Build Your Own World

Java

```
while (StdDraw.hasNextKeyTyped()) {
    char key = StdDraw.nextKeyTyped();
    if (key == 'w' || key == 'W') {
        moveUp();
    } else if (key == 'd' || key == 'D') {
        moveRight();
    } else if (key == 'a' || key == 'A') {
        moveLeft();
    } else if (key == 's' || key == 'S') {
        moveDown();
    } else if (key == ':') {
        qChecker();
    }
    else if (key == 'b' || key == 'B') {
        this.showSmall = !(this.showSmall) ;
    }
```

Goal:

Create a video game controlled by keys AWSD that allows players to move around a world full of hallways and rooms. Each world is unique to the seed input and customized by seasonal choices that determine the world's appearance; the player has control of over all aspects of the highly functional game.

Steps and Challenges:

The brainstorming phase was crucial, though it took more time than I anticipated. My partner and I started with the basics—getting the character to move using the keyboard. From there, we began to build hallways around the board, ensuring that each element functioned correctly before moving on. Our strategy was to assemble the project piece by piece, a method that not only made sense but also simplified the debugging process. However, as the project grew, so did the complexity. Keeping track of all the moving parts became increasingly difficult. Despite our best efforts to stay organized, we encountered bugs at nearly every junction. Each one required careful attention and a lot of patience to resolve.

```
public void generateHallway() {
    for (Integer[] flower1: flowerCoords) {
        for (Integer[] flower2: flowerCoords) {
            if (!(flower1.equals(flower2))) {
                double xS = Math.pow(flower1[0] - flower2[0], 2);
                double yS = Math.pow(flower1[1] - flower2[1], 2);
                double dist = Math.sqrt(xS + yS);
                if (!distToFlowers.containsKey(dist)) {
                    ArrayList<Integer[]> twoFlowers = new ArrayLi
                    twoFlowers.add(flower1);
                    twoFlowers.add(flower2);
                    distToFlowers.put(dist, twoFlowers);
                }
```

Takeaways:

- Brainstorming and planning is critical.
  - Before taking on any large-scale coding task, planning out and pre-emptively catching bugs is everything. It is MUCH better in the long term to think at a high level instead of quickly coding and throwing it away.
- Experience with GUI.
  - Having such an interactive project was different from my other data-oriented pieces of work. Utilizing STDdraw and other packages that allow the user to use their mouse and keyboard was helpful, even in the creation of this very online portfolio.
- The benefits of bite-size code.
  - With so many moving pieces working with each other, I truly experienced and learned the benefits of clean-cut code. Having easy-to-understand variable names, helper functions, and clear comments made all the difference in the last few days when putting everything together.
- Teamwork makes the dream work!
  - Working with a partner can pose its own challenges, but with clear communication and enough patience, the privlegde to work with another like-minded indivudal and create amazing projects cannot be overlooked. With each of us bringing our own unique techniques, we were able to work through problems thoughtfully to create the finished product.

```
public int getRandomX(Node n) { return n.randomX; }

5 usages  Parth Desai
public int getRandomY(Node n) { return n.randomY; }

3 usages  Parth Desai
public int getRoomHeight(Node n) { return n.roomHeight; }

3 usages  Parth Desai
public int getRoomWidth(Node n) { return n.roomWidth; }
5 usages  Parth Desai
public int getXWithin(Node n) { return n.xWithin; }
```

```
5 usages  Parth Desai
public int getXWithin(Node n) { return n.xWithin; }

5 usages  Parth Desai
public int getYWithin(Node n) { return n.yWithin; }

2 usages  Parth Desai
public int getRandomNumRooms() { return randomNumRooms; }

1 usage  Parth Desai
public int getSeedCalls() {
    return seedCalls;
}
```

Given this is a class project, I cannot put my entire code online. However, I would be more than ready to discuss the code by request.