb(0,0,0) = BLACK
3. rgb(255,255,255) = WHITE

red green blue

Colors I like	Rgb(x,x,x)
soft blue	Rgb (99,150,232)
Pink	Rgb (232,99, 218)
neon green	Rgb (121,232, 99)
tiffany blue	Rgb (33, 235, 242)
beige	Rgb(255,213,154)

Hour of Code

1st - Latch-Key Kids

Game	Grade Levels
http://tynker.com	K-2
http://code.org	3-5

Goal

1. Everyone has fun
2. Guest leaves feeling like a 'coder'
3. Guest is the driver

Lesson 4 - Shapes and Randomization

Vocabulary:

- Parameter – An extra piece of information passed to a function to customize it

New Code:

- Background(color) → put this at beginning (order matters)
- randomNumber()

→ the bigger the range the bigger the movement

Lesson 5- Variables

Vocabulary-

- Variable - A placeholder for a piece of information that can change.
- camelCase – The first letter of the variable is usually lower case, and each new word starts with a capital letter. This helps you see the words without spaces (spaces are not allowed in variable names)

console.log ("score");

New Code:

- Declare variable - var counter;
- Assign value to variable counter=0;

Naming Rules:

- No spaces
- Can't begin with number
- Spelling counts
- Case-sensitive (Size ≠ SIZE)

```
score = 10 ;
```

Equals Sign

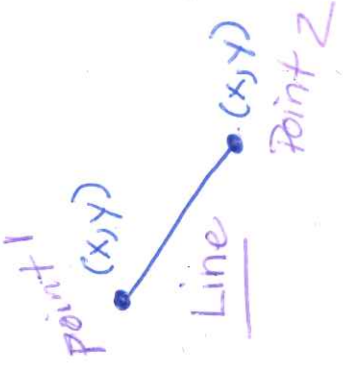
assignment value gets the value

EXAMPLE

```
Var eyeSize;
EyeSize = 20;
```

label value

or var. lives = 3;



text

textSize(30);

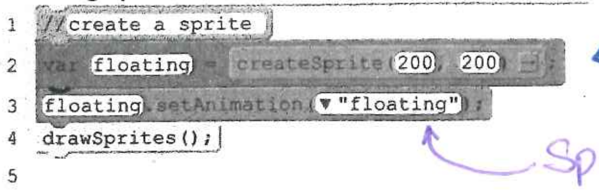
these come first

U3 L6

Shapes:
background(color) can use rgb values
rect(x, y, width, height)
ellipse(x, y, width, height)
line(x1, y1, x2, y2) one point second point
text(string, x, y, width, height) string=text
textSize(pixels) font size
Color and Style:
fill('color')
noFill()
stroke('color') border color
noStroke()
strokeWeight() thickness

U3 L6

Creating Sprites



```
1  //create a sprite
2  var floating = createSprite(200, 200);
3  floating.setAnimation("floating");
4  drawSprites();
5
```

← where (200,200)

sprite/picture

Line:

1. Is a note to the reader starts with //

2. The createSprite() block, which creates a new sprite at (200, 200) and assigns it to the variable label floating.

3. The sprite.setAnimation() block assigns an animation (or image) to the sprite.

4. Because sprites are just values stored as variables, they don't automatically get drawn on the screen. The drawSprites() block tells Game Lab to draw all of the sprites.

