Challenge 1

For this project I decided to implement particles, this was in order to add atmosphere. I don't know how to implement particles and ran into several issues while implementing them. Starting position, reset, buildup of velocity and acceleration. the particles start at the bottom of the screen and are set with random variable in order to account for the unpredictable nature of "dust" the x changes as the particles disperse. Because of the nature of an array list, I had to implement a limit within the array which was altered in order to fit the movement and desired speed. The exit condition is implemented to constantly reset the limit to array making the effect infinite. This is not something I've dealt with before. There for on the initial creation of the array list there was no reset condition the array would not reset and the particles in the array would run out. Which confused me. It wasn't till I asked a friend about implementing the reset condition that I figured that I could implement one. Which in turn made me recalculate the conditions of the object recalculating the accelerations

```
class Particles {
  //arguments for partivle class
  //adding speed acceleration and position
  float size;
  PVector acceleration = new PVector();
  PVector velocity = new PVector();
  PVector position = new PVector();


  Particles() {
    //start the particles at the base of the screen have them accelerate to the top then reset with out the current velocity or acceleration
    //set the particals to random to have them dispears unevenly and randomly
    size = random(5, 15);
    position.x = random(0, 600);
    position.y = 400;
    velocity.y = random(1, 5);
    velocity.x = random(1, 3);
    //the acceleration is low to have the dust particals move slowly and restertt at a similar pace
    acceleration.y = random(-0.1, -0.5);
    acceleration.x = random(-0.1, 0.5);
  }

  void display() {
    fill(206, 206, 202,60);
    rect(position.x, position.y, size, size);
  }
  void update() {
    position.add(velocity);
    velocity.add(acceleration);
    //updating boarders and velocity so particals reset instead of flying of screen responing them and resetting the velocity
    if (position.y < 0) {
      position.y = 600;
      position.x = random(0, 600);
      acceleration.y = random(-0.1, -0.5);
      acceleration.x = random(-0.1, 0.5);
      //add new velocity when particals reset
      velocity = new PVector();
    }
  }
}
```

Challenge 2

It was initially unclear how I would implement the illusion of writing into my project. I wanted to make the design clear and really give the illusion that the player was typing out lines of code to program something to hand in to a theoretical teacher. I had decided that the best way to achieve this was to use an array holding preset lines of text. I struggle with the implementation of how to input either a character counter or individual word counter to set a win and lost condition. The idea was flawed and caused many more problems. Having to set individual values to keys or instead having a preset image swap over and over lines of code until a player reached a certain number of swaps to uphold a win condition. While implementing it I found that the code was too costly and large to be able to fully implement. Instead, I solved this by using the copy function. This would allow for an array list to hold an image that the copy function would call on repeatedly to create individual lines of "code". I swapped out the idea of an array of text for an array list to call on the constant steady image using the copy function.

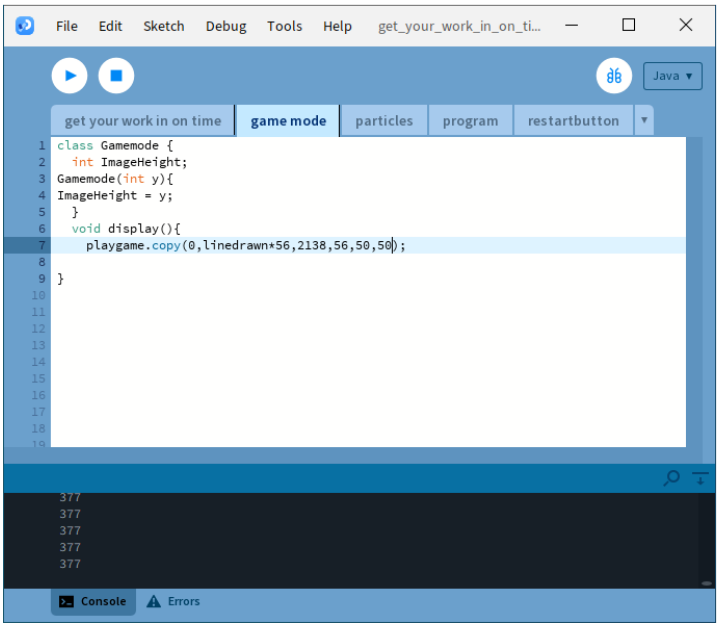
Here is the final function within the main tab, demonstrating the players ability to type and display new lines of code. Replacing the idea of individually coding in key pressed function and Booleans.

