Three-Tac-Toe

Implementation Log - Entry #1

Ravindu Hewabaddage
01-12-2024

Challenge

How does the grid know which player is clicking and how does it store data?

Brainstorming

How do we handle the different players? How do we handle turns being transferred? Do we use classes for each player? How does the grid interact with these classes? When a player clicks on a playable tile, does the grid ever interact with those classes? Would the classes only exist to hold a draw function and possibly some animations, maybe different sound effects?

Proposed Solution

Implement classes for each player to hold draw() function (for now).
Clicking on the grid will directly interact with that tile and give it a value that designates it is of that player's. (Giving it a value will remove the default 0 which designates it's playable, so having anything that's not a 0 automatically makes it unplayable)
The grid then calls upon the respective class and draws it's icon in that tile.

How I ended up implementing it

This section details how I ended up implementing this feature and what I learnt from it or any issues I encountered while implementing.

```
class Tiles {
  PVector coordinates;
  int type;

  Tiles(int tempType, PVector tempCoordinates) {
    coordinates = tempCoordinates;
    type = tempType;
  }
```

So, first of all I created the Tiles class. This class is for making the clickable tile that we have. The grid will be made up of tile objects made from this class. It stores the tiles coordinates and it's type.
Type refers to who owns the tile basically.
0 is unowned, 1 is cross, 2 is circle, 3 is triangle.
No one can click on any tile that is greater than 0 (i.e. owned)

```
  boolean isMouseHovering(){
   if(mouseX>coordinates.x-20 && mouseX<coordinates.x+20 &&  mouseY>coordinates.y-20 && mouseY<coordinates.y+20){
    return true;
   } else  {
    return false;
   }
  }
```

It also contains a boolean return type to tell the program if the mouse is hovering over it or not. This is used during mousePressed to see which tile the mouse is clicking on.

The GridManager() constructor then creates the grid using nested for loops by spawning in tiles and giving them unique coordinates.

```
GridManager() {

  for (int x=0; x<10; x++) {
    for (int y=0; y<10; y++) {
      PVector tempCoordinates = new PVector(x*40 + 20, y*40 + 20);
      t = new Tiles(0, tempCoordinates);
      println("tile coords: "+tempCoordinates);
      tileList.add(t);
    }
  }
}
```

I then had 2 functions draw in a specific order to achieve the aesthetic I wanted the game to have.

```
void draw() {
  background(255);
  if (inMenu) {
    displayMenu();
  }
  if (inPlay) {
    GM.displayColorGrid();
    GM.displayPlayingGrid();
  }
}
```

This function draws the color changing background

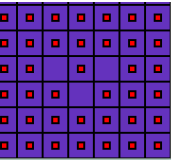
This function draws the actual tokens that show up on the grid

```
void displayColorGrid() {
  for (int x=0; x<10; x++) {
    for (int y=0; y<10; y++) {
      perlinC +=0.00001;
      fill(100,50,255-255*noise(perlinC+noise(x)+noise(y)));
      rect(40*x + 20, 40*y + 20, 40, 40);
    }
  }
}

void displayPlayingGrid() {
  for (Tiles t: tileList){
    t.displayTile();
  }
}
```

Here is the code for both the functions so you can see how I've used perlin noise to make the color changing background.

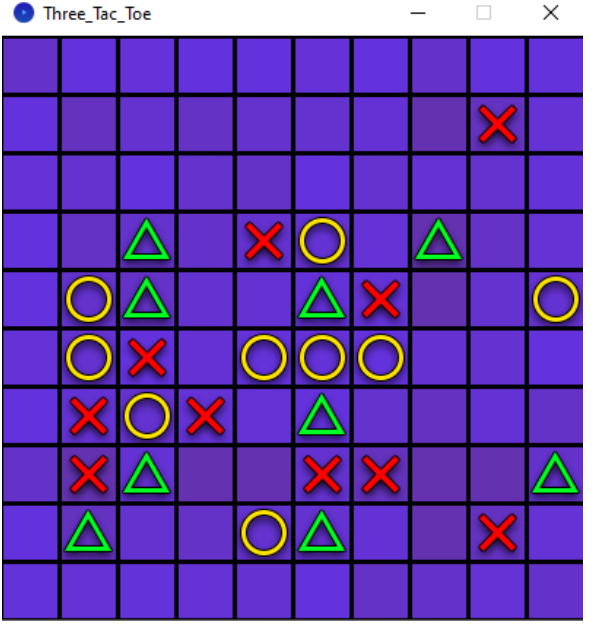
```
void displayTile() {
  strokeWeight(3);
  if(type ==0){
    fill (255,0,0);
    rect(coordinates.x, coordinates.y, 10,10);
  }
  if(type ==1){

  }
  if(type ==2){

  }
  if(type ==3){

  }
}
```

