

## Process & Decision Document

**Name:** Jasmine Henri

**Role(s):** Coder & Reviewer of AI changes

**Primary responsibility for this work:** Work with AI and independently to come up with modifications for the code that will successfully demonstrate specific emotions

**Goal of Work Session:** During this work session, I focused on making changes to the blobs movements and environment to express specific emotions.

**GenAI tools (if used):**

- Chatbot on Visual Studio Code 6

### GenAI Documentation

**Date Used:** January 25th, 2026

**Tool Disclosure:** Chatbot on Visual Studio Code 6

**Purpose of Use:** Add new code lines to the forked code to bring my suggestions and ideas to light.

**Summary of Interaction:** I explained to the chatbot what my ideas were to modify the code to express emotion and it would take my instructions and make edits to the codes, then I was able to see review the added code to determine if I either accept the change or wish to decline it.

**Human Decision Point(s):** I was happy with the specific changes that were made to the code and had to initiate minimal changes.

- Changed the colour of the blob when on platforms to random(255) to make it more fun instead of one colour change (however it caused a flickering which would have triggered light sensitive viewers so I turned to the chatbot to help me fix the flickering)
- Modified the background myself so the colour would be grey when the blob is on the ground/ in the air and light blue when the blob is on a platform

**Integrity & Verification Note:**

All decisions were verified using:

- My prior knowledge of coding from CS105 and CS106
- Knowledge of coding that was reviewed in Class 2 lecture by David

### **Scope of GenAI Use:**

Specifically, GenAI did *not*:

- Create the initial background colour change code (however it did modify the location in draw and slightly edited the coding so that the background wasn't drawn over the ledges)
- Randomize the blobs colour

### **Limitations or Misfires:**

The model demonstrated no limitations and strongly aided in bringing my design changes to life.

### **Summary of Process (Human + Tool)**

Four iterations with GenAI:

- 1: The Chatbot added code to make the platforms turn yellow when the blob is actively standing on it
- 2: The Chatbot added code to make the blob black if it's on the ground but blue if it's on a platform
- 3: The Chatbot corrected the flickering after I had added the random colour code
- 4: The Chatbot corrected and re-organized the code when I added a conditional statement to control the background colour that had interfered with the blob changing colours

### **Decision Points & Trade-offs**

I made individual design decisions to change the blob colour when on a ledge and the background colour change, however this ended up requiring the chatbots help because my changes caused unintentional changes to other aspects of the code.

### **Verification & Judgement**

I evaluated the appropriateness of my changes by:

- Reviewing notes from CS105 and CS106

- Reviewing Class 2 lecture by David
- Reflecting on how flickering colours can negatively impact light sensitive viewers

### Limitations, Dead Ends, or Open Questions

- All issues were resolved successfully by the chatbot

### Appendix - Visual Studio Code 6 Chatbot

#1 Could you make edits to the code to make the platforms turn yellow when the blob is actively standing on them

Now I'll modify the code to track which platform the blob is standing on and draw that platform in yellow. I need to:

1. Add a property to track which platform the blob is on
2. Check which platform the blob is standing on during collision detection
3. Draw platforms with different colors based on whether the blob is on them

