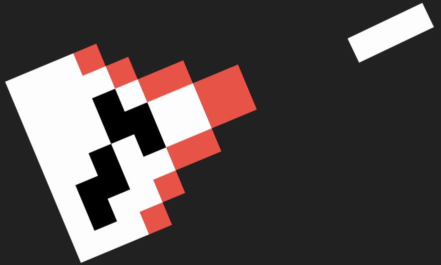
How to Make a Retro Game

Ryan Young, Natalie Haney, Haniel Villarreal, Jaehyun Na



Who Are We?

Ryan Young -

Senior in Data Science/Analytics

Looking towards masters in AI and/or a second bachelors in Environmental Science/an environmental conservation field.

Natalie Haeny-

Senior in Software Development

Hoping to continue my understanding of java and learn new languages like C# and JavaScript.

Haniel Villarreal -

Sophomore in Systems and Security

Aspiring to master the ability to provide protection to some of companies greatest digital assets

Jaehyun Na-

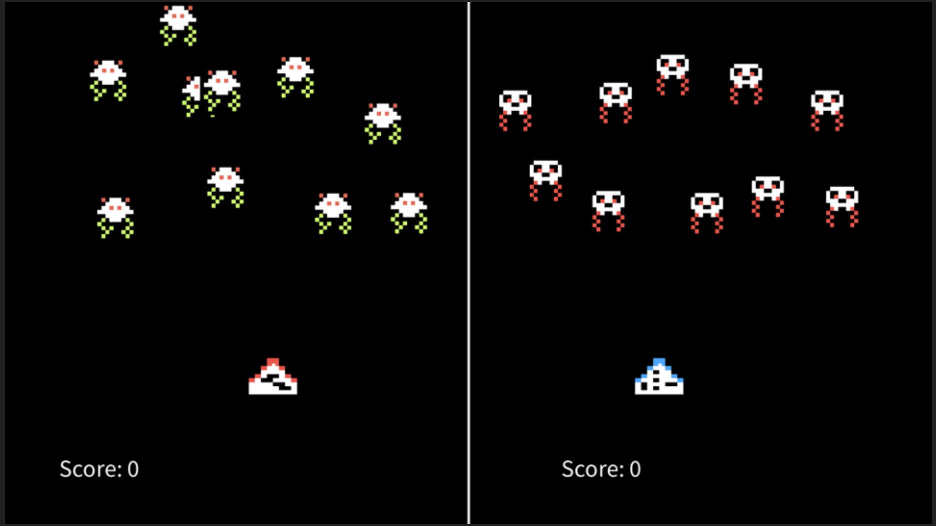
Sophomore in Software Development

Wants to be a Developer

Our project

Our group worked together to develop our own take on the arcade game, Space Invaders.

Our game incorporates the same retro style of the original while adding new features like our multiplayer function.



TAP Expo

This event allows for students and faculty to see the different tech-based designs and games that TAP students have created.

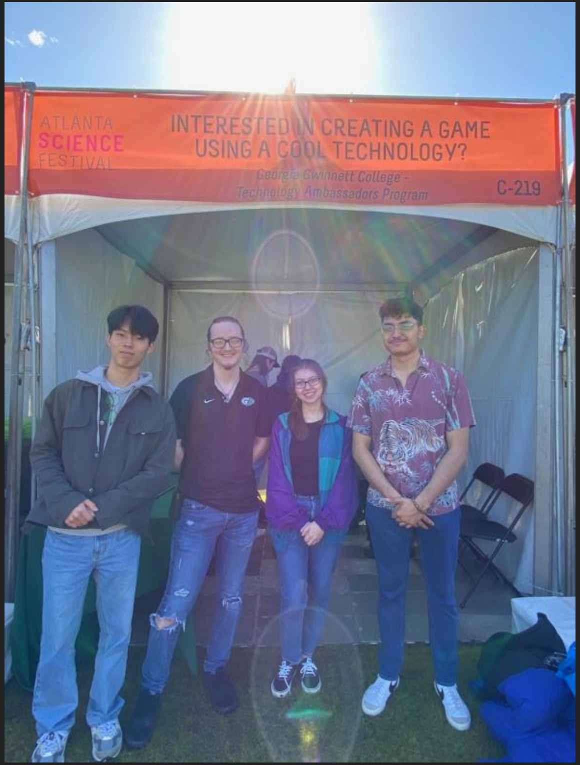
This is an amazing opportunity to get other students interested in tech and show them what they are truly able to create while also allowing TAP students to get good feedback on their work.