For testing purposes I drew a red square in the middle of the UNclicked tiles. Below is an image with those squares. Upon clicking on a tile the square disappears. I lcicked on 3 different squares in the center.



```
void displayTile() {
  strokeWeight(3);
  if (type ==0) {
  }
  if (type ==1) {
    image(tokenList[0], coordinates.x, coordinates.y);
  }
  if (type ==2) {
    image(tokenList[1], coordinates.x, coordinates.y);
  }
  if (type ==3) {
    image(tokenList[2], coordinates.x, coordinates.y);
  }
}
```

I have now drawn the sprites for the player tokens and added them to the tiles class.



Yippee!

Three-Tac-Toe

Implementation Log - Entry #2

Ravindu Hewabaddage

01-12-2024

Challenge

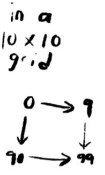
How do we detect 3 in a row?

Brainstorming

Every mousepress I have to check all the tiles to see if there are 3 in a row? Have to use a limited list to hold 3 values and see if all three values are the same as it iterates through all the tiles. This feels like such a Human Resource Machine type problem LOL. Easy to check rows but how do we do columns and diagonals? Maybe iterating through all the tiles is not correct, we have to find all tiles played and check around them to see if there are 2 of the same type in a line around them.

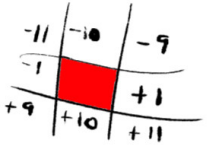
We need to make sure that while we detect 3 in a rows, we don't mistakenly count 3 that aren't connected like with stray tokens like the top left X.

So let's take a look at how the grid and how it's tiles are going to be numbered.

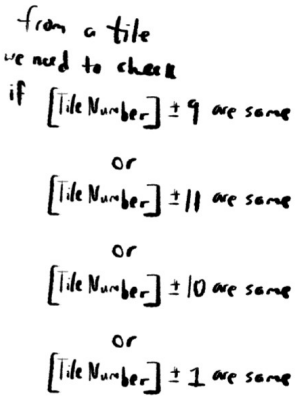


Let's take a 10x10 grid as an example. The first tile will start with a 0 and go as far as 9
It'll start a new column every 10 (0, 10, 20... 90)
Therefore we can conclude that tiles above and below a selected tile will be numbers [Tile Number] + or - 10
and tiles adjacent to a selected tile will be [Tile Number] + or - 1.

With that figured out, we can now draw this:



Notice how all the pairs we want to check have the same number they just differ in the sign.



Now we have this.
Every mousePress we need to iterate through all the tiles and when we find a tile with a token in it we need to then check around that token to find 3-in-a-rows.

Possible Issues

We do not want the program to detect strikes (3-in-a-rows) using tokens that have already been used in a different strike. How can we prevent that from happening?
We also don't want the program to keep adding scores to players every turn a mouse is clicked as it keeps iterating through every token and re-finding the same strikes.

Solution

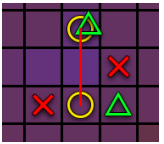
Iterate through all tiles in a grid until an unmarked token is found.

Check all opposite tile pairs around that token to see if they have the same (unmarked) token on them.

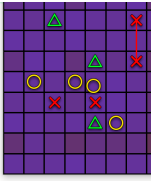
If yes set marked boolean to true on those tokens and add score and draw a line across them. (This will solve problem of marked tokens getting re-marked or being used in other strikes.

