

DoodleJump Design Section

Mini-Assignment

Due Date: emailed to both your section TAs by **Monday, October 21 at 11:59 PM.**

Instructions

- **Read the assignment handout carefully** and play around with the DoodleJump demo before you delve into this mini-assignment. We strongly recommend completing this mini-assignment before you begin coding.
- There are three sections to this mini-assignment: questions you can discuss with others, questions for you to do by yourself, and Responsible CS questions at the end.
- Bring a copy of your answers (printed or otherwise) to your design section so your design section TAs can facilitate a productive and collaborative conversation.
- **Note:** This mini-assignment will prove to be **very** helpful for the DoodleJump project. This is your first large project, and taking the time to really think about these questions will be extremely beneficial!!!

Collaborative Questions—think with a group!

Come to conceptual hours to meet other students to discuss these questions with :)

1. Draw a containment diagram for your proposed DoodleJump design. If your design needs it, also draw an inheritance diagram. Remember that the two should not be connected to each other. The handout is a good place to check to start thinking about your design and what objects you will need.
2. The game will need to have many platforms on the screen at once. How will you keep track of these platforms? What data structure can you use here?
3. As mentioned in the handout, the x and y coordinates for the platforms need to be somewhat randomly generated. If the coordinates are completely randomly generated, however, there is a chance that they will be placed too close or too far apart for the game to be playable. What constraints should you place on the coordinates for each new platform? (For example, would you want the platforms to be able to generate off-screen?)

A Quick Pseudocode Crash Course (more questions below)

Pseudocode is “code” written for humans to understand, rather than a compiler. Pseudocode is a way of describing a step-by-step process and is **not** language specific.

For example, let’s say we want to describe the process to make a latte. We want to try to make the description clear enough that it can be translated into code in any language!

Let’s make a latte:

```
method makeLatte(cupSize, milk):
    cup = Cup of size cupSize
    turn on machine
    while cup is not half full:
        add espresso to cup
    have machine froth the milk
    while cup is not full:
        add milk to cup
```

We come across this block of pseudocode and now we want to write it as a Java method. We see that we’ll have to use an instance of a machine. Let’s call it `_machine`.

```
public void makeLatte(int cupSize, Milk milk){
    Cup cup = new Cup(cupSize);
    _machine.turnOn();

    while (!cup.isHalfFull()){
        cup.addEspresso();
    }

    _machine.frothMilk(milk);

    while (!cup.isFull()) {
        cup.add(milk);
    }
}
```

The pseudocode should be specific, with roughly a line of pseudocode for every line of real code. Note that punctuations (semicolons, brackets) are omitted in pseudocode as it is language independent.

Important Note: The following DoodleJump pseudocode questions below are **not** collaborative.

Independent Questions—answer by yourself!

Assume you have a variable `_doodle` that is your instance of a `Doodle`, and a variable `_platforms` which is your instance of an `ArrayList` of `Platforms`.

As detailed in the handout, at every timestep, you'll need to:

- Update the Doodle's velocity
- Depending on the Doodle's position on the screen, either:
 - Update the Doodle's position, or...
 - Move all of your Platforms down
- Check for collisions between the Doodle and your Platforms

The following questions are designed to help you figure out how to implement these steps.

4. Write pseudocode for how you will update your `_doodle`'s position and velocity at each timestep. Use the variable `_doodle` if necessary. *Remember that your Doodle should always fall under the effect of gravity.* (You don't need to take into account keyboard input, collisions or vertically scrolling the platforms yet.). Play with the demo to get a feel for what is happening at timesteps, and make sure to read over the handout closely, as it covers much of this material.
5. Write pseudocode for a `bounce()` method that checks if the Doodle is colliding with any of the platforms in `_platforms`, and if it is, sets the rebound velocity. Keep in mind that the direction the Doodle is moving influences whether or not your Doodle should bounce off of a platform, so your function should also account for that. Try to think of and account for all of these kinds of cases.
Hint: Make use of `javafx.scene.shape.Shape.intersects(...)` method, described in the handout. You can also look it up in the javadocs to read more.
6. When your Doodle reaches the vertical midpoint of the screen, you will need to halt its upward movement and instead move the platforms down. Write pseudocode for a method `scrollPlatforms()` that will check the Doodle's position and when

appropriate, go through all of the platforms that exist and adjust their position appropriately to make them “scroll down”.

Some things to think about: When should you check for this condition? How much should your platforms actually move when this happens? Play around with the demo to get a feel for how this vertical scrolling works!

CS Responsibility

As we move forward onto bigger, more exciting projects, we can take this opportunity to step back and evaluate how the software we create can impact people.

When software developers create predictive models, there are bound to be some errors in the models. Algorithmic bias refers to the systematic errors in a system that create unfavorable outcomes that target certain groups. Read the following articles about examples cases of algorithmic bias, and how Google has attempted to deal with it.

First Article:

<https://www.newscientist.com/article/2166207-discriminating-algorithms-5-times-ai-showed-prejudice/>

Second Article:

<https://www.theguardian.com/technology/2019/oct/03/google-data-harvesting-facial-recognition-people-of-color>

After reading the articles, answer the following questions:

7. Why do machine learning models generate unintended bias?
8. What is another example of bias that could occur in a machine learning model?