Background tan

Mini Project: Create a Gratitude Card

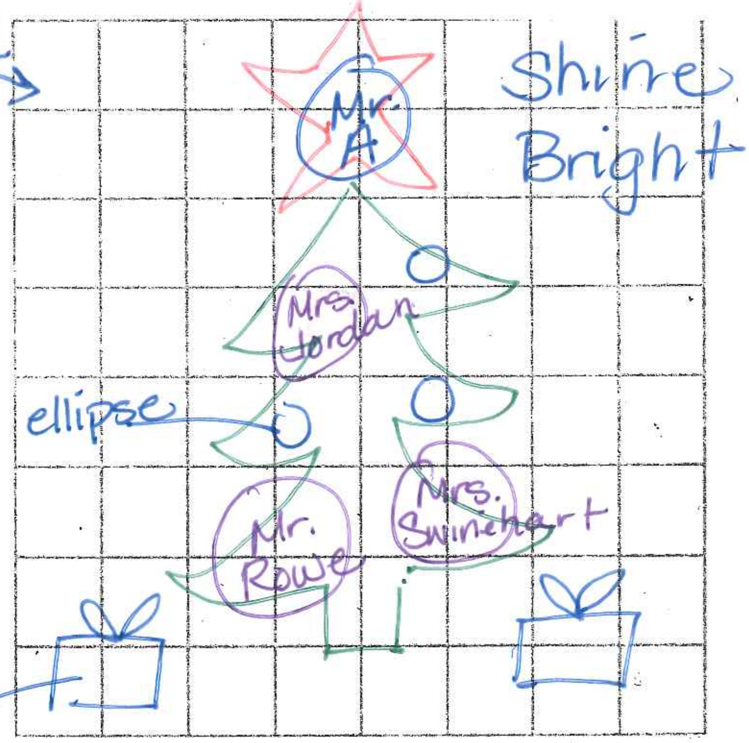
Required elements:

- ☒ Background
- ☒ At least 2 sprites
- ☒ Rectangle
- ☒ Ellipse
- ☒ Text

an adult at school

You will share your project with them.

For: Mr. Anschuetz



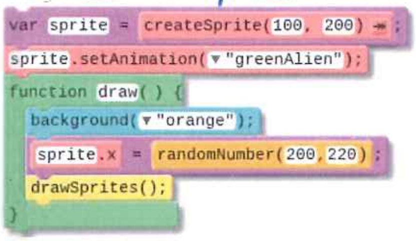
rect

Unit 3 Lesson 7- The Draw Loop

Vocabulary

- Animation - a series of images that create the illusion of motion by being shown rapidly one after the other
- Frame - a single image within an animation
- Frame Rate - the rate at which frames in an animation are shown, typically measured in frames per second

Example



```
var sprite = createSprite(100, 200);
sprite.setAnimation("greenAlien");
function draw() {
  background("orange");
  sprite.x = randomNumber(200, 220);
  drawSprites();
}
```

`sprite.x` — changes across

`sprite.y` - changes up and down

`sprite.rotation` -rotate/spin

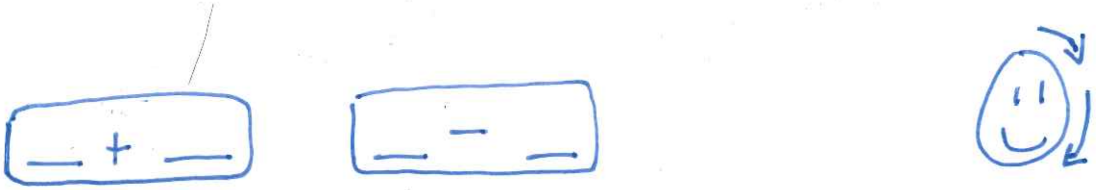
`sprite.scale` - size less than 1 - smaller
more than 1 - bigger

`sprite.visible` can see/not see

Unit 3 Lesson 8- Counter pattern

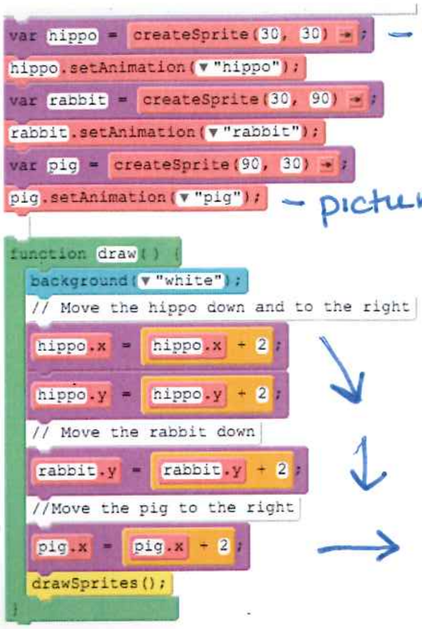
Vocabulary

- Expression - Any valid unit of code that resolves to a value.
- Variable - A placeholder for a piece of information that can change.