Edited

sketch.js+1-0

Edited

sketch.js+6-0

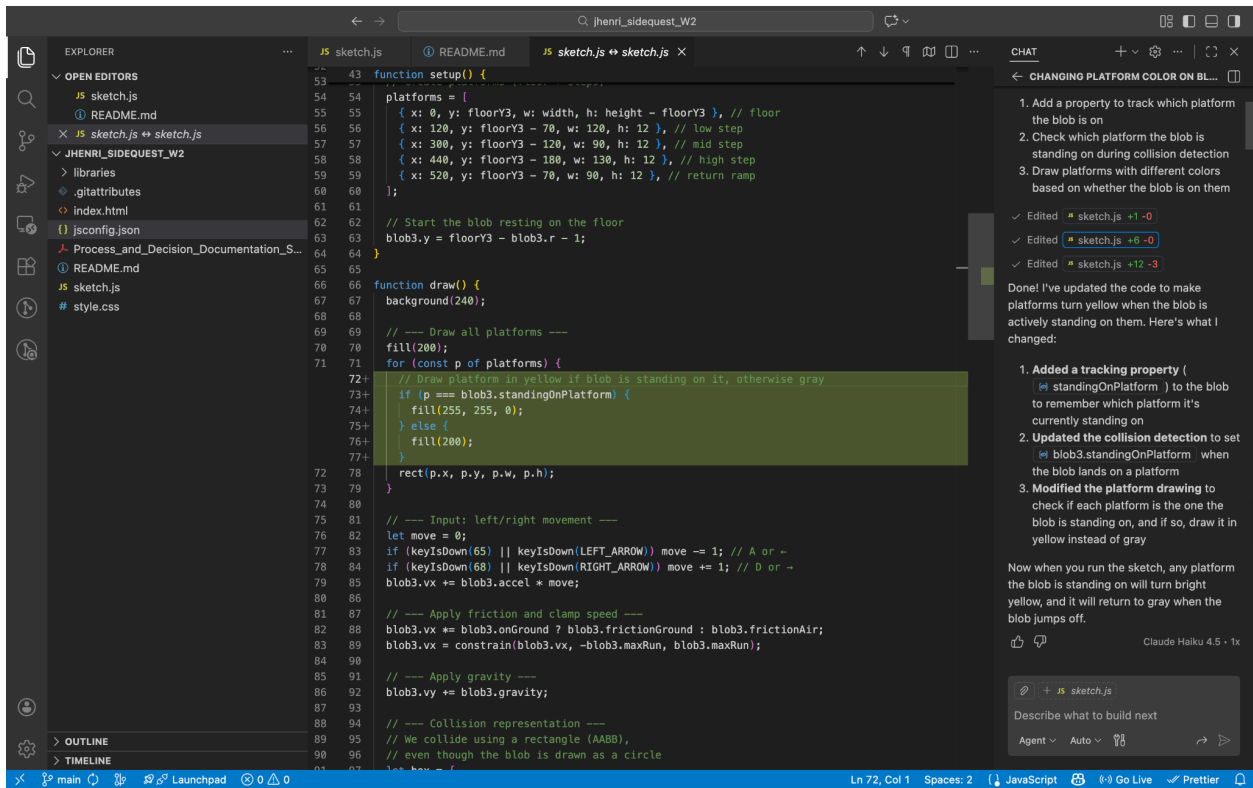
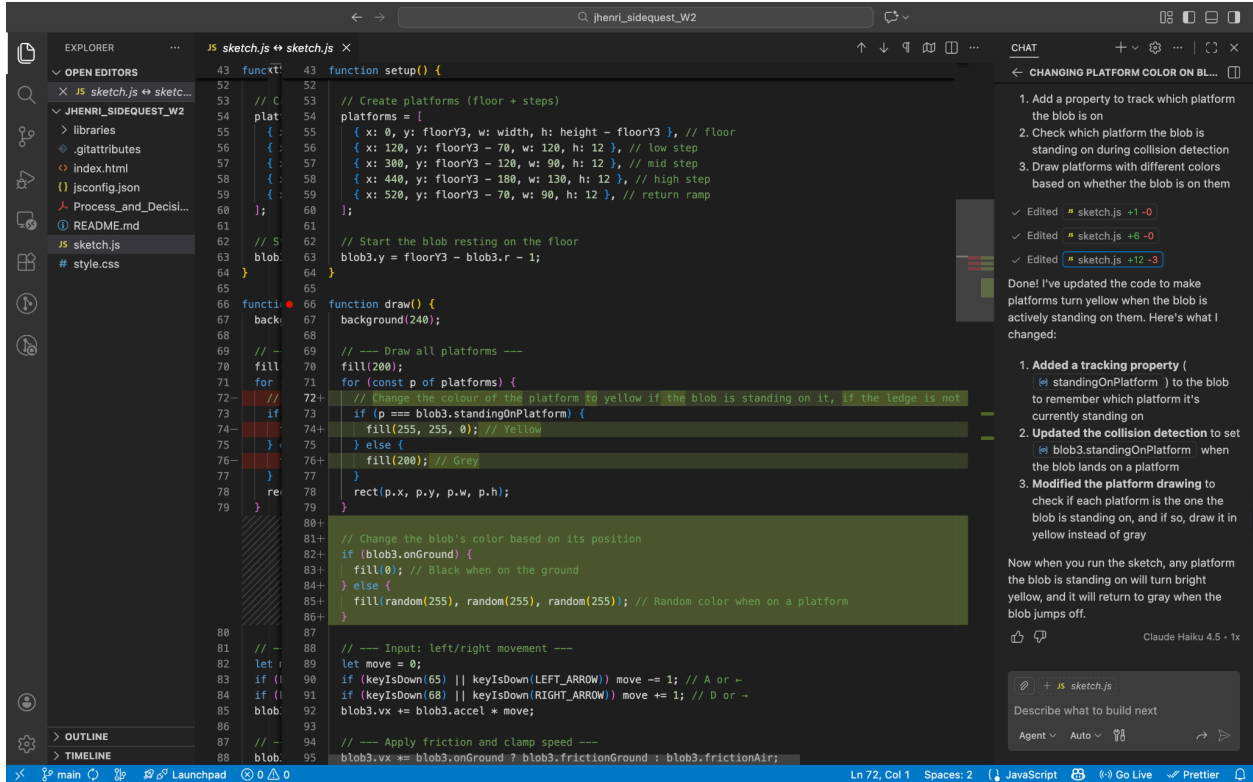
Edited