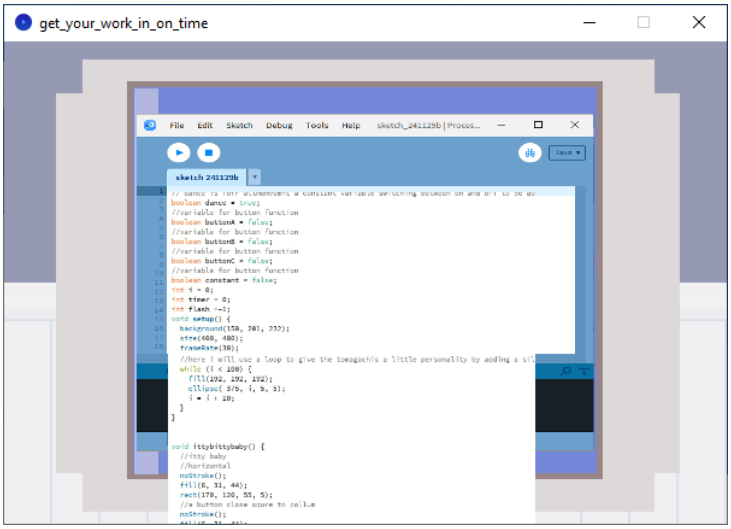
```
void keyPressed(){
  //when key presesed display next copy from the array list untill line 17 then delete first image in the list and replace bottom image as the code moves upwardes
  if(codelist.size() == 17){
  codelist.remove(0);
  }
codelist.add(new Gamemode());
linedrawn+=1;
```

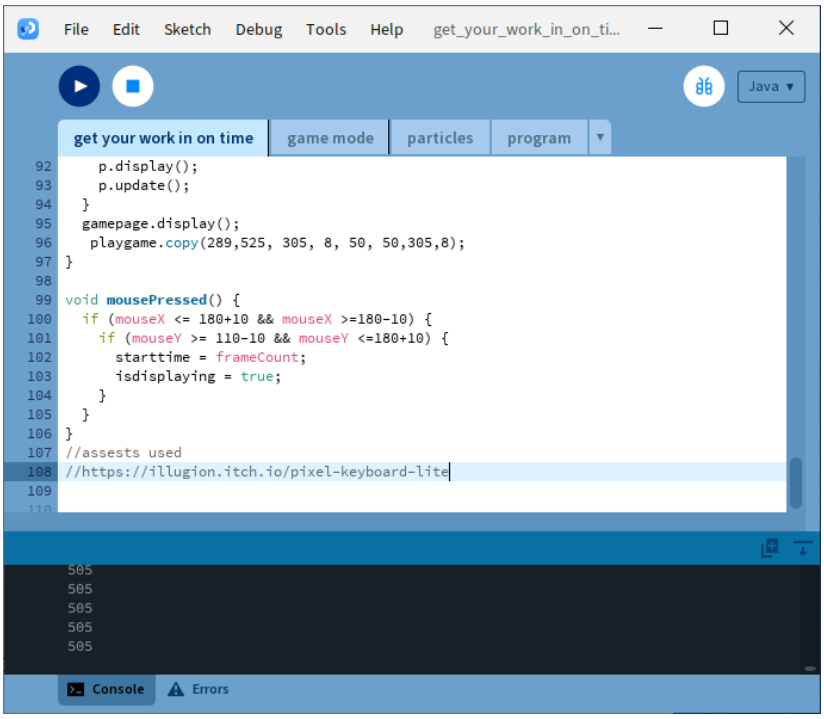
Challenge 3

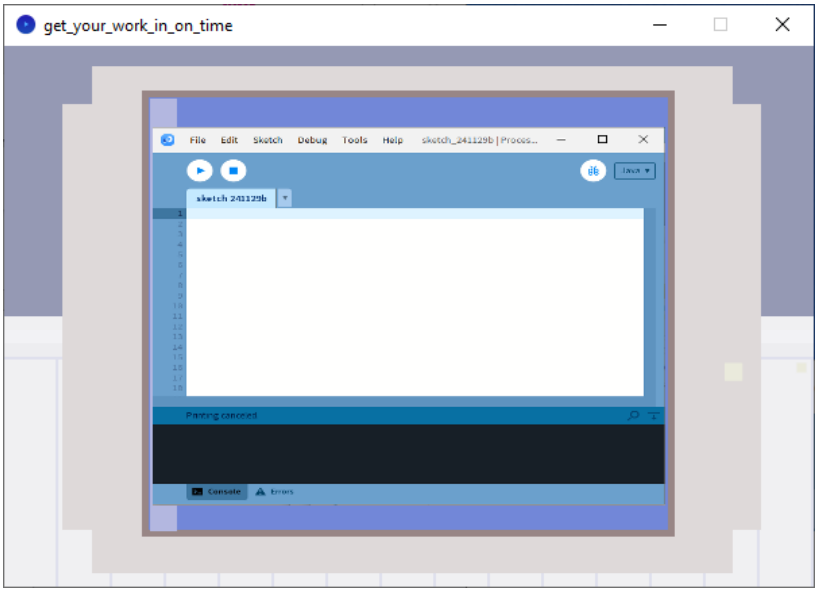
Using copy, to display the items in the array, trouble with sizing and resizing