Example
sprite.rotation = sprite.rotation + 1

Unit 3 Lesson 9 – Sprite Movement



```
var hippo = createSprite(30, 30);
hippo.setAnimation("hippo");
var rabbit = createSprite(30, 90);
rabbit.setAnimation("rabbit");
var pig = createSprite(90, 30);
pig.setAnimation("pig");

function draw() {
  background("white");
  // Move the hippo down and to the right
  hippo.x = hippo.x + 2;
  hippo.y = hippo.y + 2;
  // Move the rabbit down
  rabbit.y = rabbit.y + 2;
  //Move the pig to the right
  pig.x = pig.x + 2;
  drawSprites();
}
```

- where

create variabe

- picture

↘

↓

→

Draw loop
Background
movement
movement
Draw sprite

sprite. x - 2 ←   sprite.y - 2 ↑

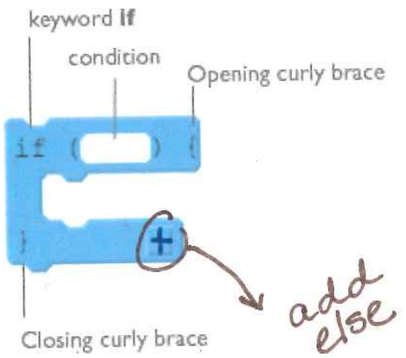
Unit 3 Lesson 10- Boolean Operators

Vocabulary

- **Boolean** - A single value of either TRUE or FALSE
- **Conditionals** - Statements that only run under certain conditions.
- **Expression** - Any valid unit of code that resolves to a value.

Unit 3 Lesson 11- Conditionals Boolean Operators

==	- is equal to	==
!=	-is not equal to	!=
>	- greater than	>
<	- less than	<
>=	- greater than or equal	
<= is less than or equal to	- less than or equal	



Vocabulary:

- Boolean Expression- in programming, an expression that evaluates to TRUE or FALSE
- If-Statement- The common programming structure that implements 'conditional statements'.

** Boolean Operators ask a question**

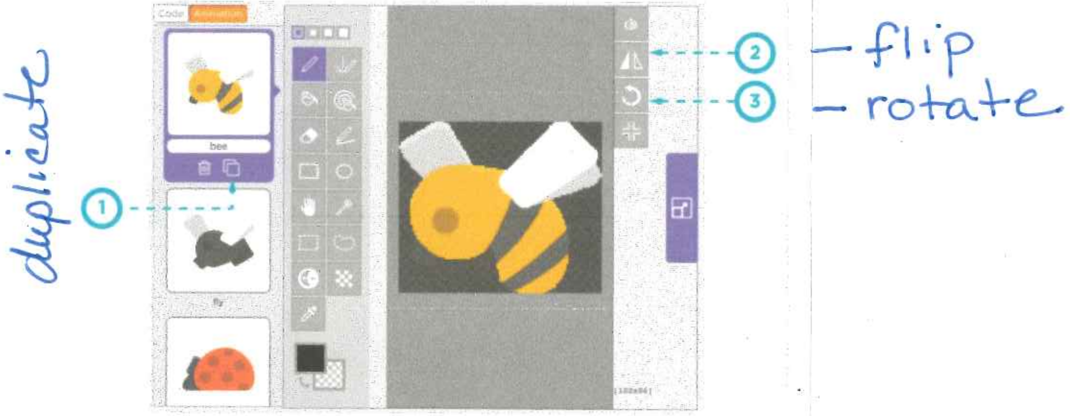
Unit 3 Lesson 12- Keyboard Input (and Edit Sprites)

New Code:

KeyDown(code)   example: keyDown("space");