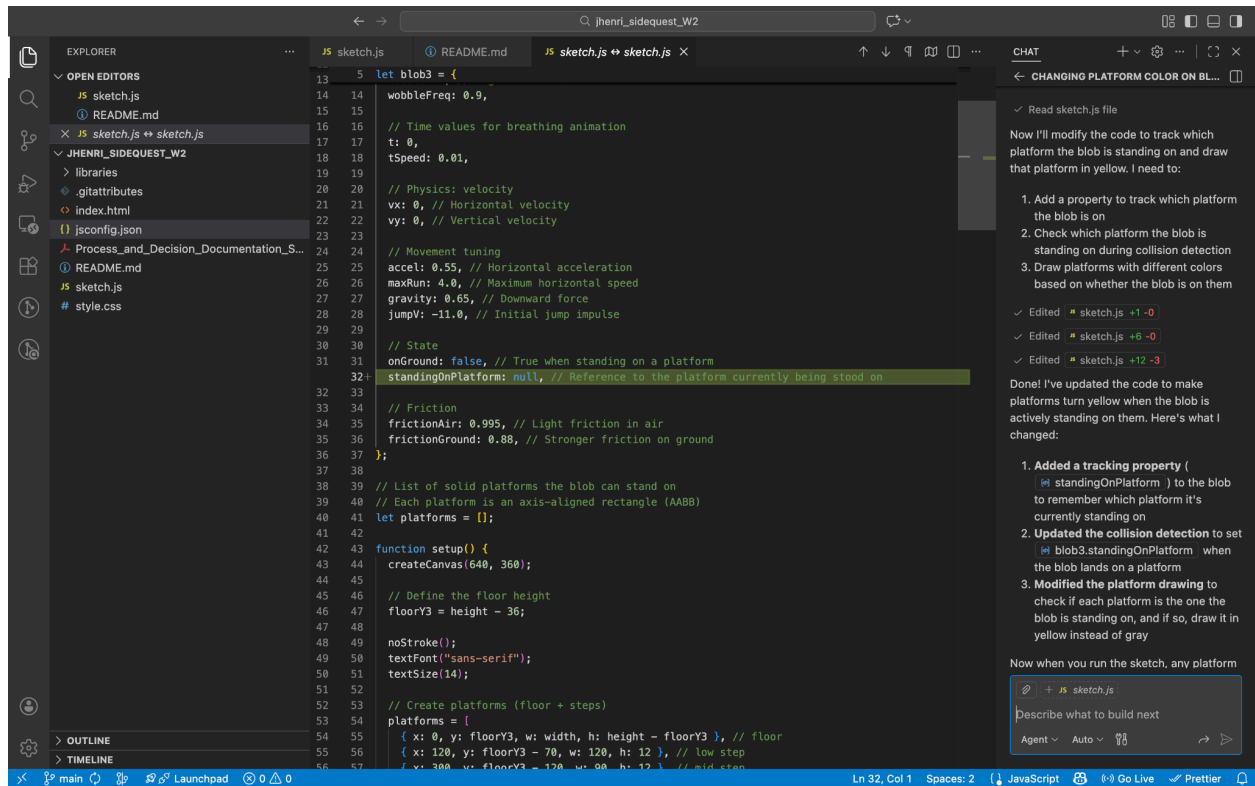
sketch.js+2-0

Done! I've updated the code to make platforms turn yellow when the blob is actively standing on them. Here's what I changed:

1. **Added a tracking property** (standingOnPlatform) to the blob to remember which platform it's currently standing on
2. **Updated the collision detection** to set blob3.standingOnPlatform when the blob lands on a platform
3. **Modified the platform drawing** to check if each platform is the one the blob is standing on, and if so, draw it in yellow instead of gray

Now when you run the sketch, any platform the blob is standing on will turn bright yellow, and it will return to gray when the blob jumps off.