This event also allows for students to get to know others interested in tech as well as different tech professors.

Atlanta Science Festival

At the Atlanta Science Festival, we did much of what was done at TAP Expo, getting people in our booth, getting people interested in programming and what it can do.

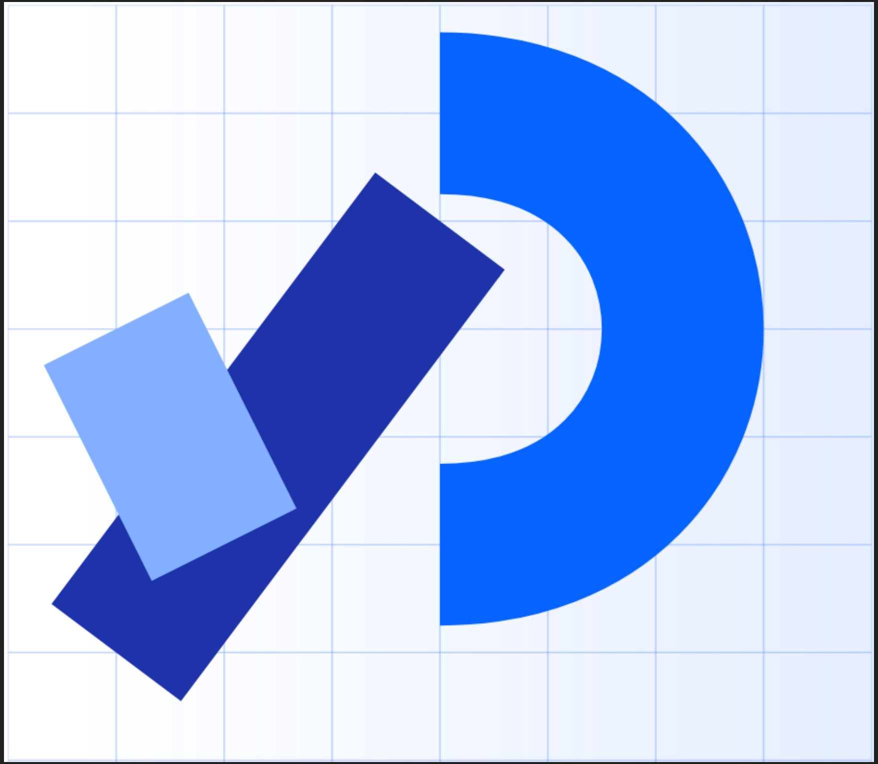
We got interest from lots of different people in all subject areas. It was a great time for everyone.


ATLANTA SCIENCE FESTIVAL
INTERESTED IN CREATING A GAME USING A COOL TECHNOLOGY?
Georgia Gwinnett College - Technology Ambassadors Program
C-219

What is Processing?

We built our project using Processing. So what is it?

Processing began as a JDK language, or Java Development Kit. It is more of its own thing after Processing 4, but it still uses Java as its base language. More simply, Processing is a software 'sketchbook', used for easy Java introduction with a focus for learning and prototyping different code, making a game very simple to produce.



Downloading Processing

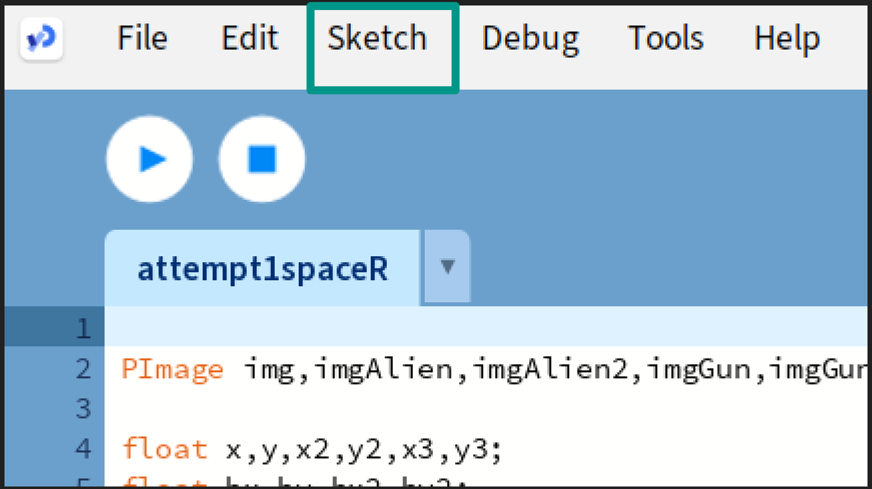
Processing should already be installed on school computers, but if you want the most recent version for your personal computer, go to:

https://processing.org/download

Go to your downloads folder, extract the zip file to your desired location and run the processing.exe application to open the sketchbook.