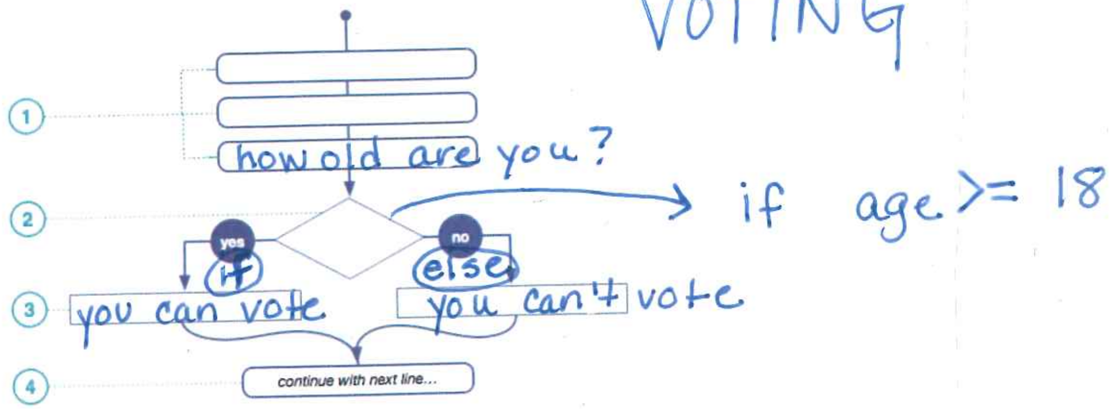
Copy, Flip, and Rotate

Three useful tools for creating multiple animations to use with the same sprite are *Copy*, *Flip*, and *Rotate*

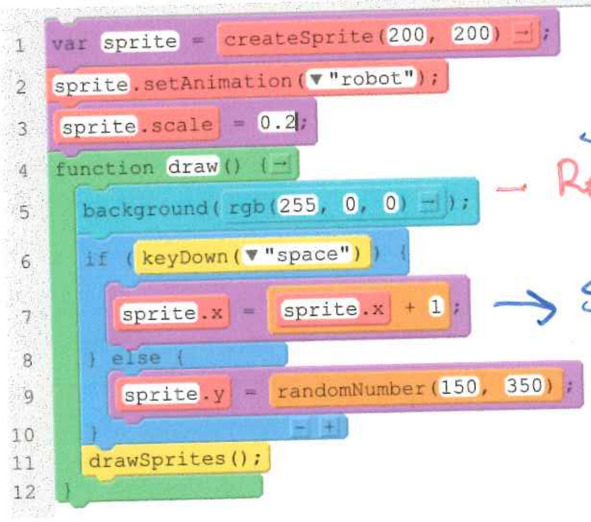


Unit 3 Lesson 13- Other Forms of Input (mouse, If-Else block)

VOTING



Unit 3 Lesson 13- Other Forms of Input (mouse, If-Else block)



```
var sprite = createSprite(200, 200);
sprite.setAnimation("robot");
sprite.scale = 0.2;
function draw() {
  background(rgb(255, 0, 0));
  if (keyDown("space")) {
    sprite.x = sprite.x + 1;
  } else {
    sprite.y = randomNumber(150, 350);
  }
  drawSprites();
}
```