#2: Can you make edits to the code to make the blob black if on the ground but if the blob is on a platform the blob colour is changed to blue

Now I'll add the blob color logic. I need to set the blob color before drawing it. I'll add code after the platforms are drawn to set the color based on whether the blob is on the floor or a platform, then modify the `drawBlobCircle` function to use the pre-set color.

Let me fix the JSON structure:

Edited

sketch.js+9-2

Done! I've updated the code so the blob will now:

- Turn **black** when standing on the floor (the main ground platform)
- Turn **blue** when standing on any of the elevated platforms

The color is set in the `draw()` function based on which platform the blob is currently standing on, and the `drawBlobCircle()` function uses that pre-set color.

```
66 function draw() {
67   background(240);
68
69   // --- Draw all platforms ---
70   fill(200);
71   for (const p of platforms) {
72     // Change the colour of the platform to yellow if the blob is standing on it, if the ledge is not being
73     if (p === blob3.standingOnPlatform) {
74       fill(255, 255, 0); // Yellow
75     } else {
76       fill(200); // Grey
77     }
78     rect(p.x, p.y, p.w, p.h);
79   }
80
81+ // Set blob color based on which platform it's standing on
82+ if (blob3.standingOnPlatform === platforms[0]) {
83+   // Standing on the floor (first platform)
84+   fill(0); // Black
85+ } else if (blob3.standingOnPlatform !== null) {
86+   // Standing on any other platform
87+   fill(random(255), random(255), random(255));
88+ }
89
90 // --- Input: left/right movement ---
91 let move = 0;
```

```
66 function draw() {
149 // --- Convert collision box back to blob centre ---
150 blob3.x = box.x + box.w / 2;
151 blob3.y = box.y + box.h / 2;
152
153 // Keep blob inside the canvas horizontally
154 blob3.x = constrain(blob3.x, blob3.r, width - blob3.r);
155
156 // --- Draw the animated blob ---
157 blob3.t += blob3.tSpeed;
158 drawBlobCircle(blob3);
159
160 // --- HUD ---
161 fill(0);
162 text("Move: A/D or +/- • Jump: Space/W/↑ • Land on platforms", 10, 18);
163 }
164
165 // Axis-Aligned Bounding Box (AABB) overlap test
166 // Returns true if rectangles a and b intersect
167 function overlap(a, b) {
168   return (
169     a.x < b.x + b.w && a.x + a.w > b.x && a.y < b.y + b.h && a.y + a.h > b.y
170   );
171 }
172
173 // Draws the blob using Perlin noise for a soft, breathing effect
174 function drawBlobCircle(b) {
175+   fill(20, 120, 255);
176   beginShape();
```

← CHANGING PLATFORM COLOR ...  
✓ Read lines 1 to 200 of sketch.js

Now I'll add the blob color logic. I need to set the blob color before drawing it. I'll add code after the platforms are drawn to set the color based on whether the blob is on the floor or a platform, then modify the drawBlobCircle function to use the pre-set color.

Let me fix the JSON structure:

✓ Edited `sketch.js +10 -4`

Done! I've updated the code so the blob will now:

- Turn **black** when standing on the floor (the main ground platform)
- Turn **blue** when standing on any of the elevated platforms

The color is set in the `draw()` function based on which platform the blob is currently standing on, and the `drawBlobCircle()` function uses that pre-set color.

← CHANGING PLATFORM COLOR ...  
✓ Read lines 1 to 200 of sketch.js

Now I'll add the blob color logic. I need to set the blob color before drawing it. I'll add code after the platforms are drawn to set the color based on whether the blob is on the floor or a platform, then modify the drawBlobCircle function to use the pre-set color.