Setting up our game

After opening up processing, one more step must be done in order to run our game.



```
PImage img,imgAlien,imgAlien2,imgGun,imgGur

float x,y,x2,y2,x3,y3;
```

After **clicking on Sketch** a drop down menu should appear.

Click on "**Add File…**"

Finally there should be **7 png files** to individually add into the game.

The game should be all ready now!

Processing Basics

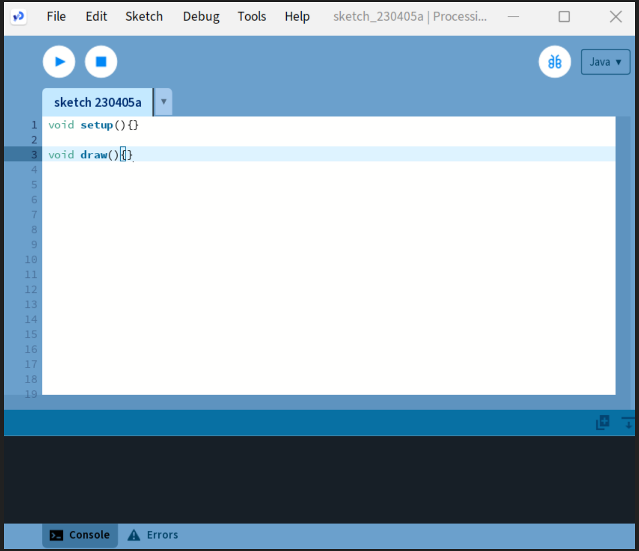
Processing needs at least two functions to properly work.

The **void setup()** and the **void draw()**

The **setup()** only runs once and is where environment properties like screen size are initialized.

The **draw()** function continuously executes the lines of code inside the function until the program is stopped. This allows objects to be animated seamlessly.

Add these two functions into processing



```
void setup(){}

void draw(){}
```

Processing Basics

Creating the screen is very important since that is where you will see everything you create!

Inside the **setup()** function add:

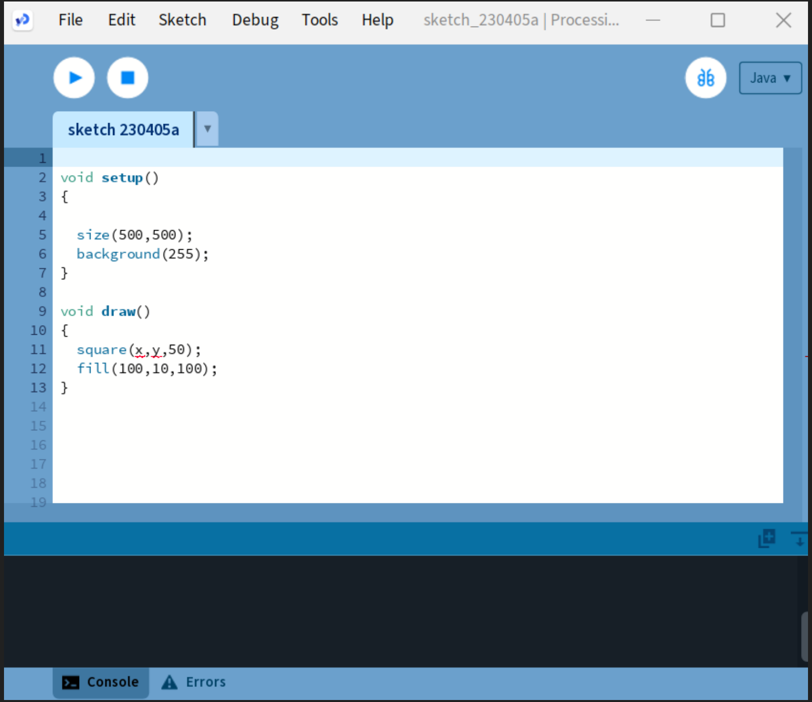
```
size(500,500);

background(255);
```

Next you are going to want to add an object onto the screen!

