HW 6 - Journal

Game Concept

My inspiration for the initial game was Snake. While programming snake, I thought to myself why not make this game 2 player game where each player competes to have the other player touch their tail. But then I realized that was just TRON. So I went with TRON. However, one big difference my game has from TRON is that the tail not exactly follows the players previous position. The tail is divided into little segments and each segment follows the one in front of itself.

Controls

Player 1 - Direction - Player 2

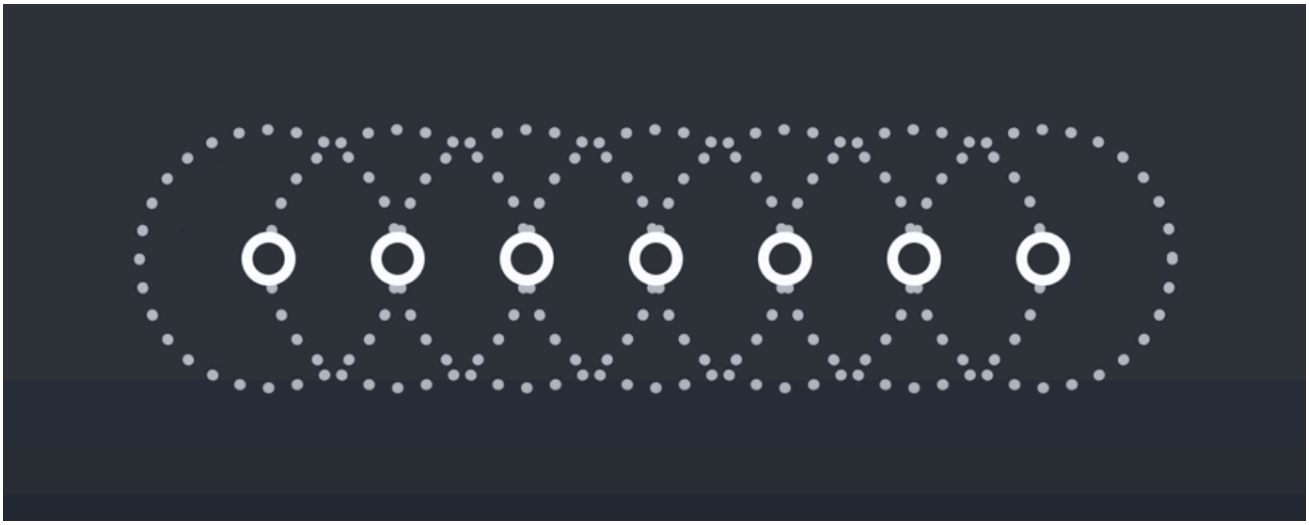
- W - Forward - Up arrow
- A - Left - Left arrow
- D - Right - Right arrow

Generative Rules

The tail

First, let's talk about the tail. I was looking at a procedural animation techniques and found something called "Chaining distance constraints". The tail is made out of these little segments.

- Each segment has a head it follows.
- Every time the head moves, the segment rotates to the head's angle and moves in.



The player

The player has 3 directions they can go.

- Forward
- Left
- Right Once the player changes their direction, they will keep going to that direction until they change to another direction. I decided to make it so because one player can just hold down a button and not let the other player play.

Gameplay objectives

Both players spawn in an arena and has 5 lives each. The objective is to let the other player touch the player's tail and diminish the opponent's health. If any player's health reaches 0, the other player will win the game.

Core mechanics and loop

Each round the player's position is reset. However, their objective is still the same - beat their opponent. I think the looping mechanic in this game is straightforward.

Anticipated player experience

I hope the players will project themselves on the avatars and have fun playing this game. When the stakes are high (like having 1 health), players will have to give their absolute focus, going into flow state.