Bugs

The scourge of the universe... Look at that. How is that even possible. Their mere existence doesn't even make sense. They lead despicable, pathetic, unfounded li-Oh wait I forgot I changed my grid to be 9x9 I just have to make the code reflect that...



```
      if (GM.tileList.get(t.TN-9).type == t.type && GM.tileList.get(t.TN+9).type == t.type) {
        strikeFound(GM.tileList.get(t.TN-9).coordinates, GM.tileList.get(t.TN+9).coordinates);
      }
      if (GM.tileList.get(t.TN-11).type == t.type && GM.tileList.get(t.TN+11).type == t.type) {
        strikeFound(GM.tileList.get(t.TN-11).coordinates, GM.tileList.get(t.TN+11).coordinates);
      }
      if (GM.tileList.get(t.TN-10).type == t.type && GM.tileList.get(t.TN+10).type == t.type) {
        strikeFound(GM.tileList.get(t.TN-10).coordinates, GM.tileList.get(t.TN+10).coordinates);
      }
      if (GM.tileList.get(t.TN-1).type == t.type && GM.tileList.get(t.TN+1).type == t.type) {
        strikeFound(GM.tileList.get(t.TN-1).coordinates, GM.tileList.get(t.TN+1).coordinates);
      }
    }
  }
}

void strikeFound(PVector tempStartCoords, PVector tempEndCoords) {
  clickEnabled = false;
  strike = new Strikes(tempStartCoords, tempEndCoords);
 }
}
```



```
turn: 1
tile's type: 1
check strikes called in round: 1
Checking tile at coords: 2.0, 0.0 Tyle Type: 2
Checking tile at coords: 3.0, 2.0 Tyle Type: 3
Checking tile at coords: 3.9, 5.9999995 Tyle type: 1
Checking tile at coords: 4.0, 5.0 Tyle Type: 2
Checking tile at coords: 5.0, 4.0 Tyle Type: 3
Checking tile at coords: 5.9, 5.9999995 Tyle type: 1
Checking tile at coords: 6.0, 7.0000005 Tyle type: 3
Checking tile at coords: 5.9999995, 7.0000005 Tyle type: 2
Checking tile at coords: 7.0000005, 2.0 Tyle Type: 1
Checking tile at coords: 7.0000005, 4.0 Tyle type: 1
```

Three-Tac-Toe

Implementation Log - Entry #3

Ravindu Hewabaddage
01-12-2024

Challenge

How do we draw a line through a strike?

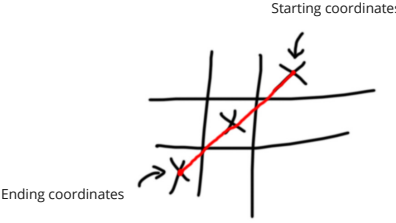
Brainstorming

What should the line do:
Needs to go through tokens when they get detected to be in a strike.
It needs to be in the right direction
It needs to be unchanging
It needs to be permanently displayed, until the game restarts.

We can use the coordinates of both the tiles around the tile the program's detecting a strike on to draw.
Do we have a strike class that has a draw() function with 2 PVector inputs that take coordinates from those tiles in the form of PVectors? the program upon detecting a strike creates a strike object from this class and adds it to an arraylist that is used in draw to draw all the strikes. I will not lie, I thought this was going to be way harder than it turned out to be.

Solution

Create a 'Strike' class
To initialize a strike object it takes in 2 PVectors and saves them as PVector start and PVector end
These 2 PVectors are going to be the 'coordinates' PVectors from the 2 tiles that form a strike around a center token.
The main program will initialize a strike object with these 2 coords and store it in an arraylist.
In the draw() function, it'll iterate through this list and call the display() function WITHIN the object to have them display permanently while the gamescreen is running.



Three-Tac-Toe

Implementation Log - Entry #4

Ravindu Hewabaddage
01-12-2024

Challenge

How to deal with draws and decide the winner

Brainstorming

To decide who the winner is we must first compare player 1 and 2 and see who has the most points, then compare that player with player 3 and see who has the most.
How do we detect if neither player has more than the other? Easy to do but I want to find the most efficient way of doing it.
Can I have if A > B then else B> A then another else statement to determine what happens if A & B are the same? I could add 2 conditions to the If statement like so: If(A>B && A!=B) then another if statement to check if B<A then another if statement to see if they are equal.

sketch 241201a

```
int A = int(random(0,3));
int B = int(random(0,3));
int C = int(random(0,3));
String winner;

void start() {
  println("A: " + A +"\nB: " + B +"\nC: " + C );
  if (A>B) {
   println ("A larger than B");
   if (A>C) {
    println ("A is larger than C too");
   }
  }
  if (B>A){
    println ("B larger than A");
   if (B>C) {
    println ("A is larger than C too");
   }
  }
  if (B == A){
   println("A and B are equal");
   if (B>C) {
    println ("A and B are larger than C");
   }
   if (B<C) {
    println("C is larger than A and B");
   }
   if (B ==C ) {
    println ("all 3 are equal");
   }
  }
```

```
A: 2
B: 0
C: 1
A larger than B
A is larger than C too
```

I'm still missing certain statements like C being larger than B. There has to be a much more efficient way of doing this without taking this many lines. Imagine an HRM workspace...

I looked up array functions on processing reference because I remembered there was a way to sort lists. This is me experimenting with the sort function to see what it does:+

```
//trying to use sort
for(int i=0;i<3;i++){
 scoreList[i] = int(random(0,3));
}
println("original list item: ");
for(int i=0;i<3;i++){
print(scoreList[i] + ", ");
}
scoreList = sort(scoreList);
println("sorted list item: ");
for(int i=0;i<3;i++){
print(scoreList[i] + ", ");
}
```

```
original list item:
1, 0, 1, sorted list item:
0, 1, 1,
```

The only issue with sorting is, I don't know which score belongs to who anymore.

```
//trying to use sort
scoreList[0] = "A: "+ int(random(0, 3));
scoreList[1] = "B: "+ int(random(0, 3));
scoreList[2] = "C: "+ int(random(0, 3));

println("original list item: ");
for (int i=0; i<3; i++) {
  print(scoreList[i] + ", ");
}
scoreList = sort(scoreList);
println("\nsorted list item: ");
for (int i=0; i<3; i++) {
  print(scoreList[i] + ", ");
}
```

```
A: 2
B: 2
C: 1
A and B are equal
A and B are larger than C
original list item:
A: 1, B: 0, C: 2,
sorted list item:
A: 1, B: 0, C: 2,
```

Tried sorting string lists and it doesn't seem to do anything.

What if instead of finding out who has the most points we just plot all the points in a bar graph so it's clear who won and who didn't. This way we don't need to change the text to match each person. (I wasn't planning on using text, just a sprite with the words "you won" or whatever to match the situation.
But how do we plot a bar graph?
Using rectangle with constant width but changing height, but y position also needs to change if rectMode is center, so if we change that to CORNERS we can subtract a certain amount of pixels from the ending y value to get height. We can draw a horizontal line across from each bar to the left that shows the value and an image of the player icon above the bar to show whose it is.
Issue: How would we show changing text positions with only 1 text function?
Solution: have a function that requires 3 inputs, position x, position y and text then inside this function a text() function takes the inputs and displays itself. Technically I'll only be using 1 text function, it'll just be called multiple times.

Solution

Draw a bar graph with all the scores so it communicates who won with score being shown on the sides of the graph using a function that gives it position and score.

Three-Tac-Toe

Implementation Log - Entry #5

Ravindu Hewabaddage
01-12-2024

Challenge

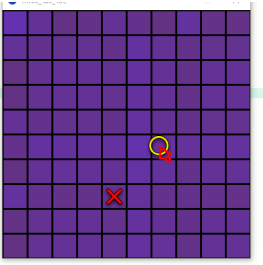
How to signify to the player whose turn it is.

Brainstorming

This one won't be too difficult, I already know how I would go about doing it.
This question was posed to me by my friend Sebastian who I was working next to in the lab with. It's a simple question that I never really would have thought of since I was just making the game and not playing it. Really shows how important playtesting is I guess.
But anyway, back on track: How do I plan on getting this done?
Simple. I think I'll draw the current player's token on the mouse following it, so they know what's up. How do I plan on doing this? In the draw() function within the (inPlay) conditional statement I can draw the token using the image function and have it follow the mouse cursor's coordinates.