Let me fix the JSON structure:

✓ Edited `sketch.js +10 -4`

Done! I've updated the code so the blob will now:

- Turn **black** when standing on the floor (the main ground platform)
- Turn **blue** when standing on any of the elevated platforms

The color is set in the `draw()` function based on which platform the blob is currently standing on, and the `drawBlobCircle()` function uses that pre-set color.

#3 I've set the else/ if statement to be set to a random colour when the blob is on a ledge, however it is causing colour flickering. Could you edit the code so that when the blob lands on a platform it sticks to one random colour but if it jumps to another it switches to a different singular random colour

I can see the issue - random() is being called every frame in the draw loop. I need to:

1. Add a property to store the platform's color when the blob lands on it
2. Only generate a new random color when the blob lands on a *different* platform
3. Use that stored color instead of regenerating it each frame

Edited

sketch.js+7-2

Done! I've fixed the flickering by:

1. **Added a platformColor property** to the blob that stores the color for the current platform
2. **Generate a new random color only once** when the blob first lands on a platform (when platformColor is null)
3. **Reuse that color** every frame while on the same platform
4. **Reset the color** when the blob lands back on the floor

Now when the blob jumps to a different platform, it will get a new random color, but it will stay that color consistently while on the same platform without flickering.

```
5 let blob3 = {  
6   // Physics: velocity  
7   vx: 0, // Horizontal velocity  
8   vy: 0, // Vertical velocity  
9  
10  // Movement tuning  
11  accel: 0.55, // Horizontal acceleration  
12  maxRun: 4.0, // Maximum horizontal speed  
13  gravity: 0.65, // Downward force  
14  jumpV: -11.0, // Initial jump impulse  
15  
16  // State  
17  onGround: false, // True when standing on a platform  
18  standingOnPlatform: null, // Reference to the platform currently being stood on  
19  platformColor: null, // Stores the color for the current platform  
20  
21  // Friction  
22  frictionAir: 0.995, // Light friction in air  
23  frictionGround: 0.88, // Stronger friction on ground  
24  
25  };  
26  
27 // List of solid platforms the blob can stand on  
28 // Each platform is an axis-aligned rectangle (AABB)  
29 let platforms = [];
```

← CHANGING PLATFORM  
being called every frame  
loop. I need to:

1. Add a property to s  
platform's color wh  
lands on it
2. Only generate a ne  
color when the blo  
different platform
3. Use that stored col  
regenerating it eac

✓ Edited sketch.js →

Done! I've fixed the flick

1. Added a platformColor property to the blob to store the color for the current platform
2. Generate a new random color only once when the blob first lands on a platform (when platformColor is null)

```

66
67
68
69
70+ // Set blob color based on which platform it's standing on
71+ if (blob3.standingOnPlatform === platforms[0]) {
72+   // Standing on the floor (first platform)
73+   background(240); // Grey background
74+   fill(0); // Black
75+   blob3.platformColor = null;
76+ } else if (blob3.standingOnPlatform !== null) {
77+   // Generate a new random color only when landing on a new platform
78+   if (blob3.platformColor === null) {
79+     blob3.platformColor = color(random(255), random(255), random(255));
80+   }
81+   background(220, 240, 255); // Light blue background
82+   fill(blob3.platformColor);
83+ }
84
85 // --- Draw all platforms ---
86 fill(200);
87 for (const p of platforms) {
88   // Change the colour of the platform to yellow if the blob is standing on it, if the ledge is not being
89   if (p === blob3.standingOnPlatform) {
90     fill(255, 255, 0); // Yellow
91   } else {
92     fill(200); // Grey
93   }
94   rect(p.x, p.y, p.w, p.h);
95 }

```

```

- // Set blob color based on which platform it's standing on
- if (blob3.standingOnPlatform === platforms[0]) {
-   // Standing on the floor (first platform)
-   fill(0); // Black
- } else if (blob3.standingOnPlatform !== null) {
-   // Standing on any other platform
-   fill(random(255), random(255), random(255));
- }

```

### CHANGING PLATFORM COLOR ...

being called every frame in the draw loop. I need to:

1. Add a property to store the platform's color when the blob lands on it
2. Only generate a new random color when the blob lands on a *different* platform
3. Use that stored color instead of regenerating it each frame

✓ Edited sketch.js +16 -9

Done! I've fixed the flickering by:

1. Added a `platformColor` property to the blob that stores the color for the current platform
2. Generate a new random color only once when the blob first lands on a platform (when `platformColor` is null)
3. Reuse that color every frame while on the same platform
4. Reset the color when the blob lands back on the floor

Now when the blob jumps to a different platform, it will get a new random color, but it will stay that color consistently while on the same

+ JS sketch.js

Describe what to build next

#### #4 Now that i've added some other changes the blob colour change isnt happening

I see the issue - the color logic is at the start of draw() before blob3.standingOnPlatform has been set by collision detection. I need to move it to after the collision detection. Let me read more to see where collision detection ends:

Read sketch.js, lines 140 to 175

Now I'll move the color-setting logic from the beginning of draw() to right before drawBlobCircle() is called, after collision detection:

Edited

sketch.js+20-15

Done! I've moved the blob color logic to after collision detection. The issue was that the code was checking blob3.standingOnPlatform at the beginning of the draw loop before it was actually set by the collision detection. Now it will:

- Turn **black** when standing on the floor
- Turn to a **random color** when landing on a platform
- Maintain that color while jumping



- Return to black when landing back on the floor

```

4 function setup() {
5   // Create platforms (floor + steps)
6   platforms = [
7     { x: 0, y: floorY3, w: width, h: height - floorY3 }, // floor
8     { x: 120, y: floorY3 - 70, w: 120, h: 12 }, // low step
9     { x: 300, y: floorY3 - 120, w: 90, h: 12 }, // mid step
10    { x: 440, y: floorY3 - 180, w: 130, h: 12 }, // high step
11    { x: 520, y: floorY3 - 70, w: 90, h: 12 }, // return ramp
12  ];
13
14  // Start the blob resting on the floor
15  blob3.y = floorY3 - blob3.r - 1;
16  }
17
18 function draw() {
19   background(240);
20
21   // Set blob color based on which platform it's standing on
22   if (blob3.standingOnPlatform === platforms[0]) {
23     // Standing on the floor (first platform)
24     background(240); // Grey background
25     fill(0); // Black
26     blob3.platformColor = null;
27   } else if (blob3.standingOnPlatform !== null) {
28     // Generate a new random color only when landing on a new platform
29     if (blob3.platformColor === null) {
30       blob3.platformColor = color(random(255), random(255), random(255));
31     }
32     background(220, 240, 255); // Light blue background
33     fill(blob3.platformColor);
34   }
35 }
36
37 function draw() {
38   // --- Convert collision box back to blob centre ---
39   blob3.x = box.x + box.w / 2;
40   blob3.y = box.y + box.h / 2;
41
42   // Keep blob inside the canvas horizontally
43   blob3.x = constrain(blob3.x, blob3.r, width - blob3.r);
44
45   // Set blob color based on which platform it's standing on (after collision detection)
46   if (blob3.standingOnPlatform === platforms[0]) {
47     // Standing on the floor (first platform)
48     fill(0); // Black
49     blob3.platformColor = null;
50   } else if (blob3.standingOnPlatform !== null) {
51     // Generate a new random color only when landing on a new platform
52     if (blob3.platformColor === null) {
53       blob3.platformColor = color(random(255), random(255), random(255));
54     }
55     fill(blob3.platformColor);
56   } else {
57     // In the air - use the last platform color or default
58     if (blob3.platformColor !== null) {
59       fill(blob3.platformColor);
60     } else {
61       fill(20, 120, 255); // Default blue
62     }
63   }
64 }
65
66
67

```

### ← CHANGING PLATFORM COLOR ...

Read [sketch.js](#), lines 140 to 179

I see the issue - the color logic is at the start of draw() before `blob3.standingOnPlatform` has been set by collision detection. I need to move it to after the collision detection. Let me read more to see where collision detection ends:

Read [sketch.js](#), lines 140 to 179

Now I'll move the color-setting logic from the beginning of draw() to right before `drawBlobCircle()` is called after collision detection:

Edited [sketch.js](#) +20 -15

Done! I've moved the blob color logic to after collision detection. The issue was that the code was checking `blob3.standingOnPlatform` at the beginning of the draw loop before it was actually set by the collision detection. Now it will:

- Turn **black** when standing on the floor
- Turn to a **random color** when landing on a platform
- Maintain that color
- Return to black when landing back on the floor