Source Code

```
import java.util.*;
PFont font;
float arenaX;
float arenaY;
float arenaW;
float arenaH;

float rotateRight = 0.01;
float rotateLeft = -0.01;
int round = 0;
int enemyCount;
int allyCount;
Player p1;
Player p2;
```

```
int countdownTime = 3;
int startTime;
boolean roundStarted = false;

float healthMenuY;
float healthMenuX;

void setup(){
  size(1080, 1080);
  background(0);
  frameRate(60);
  arenaX = width/8;
  arenaY = height/8-100;
  arenaW = width/8*6;
  arenaH = height/8*6-100;
  healthMenuY = height/8*6;
  healthMenuX = width/10*3;

  font = createFont("dogica.ttf", 32);
  textFont(font);

  p1 = new Player(arenaX + arenaW/3, height/2, true);
  p2 = new Player(arenaX + arenaW/3*2, height/2, false);
  startTime = millis();
}

void keyPressed(){
  if (key ==  'a' || key == 'w' || key == 'd'){
    p1.motion = key;
  }
  if (key ==  'h' || key == 'k' || key == 'l'){
    p2.motion = key;
  }
}

void draw(){
  // Count Down
  background(0);
  if (!roundStarted){
    int elapsedTime = (millis() - startTime) / 1000;  // Convert to seconds
    int remainingTime = countdownTime - elapsedTime;  // Countdown

    if (remainingTime > 0) {
      // Show countdown on screen
      fill(255);
```

```
      textSize(50);
      textAlign(CENTER, CENTER);
      text(remainingTime, width / 2, height / 2);
      return;
    } else { roundStarted = true; }
  }
  // Game Over Menu
  if (p1.health == 0 || p2.health == 0){
    textSize(40);
    if (p1.health == 0){
      textAlign(CENTER);
      fill(p2.c);
      text("PLAYER 2 WINS!", width/2, height/2);
    } else {
      textAlign(CENTER);
      fill(p1.c);
      text("PLAYER 1 WINS!", width/2, height/2);
    }
    textSize(20);
    text("Press R to restart!", width/2, height/3*2);
    if (keyPressed){
      if (key == 'r'){
      p1 = new Player(arenaX + arenaW/3, height/2, true);
      p2 = new Player(arenaX + arenaW/3*2, height/2, false);
      roundStarted = false;
      startTime = millis();
      }
    }
  // Main Game Play
  } else {
    stroke(255);
    noFill();
    rectMode(CORNER);
    rect(arenaX, arenaY, arenaW, arenaH);
    pushMatrix();
      translate(healthMenuX, healthMenuY);
      textAlign(RIGHT);
      textSize(40);
      fill(p1.c);
      text("P1:", 40, 40);
      noStroke();
      for (int i=0; i<p1.health; i++){
        rect(50 + i*50, 0, 30, 60);
      }
      translate(0, 90);
      fill(p2.c);
```

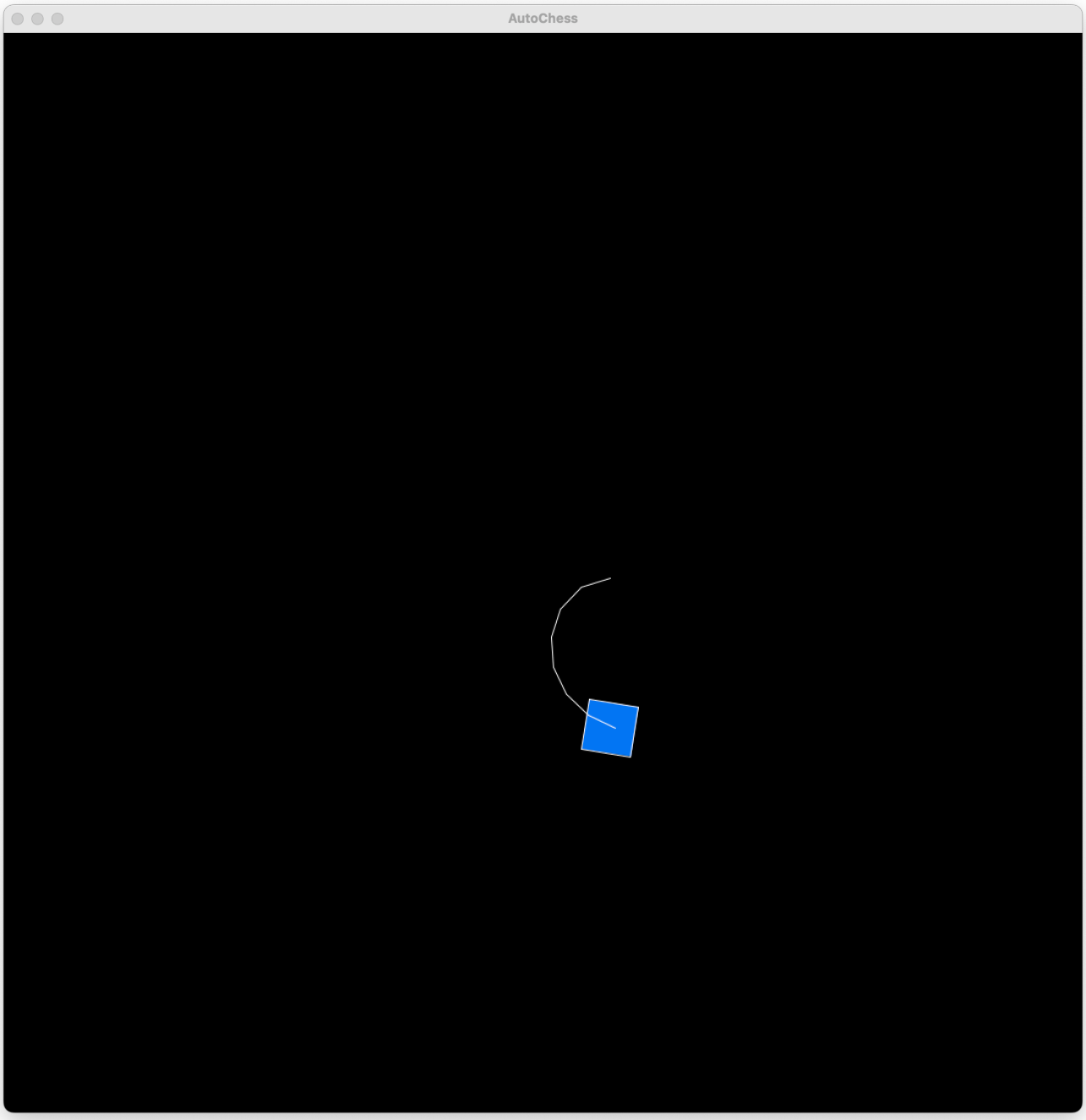
```
      text("P2:", 40, 40);
      for (int i=0; i<p2.health; i++){
        rect(50 + i*50, 0, 30, 60);
      }
    popMatrix();
    p1.draw();
    p2.draw();
    p1.checkTouchOther(p2);
    p2.checkTouchOther(p1);
  }
}

void resetRound(){
  roundStarted = false;
  startTime = millis();
  p1.playerReset();
  p2.playerReset();
}
```