} sprite

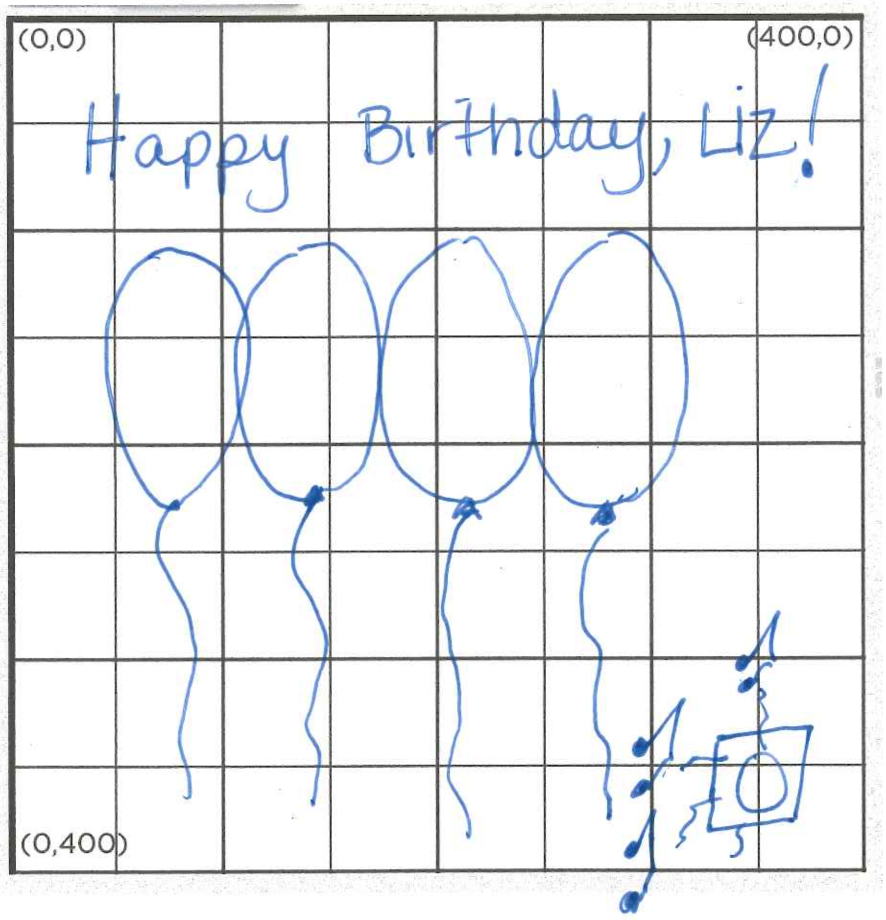
– Red

→ sprite moves right

– sprite jumps up and down

Unit 3 Lesson 14- Project

Criteria	Yes/No	Description
Uses at least 3 sprites	Yes	balloons tickets Detroit Fisher Hamilton Speaker
At least one sprite responds to user input (eg. `keyDown`, `mouseDidMove`) mouse or keyboard	Yes	all balloons – mouse speaker
Updates at least 3 different sprite properties in the draw loop (eg. `sprite.x`, `sprite.scale`, `sprite.visible`)	Yes	balloons invisible
Uses at least 1 conditional that is triggered by a variable or sprite property (eg. `sprite.y > 300`) Uses an if block if block		If balloon.y > 300 then balloon.visible = 0
Use a counter pattern to change a variable or sprite property. Use a Boolean block. (eg `score = score + 1`) + -		the ticket is going to rotate in a circle
The card includes text that tells the user how to use the card		Click on the ballons in order



Send to: Elizabeth

move mouse over balloons.

#1 pop ↓
Detroit

#2 pop ↓
Fisher Theatre

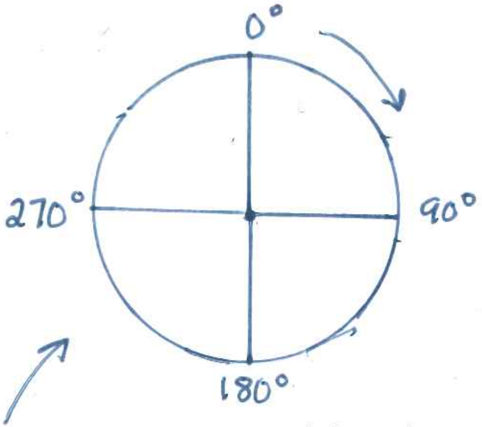
#3 pop ↓
Alexander Hamilton

#4 pop ↓
Hamilton tickets
music plays

Unit 3 Lesson 15- Velocity

New Code:

- sprite.rotationSpeed
- sprite.velocityX
- sprite.velocityY



hint:
0° is the exact same as 360°

Helpful hints:

- Rotation number is degrees- remember that there are 360 degrees in a full rotation
- Negative numbers – counter-clockwise
- Velocity will control the speed and direction
- Velocity blocks can be used inside and outside the draw loop

Math Review:

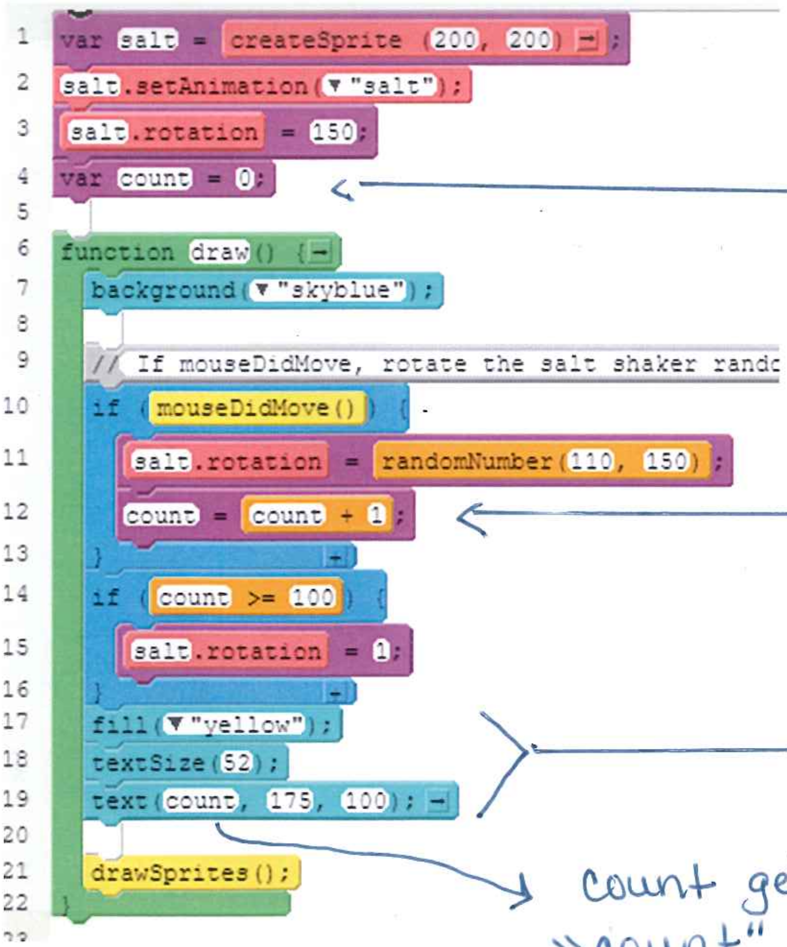
$\boxed{7 - 1} = \boxed{7 + -1}$

6 6

$\boxed{7 - -1} = \boxed{7 + 1}$

8 8

Creating a Score



```
var salt = createSprite(200, 200);
salt.setAnimation("salt");
salt.rotation = 150;
var count = 0;

function draw() {
  background("skyblue");

  // If mouseDidMove, rotate the salt shaker rando
  if (mouseDidMove()) {
    salt.rotation = randomNumber(110, 150);
    count = count + 1;
  }
  if (count >= 100) {
    salt.rotation = 1;
  }
  fill("yellow");
  textSize(52);
  text(count, 175, 100);

  drawSprites();
}
```

① Create a variable (a place holder to hold a value)

② gets the current value and adds one

③ code that lets You see the count/score

count gets value (example 1)
"count" gets letters (count)

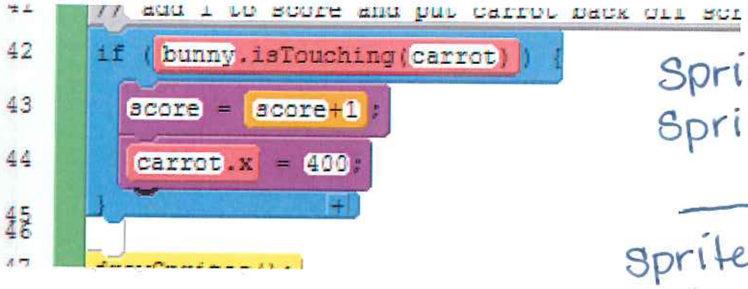
Unit 3 Lesson 16- Collision Detection

isTouching()

Writing out the math each time you want to check whether two sprites are touching can take a while, so a programmer created the isTouching block, which can check whether one sprite is touching another sprite (the target).

Scoreboard

You can also use isTouching to decide whether you should increase the score. In this game, the score is stored inside the 'score' variable. It is displayed on the screen using the text block.



```
// add 1 to score and put carrot back off scr
if (bunny.isTouching(carrot)) {
  score = score+1;
  carrot.x = 400;
}

drawSprites();
```

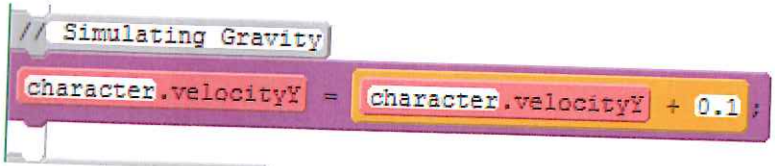
Sprite.x
Sprite.y } Control position (place)