Solution

```
void draw() {
  background(255);
  if (inMenu) {
    displayMenu();
  }
  if (inPlay) {
    GM.displayColorGrid();
    GM.displayPlayingGrid();
    image(tokenList[TM.turn-1],mouseX, mouseY);
  }
}
```



I cannot screenshot my mouse so I drew it in red to visualize what's happening.

Three-Tac-Toe

Implementation Log - Entry #6

Ravindu Hewabaddage
01-12-2024

Challenge

How to draw a strike over time (animate)

Brainstorming

So I ended up not implementing the score system even with so much work done planning it. Why? Because I felt that getting 3 in a row and winning is just a lot more intuitive and possibly fun? So now I want to do an animation where it doesn't abruptly end the moment you finish 6 rounds or get 3 in a row, it instead slowly draws a strike across over time and then transitions to the game over screen. (Strike animation could be accelerated possibly to tick the physics requirement?) I could use frameCount%60 and use map to bring it out to the full length? like line (x1, y1, x2 * (map((frameCount%60),0,59,0,1), y2 * (map((frameCount%60),0,59,0,1)) Which takes the modulo from 0,59 and maps it from 0 to 1 which is basically taking the value of x2 & y2 from 0 to full value over a second as the framerate is 60. and then an if condition to stop it after reaching 59 but I have a feeling there's a better way to do it.

x2* (constrain(map((frameCount%120), 0, 59, 0, 1),0,1) this should constrain the values from 0 to 1 where as without the constrain function it would go from 0 to 2. This eliminates the need to have an if function that takes us out of the counting and let it run for a whole second longer while being present. Then when we reach 119 we can use an if statement to take us to the gameover screen so it doesn't reset the strike.

Solution

Score Manager checks for strikes every mouseclick.
When we find a mouse click, it triggers stroke drawing animation and disables clicking (boolean required for enabling clicking while inPlay).
Increase round timer to 27 and change grid to 9x9 so as to avoid softlocks where players don't have any tiles to click on to end the game or the game prematurely ends with a few tiles left empty.
after 119 frames it transitions to game over screen which shows winner. If no winner it shows game over screen without a winner. With winner: token around cursor with crown bobbing up and down above it.

How I ended up implementing it

```
void drawStrike(){
  line(startCoords.x, startCoords.y, endCoords.x*constrain(map(((frameCount-getFrameCount)%180),0,120,0,1),0,1), endCoords.y*constrain(map(((frameCount-getFrameCount)%180),0,120,0,1),0,1));
  if ((frameCount-getFrameCount)%180 == 179){
    inPlay = false;
    gameOver = false;
  }
}
```

Did this looooong line of code to draw an animating line work? No ofcourse not, that would be waaaaay too easy. What ended up happening was it was drawing from the origin to the designated location. Not how I expected it to work. Trying to figure out how to fix this I thought I might have to do this line but 4 different times for each case but that terrifying thought was swept aside immediately. I'm a game designer, I can figure out the laziest, stupidest way to implement this. It's literally going to be my job.

```
void drawStrike(){
  strokeWeight(5);
  stroke(255,0,0);
  line(startCoords.x, startCoords.y, startCoords.x + diffCoords.x*constrain(map(((frameCount-getFrame)%120),0,59,0,1),0,1), startCoords.y + diffCoords.y*constrain(map(((frameCount-getFrame)%120),0,59,0,1),0,1));
  if ((frameCount-getFrameCount)%120 == 119){
   toPlay = false;
   clickEnabled = true;
   gameOver = true;
  }
}
```

Ended up writing an equally longer line of code but it caused an interesting problem.

```
Strikes(PVector tempStartCoords, PVector tempEndCoords) {
  getFrame = frameCount;
  startCoords = tempStartCoords;
  endCoords = tempEndCoords.copy();
  diffCoords = endCoords.sub(startCoords);
  velocity = new PVector(0,0);
  getFrameCount = frameCount;
}
```

So because I was bringing in the coordinates from the player token and since I was setting it endCoords directly and doing a calculation to change it, it changed the position of the token in game. Which SERIOUSLY confused me but thank god for Kyle Gunter. Asked him for help, he looked through my code and explained how addresses work and told me to add a .copy() so it doesn't change the original object's data.