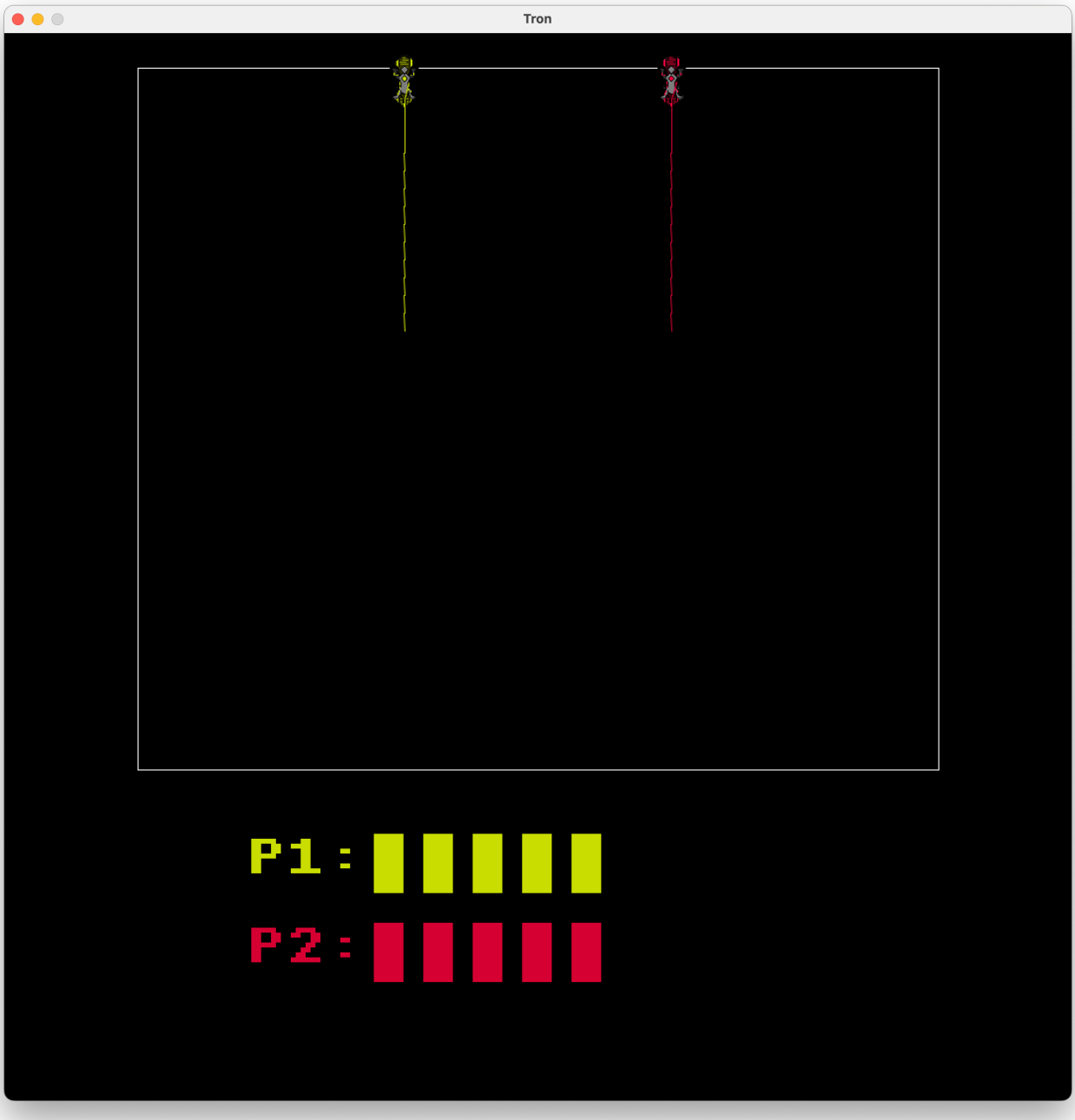
Other relative codes are in the game folder.