sprite.velocityY – controls speed and direction

Unit 3 Lesson 17- Complex Sprite Movement

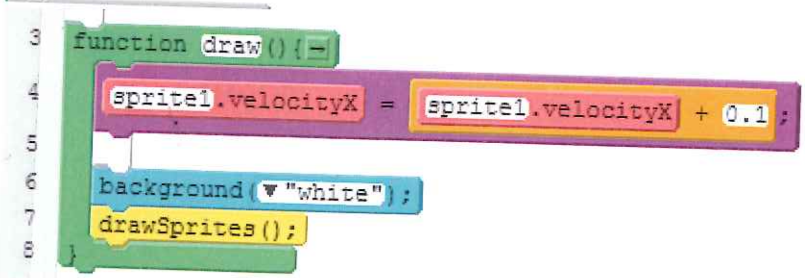
Velocity and the Counter Pattern

Using a sprite.velocityX property with the counter pattern will change a sprite's velocity during the program. This makes the sprite speed up.



```
// Simulating Gravity
character.velocityY = character.velocityY + 0.1;
```

} Sprite speeds up ↓



```
function draw() {
  sprite1.velocityX = sprite1.velocityX + 0.1;

  background("white");
  drawSprites();
}
```

Sprite speeds up →

Hint

- * positive and negative control direction
- * the bigger the number the faster the velocity

Unit 3 Lesson 18- Collisions

displace

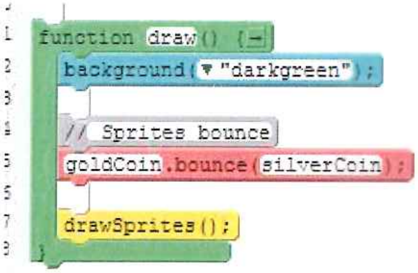
The displace block causes the sprite to **push** the target as long as they are touching each other. The sprite keeps moving normally. sprite.displace(target) both move same direction

collide

The collide block makes the **sprite stop** when it runs into the target. If the target is moving, it will push the sprite with it. The target keeps moving normally. (like the turtle and tree)

bounce

The bounce block makes the **sprite and the target bounce** when they touch each other. Both the sprite and the target change how they are moving.

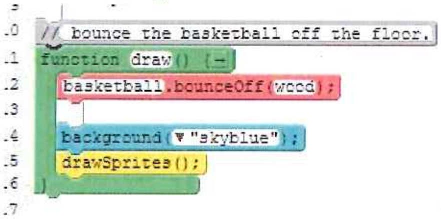


```
function draw() {
  background("darkgreen");

  // Sprites bounce
  goldCoin.bounce(silverCoin);

  drawSprites();
}
```

bounceOff

The bounceOff block makes the sprite bounce off the target. The target keeps moving normally. sprite bounces

basketball bounces → off floor.



```
// bounce the basketball off the floor.
function draw() {
  basketball.bounceOff(wood);

  background("skyblue");
  drawSprites();
}
```

Unit 3 Lesson 19 - Functions

Five Step instructions for morning routine.

1. brush teeth
2. make coffee
3. eat breakfast
4. take shower
5. get dressed.

1 get brush wet
2 put paste on brush
3 scrub teeth
4 brush tongue
5 rinse mouth

1. brush in circular motion
2. brush each tooth bottomtotop
3. brush back side of teeth
4. brush top of teeth
5. floss between teeth

Unit 3 Lesson 22 – Final Project

Criteria	Yes/No	Comments
The game includes at least three **sprites** with appropriate animations		
The game includes at least one variable that is updated during the game and affects the way the game is played (example- score)		
The game has at least two backgrounds that are triggered to change by a conditional		
The draw loop has been divided up into appropriately named **functions**		
The program code makes use of whitespace, indentation, and comments to aid the reader		
The game is playable and works as intended		
The game is creative and original		

KEYBOARD SHORTCUTS

Ctrl + B **Bold** **B**

Ctrl + I *Italic* *I*

Ctrl + U Underline U

Ctrl + X Cut

Ctrl + C Copy

Ctrl + V Paste

Ctrl + Z Undo

Ctrl + Y Redo

Ctrl + S Save

Ctrl + P Print

Ctrl + A Select All

Ctrl + K Insert Link

Ctrl + J Justify Paragraph

Ctrl + D Font options

Date	Speed	Adjusted
#1		
#2		
#3		
#4		
#5		
Final (goal 30 WPM)		