Inside the **draw()** function add:

```
square(x,y,50);

fill(100,10,100):
```

Don't forget the ;



```

void setup()
{

  size(500,500);
  background(255);
}

void draw()
{
  square(x,y,50);
  fill(100,10,100);
}
```

Parameters and Arguments

What are parameters?

Parameters are variables that define a value during a function definition.

Example:

```
fill(R,G,B);
```

The fill() function, in processing can, uses the rgb color model.

R = red; G = green; B = blue

As you can see, functions can contain multiple parameters. These, however must be separated by commas.

What are arguments?

Arguments are values passed to a function whenever the function is called.

Example:

```
fill(100,10,100);
```

Try putting this into your processing program.

What color is it?

Feel free to mess around with the values.

Processing Basics

Those x and y values must be store somewhere! Above the void setup() store the values by adding:

int x = 250;

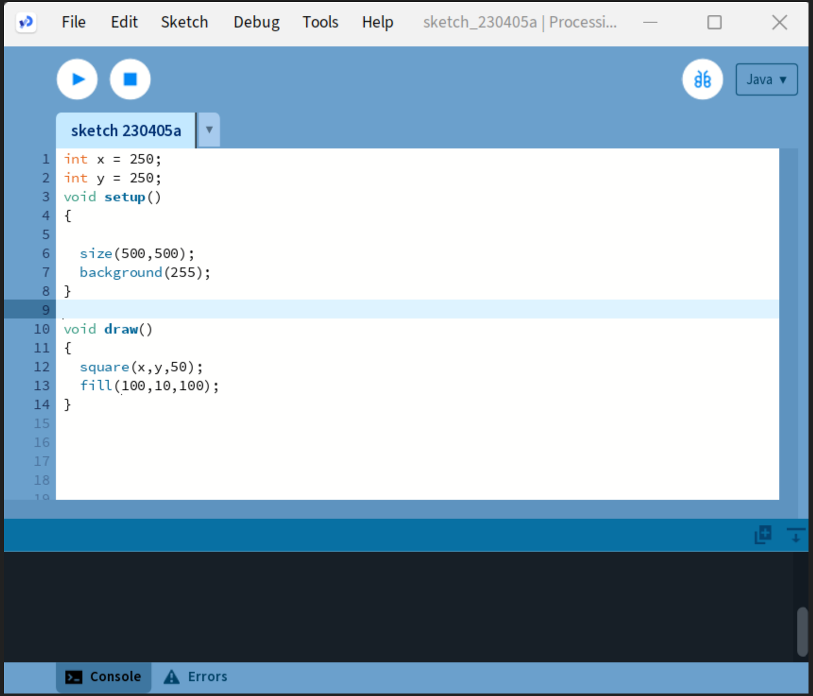
int y = 250;

The square should now appear on your screen!

Add two more value under those called:

```
boolean lPressed = false;

boolean rPressed = false;
```

It should look like this!



```
int x = 250;
int y = 250;
void setup()
{

  size(500,500);
  background(255);
}

void draw()
{
  square(x,y,50);
  fill(100,10,100);
}
```

What is an ‘if’ statement?

An ‘if’ statement is a way to *check* something. It returns what’s called a boolean, or a true/false.

For example:

```
x=2;

if (x>1):

  return True;
```

This would return True. We could then use that True return to trigger other things in the program.

We used these in the processing basics to check if a certain key is pressed for movement. We can use it to trigger other events as well.

If statements are incredibly useful, and interesting to play with. Like the ‘truthiness’ of various objects.

Processing Basics

Let's add movement!