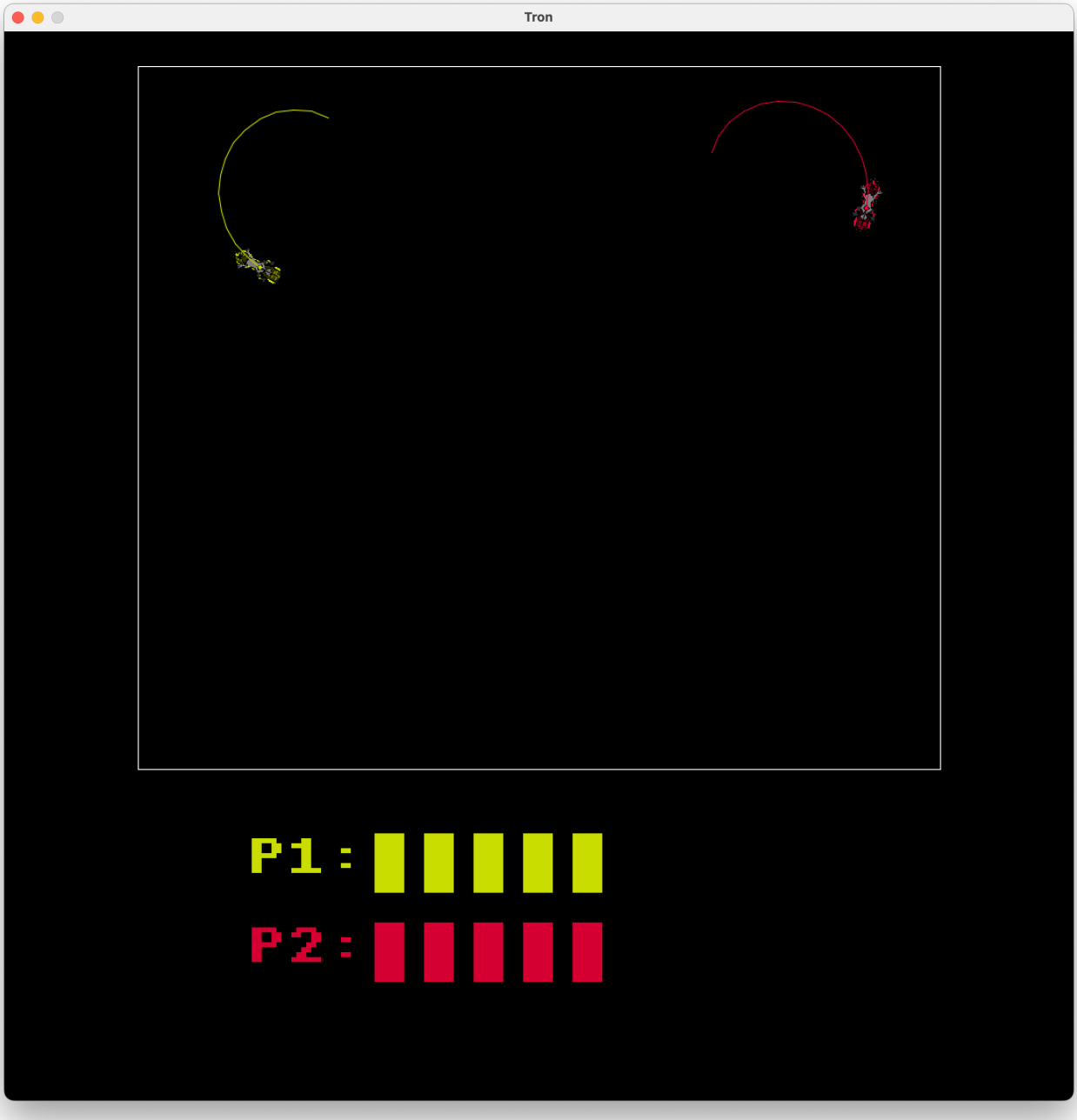
Outcome

Initial concept



Final Product


Tron
P1:
P2:


Tron
P1:
P2: