Forest Scene project

Brief outline of what it does

This project is a simulation of falling leaves in a forest. It creates a visual representation of leaves falling from the trees and being influenced by wind.

It uses the Processing framework to create a visual representation of leaves falling from trees and being influenced by wind. The code initializes a set number of leaves with random positions, sizes, and angles. The leaves are updated and displayed in the `draw()` function, with wind effects applied when the mouse is clicked. When a leaf reaches the ground, it stops falling. If a leaf goes off the screen, it is repositioned with new random values.

Overall, the code creates a visually pleasing animation of falling leaves in a forest scene, with the leaves responding to mouse clicks and wind effects.

Brief outline of the code

- The code initializes a canvas and loads leaf images and a forest background image.
- An array is created to store leaf images.
- The `Leaf` class is defined with properties for position, velocity, acceleration, wind, size, angle, isGrounded, and imageIndex.
- The `update()` method of the `Leaf` class updates the leaf's position based on velocity and applies gravity.
- The `mouseClicked()` function triggers a strong wind effect by setting a random wind direction for each leaf.
- The `draw()` function displays the forest background image and updates and displays each leaf by calling the `update()` and `display()` methods.
- If a leaf touches the ground, it stops falling and sets the `isGrounded` flag to true.
- If a leaf goes off the visible screen, its properties are reset, and it is repositioned with new random values.
- The `display()` method of the `Leaf` class rotates and displays the leaf image based on the leaf's position and angle.

Notes on the design decisions

As part of my project, I devoted the utmost attention to the aesthetic aspects, which led to a creative concept involving the independent creation of designs for leaves, trees, lakes, and birds. The new game, The Legend of Zelda: Tears of the Kingdom, released last month, served as inspiration for me in creating these designs for my project. However, no aspect was left unattended, and special emphasis was placed on the color palette, which I personally selected to perfectly convey the autumn atmosphere. For me, autumn is associated with a variety of warm and vibrant hues.

First and foremost, the colour orange, closely linked to maple leaves, is one of the key elements of my palette. The second important shade is red, encompassing dark red, burgundy, and raspberry nuances, typical of autumn leaves from trees such as maple, oak, and ash. Yellow, a bright and warm colour, also plays a crucial role in my palette, characterizing the autumn leaves of various tree species, including birch, aspen, and ginkgo. Brown tones, ranging from light brown to deep chocolate, are also widely prevalent among autumn leaves, particularly in oak and maple trees. Finally, the golden colour imparts warmth and a noble appearance to certain autumn leaves.

As a result of my work on the project, my color palette has acquired high aesthetic value, embodying not only my inspirations from the game but also representing the diversity of hues characteristic of autumn.

Notes on how I have decided to make a wind simulation interesting for future users

In my code, I wanted to explore the design decision behind simulating wind to create a more engaging and captivating experience for the user of my project.

- ❏ Utilising Vector Operations:

To simulate wind, I manipulate leaf velocity using vector operations, creating gusts.

- ❏ Random Wind Direction:

Clicking the mouse generates a strong gust with a random direction. A random 2D unit vector, multiplied by a factor (0.5), determines gust strength.

- ❏ Adding Additional Effects:

To enhance realism, I introduce variations in leaf speed or direction based on the wind force, creating a dynamic scene.

- ❏ Grounded State:

Leaves have a grounded state when they reach the ground. They settle by adjusting position and setting vertical velocity to zero.

- ❏ Dynamic Leaf Properties:

Leaves have random properties (size, angle, image index) for diversity. They initialise with unique attributes and reset when off-screen, ensuring a changing display.

Explanation for the user on how to run the project

- ❏ When you run the simulation, a virtual forest scene appears on your screen.
- ❏ The scene is populated with several leaves, each represented by a unique image.
- ❏ As time passes, the leaves gently float down, simulating their descent from the trees.
- ❏ The leaves are affected by gravity, giving them a natural and realistic motion.
- ❏ If you want to add some excitement to the scene, simply click your mouse.
- ❏ Your click generates a powerful gust of wind, causing the leaves to sway and change their trajectory.
- ❏ The strength of the wind can be adjusted to your preference.
- ❏ Watch as the leaves elegantly respond to the changing winds, creating a mesmerizing dance in the forest.

Customisation for each user

What can you change in the code?

- You can personalize your experience by replacing the leaf images with your own designs.
- If you wish to adjust the size of the canvas, you have the flexibility to do so.
- Feel free to explore and experiment with different parameters to create unique effects.
- While the project is designed to simulate falling leaves, you can use your creativity to add new elements or modify existing ones.