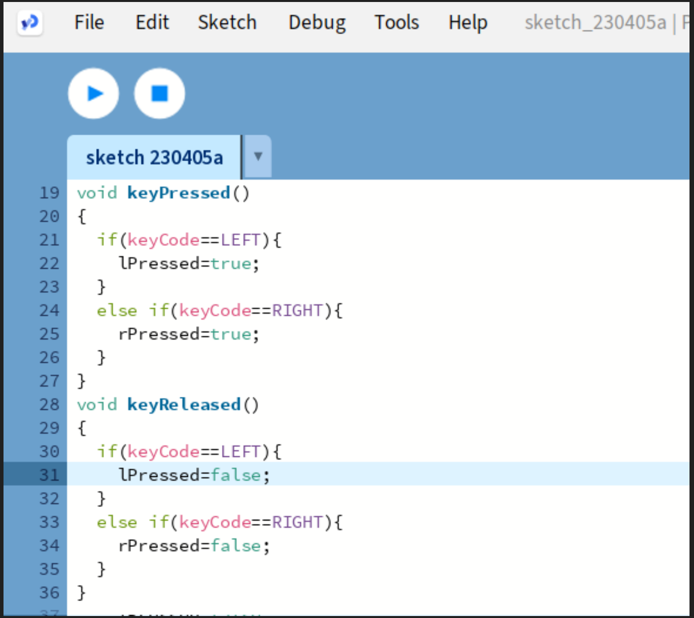
We need two new functions for this. Under the draw() add,

```
void keyPressed(){} and void keyReleased(){}
```

Inside the keyPressed() add an if statement

```
if(keyCode==LEFT){
        lPressed = true;
}
else if(keyCode ==RIGHT){
        rPressed=true;
)
```

Try and add keyReleased(){} on your own, but if you need help look here!



```
void keyPressed()
{
  if(keyCode==LEFT){
    lPressed=true;
  }
  else if(keyCode==RIGHT){
    rPressed=true;
  }
}
void keyReleased()
{
  if(keyCode==LEFT){
    lPressed=false;
  }
  else if(keyCode==RIGHT){
    rPressed=false;
  }
}
```

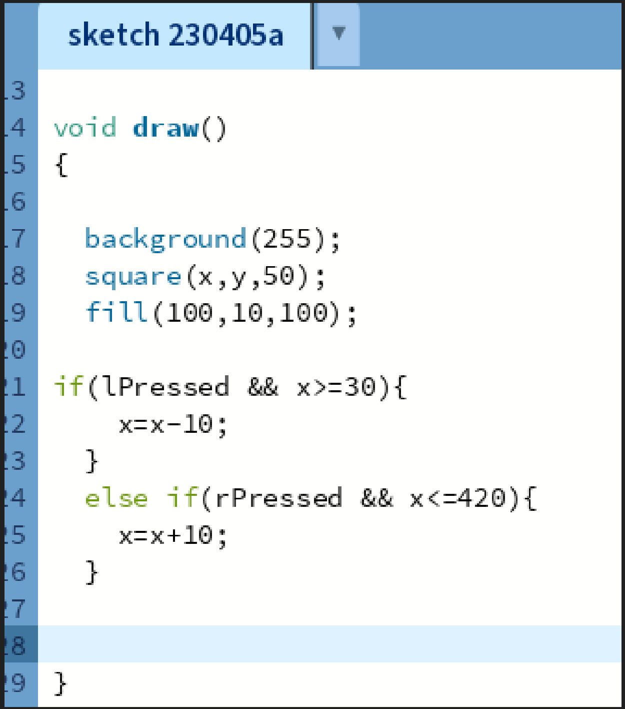
Processing Basics

After you finish, see what happens!

Finally, let's make it to were the square can travel along the x axis. Inside the draw() function add:

```
        if(lPressed && x>=30){

        x=x-10;

  }

  else if(rPressed && x<=420){

        x=x+10;

  }
```

It's important to add values less or greater than x to make sure the square stays inside the boundaries.


sketch 230405a

```

void draw()
{

  background(255);
  square(x,y,50);
  fill(100,10,100);

if(lPressed && x>=30){
    x=x-10;
  }
  else if(rPressed && x<=420){
    x=x+10;
  }


}
```

Time to Make a Game

Let’s take what we just did, and turn it around to make our game.

What is a Class?

In the simplest terms, a class is code for creating and manipulating *objects*.

An *object* is the core of *Object-Oriented Programming.*

Java, C++, C#, JavaScript, and Python are some common object-oriented programming languages.

What is Object-Oriented Programming?

It is a method of programming based around creating and manipulating *objects.*

How do we use Classes?

In our project, we use classes to overall simplify code and prevent us from unnecessarily repeating dozens of lines of code.

We have a class for the Player, for Enemies, and Bullets, because all of these are *objects* we use for our game to be played.