While creating the program I realized I would have to use the function copy to create different states an array could draw from to create the illusion of writing code. In order to do this, I created an image of a pervious coding project by taking a photo of it. The problem occurred when first considering the size of the image as well as the creation of the image. You'll notice looking through the code that the image is a long state of code that I "patched" together in Miro as to make sure the position and state of the image was the same as the one presented in the blank sketch processing image. Because of this the size of the image is rather larger and uneven. To fix this I have had to crop and resize the state of the image. For the copy to display portions of the image within a set time limit. To do this I researched how many characters the average person could write in 30(time frame the window is open for) and divided the image by that amount to make intervals at which lines of code will be displayed.




get_your_work_in_on_time

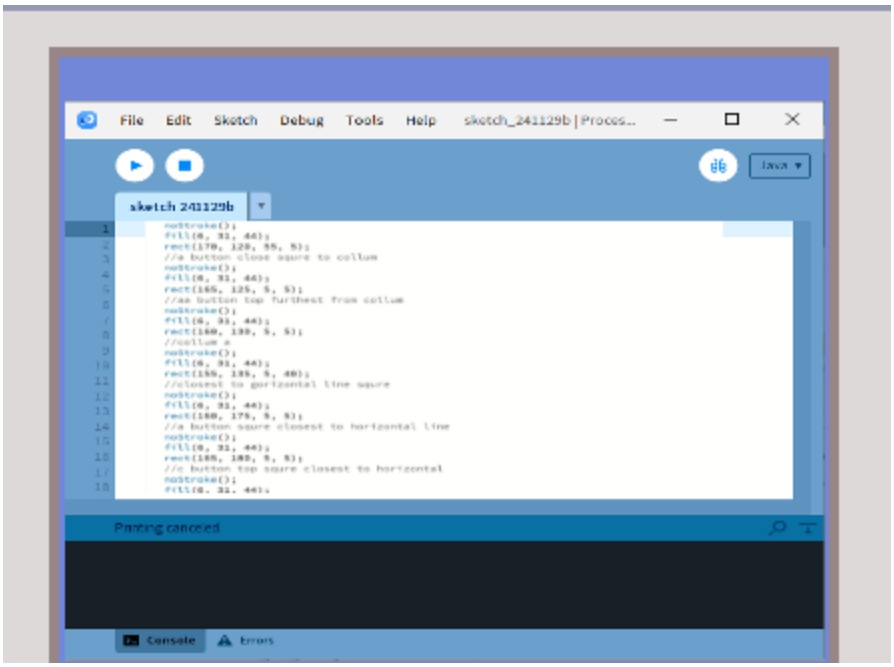

File Edit Sketch Debug Tools Help
get_your_work_in_on_ti...
Java
get your work in on time
game mode
particles
program
p.display();
p.update();
}
gamepage.display();
playgame.copy(289,525, 305, 8, 50, 50,305,8);
}
void mousePressed() {
if (mouseX <= 180+10 && mouseX >=180-10) {
if (mouseY >= 110-10 && mouseY <=180+10) {
starttime = frameCount;
isdisplaying = true;
}
}
}
//assests used
//https://illugion.itch.io/pixel-keyboard-lite
505
505
505
505
505
Console
Errors



See where the code attempts to use copy to display section of code no longer displays the coding image and instead plays nothing. In order to fix this I adjusted the call to he array placing the call to image within in the copy and resizing it before calling the array in the main page . limiting the size of the array so it only plays the copy between 17 lines from the beginning of the call to the end of the 17th line from the starting point.

```
void keyPressed(){
  //when key presesed display next copy from the array list untill line 17 then delete first image in the list and replace bottom image as the code moves upwardes
  if(codelist.size() == 17){
  codelist.remove(0);
  }
codelist.add(new Gamemode());
linedrawn+=1;
}
```

This line of code adds the new line of code at the 17th line while also removing the line at the beginning of the array. Giving the illusion of movement.


File Edit Sketch Debug Tools Help
sketch_241129b | Process...
sketch_241129b
noStroke();
fill(0, 31, 44);
rect(170, 120, 55, 5);
//a button close squre to collum
noStroke();
fill(0, 31, 44);
rect(165, 125, 5, 5);
//aa button top furthest from collum
noStroke();
fill(0, 31, 44);
rect(160, 130, 5, 5);
//collum a
noStroke();
fill(0, 31, 44);
rect(155, 135, 5, 40);
//closest to gorizontal line squre
noStroke();
fill(0, 31, 44);
rect(160, 175, 5, 5);
//a button square closest to horizontal line
noStroke();
fill(0, 31, 44);
rect(165, 180, 5, 5);
//c button top squre closest to horizontal
noStroke();
fill(0, 31, 44);
Pairing canceled
Console
Errors