

StormUI

October 13, 2024

1 Sprint 3: Unreal Engine UI Setup and Real-Time Data Updates

Owen Kroeger:

Video Link: [Sprint 3](#)

Jira Link: [Supercell Simulator](#)

1.1 Objective:

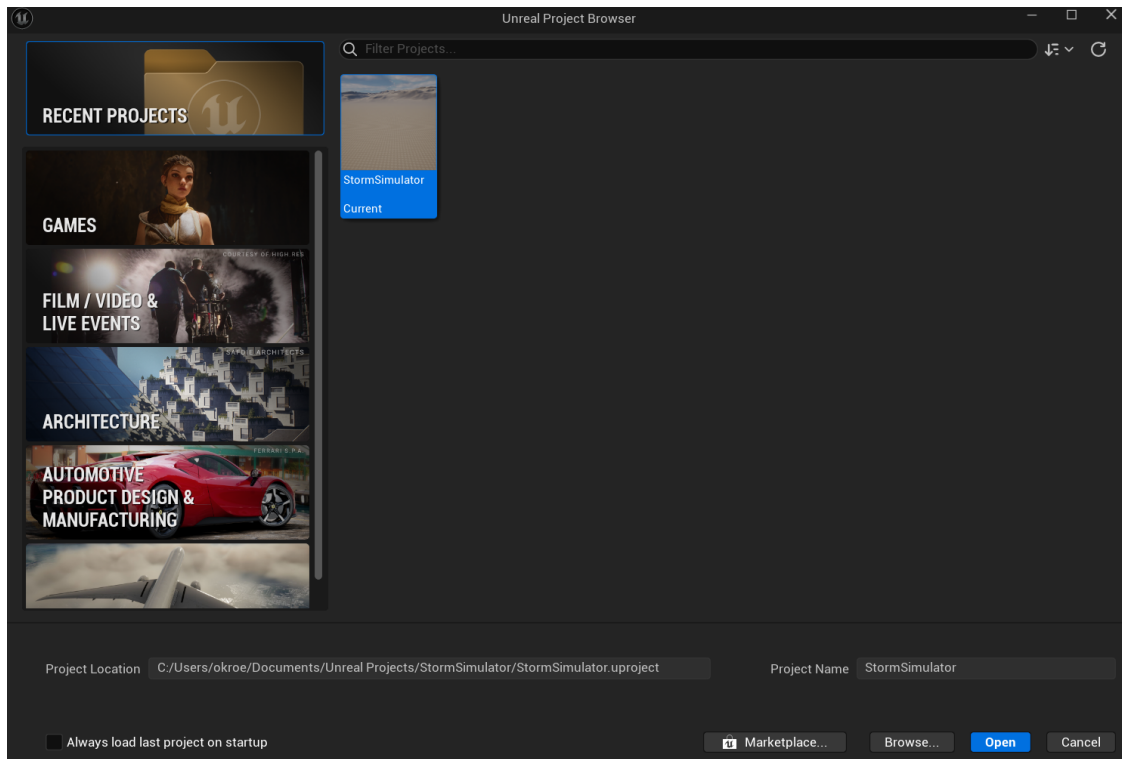
- **Learn Unreal Engine:** Familiarize with Unreal Engine's interface and Blueprint system.
- **UI Setup:** Set up a user interface (UI) with two panels—one for user input via sliders and one for displaying real-time data output.
- **Real-Time Updates:** Implement real-time updates where changes in slider values dynamically update the corresponding data display.

1.2 Learning Unreal Engine:

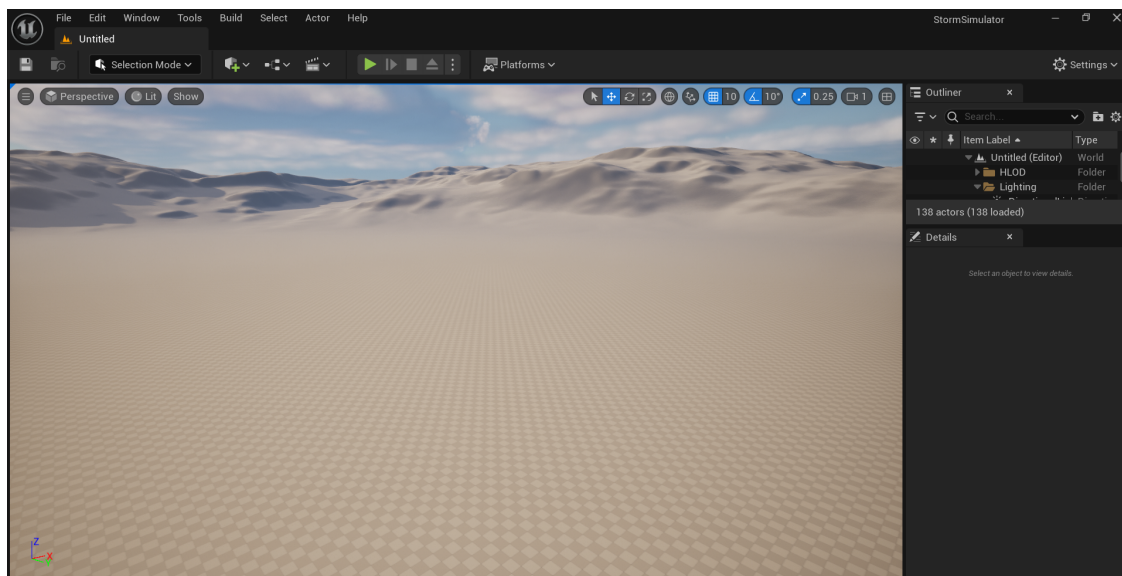
In this sprint, I began by getting comfortable with Unreal Engine, particularly focusing on the UI creation process and learning how to use Blueprints for real-time interactivity. This involved learning how to navigate the Unreal Editor, understanding the various UI elements available, and how to integrate them into my project.

1.2.1 Key Steps:

Setting Up Unreal Engine: Downloaded and installed Unreal Engine via the Epic Games Launcher.



Creating a New Project: Chose the Blank template to start with minimal assets. This was ideal for focusing on the UI development.



Understanding the Unreal Editor Layout: Learned how the various panels work, including the Content Browser, Viewport, and Details Panel.

Introduction to Blueprints: Used Blueprints for visual scripting, which enabled me to create interactions without writing C++ code. This was especially useful for linking UI elements like sliders to real-time data updates.

1.2.2 Key Takeaways:

Blueprint Visual Scripting was essential for designing the UI interactions. The Widget Blueprint is the core tool for creating UIs, where you can layout and program widgets like sliders, text boxes, and buttons.

1.3 Setting Up the UI:

The next step was setting up a basic UI that included two main panels:

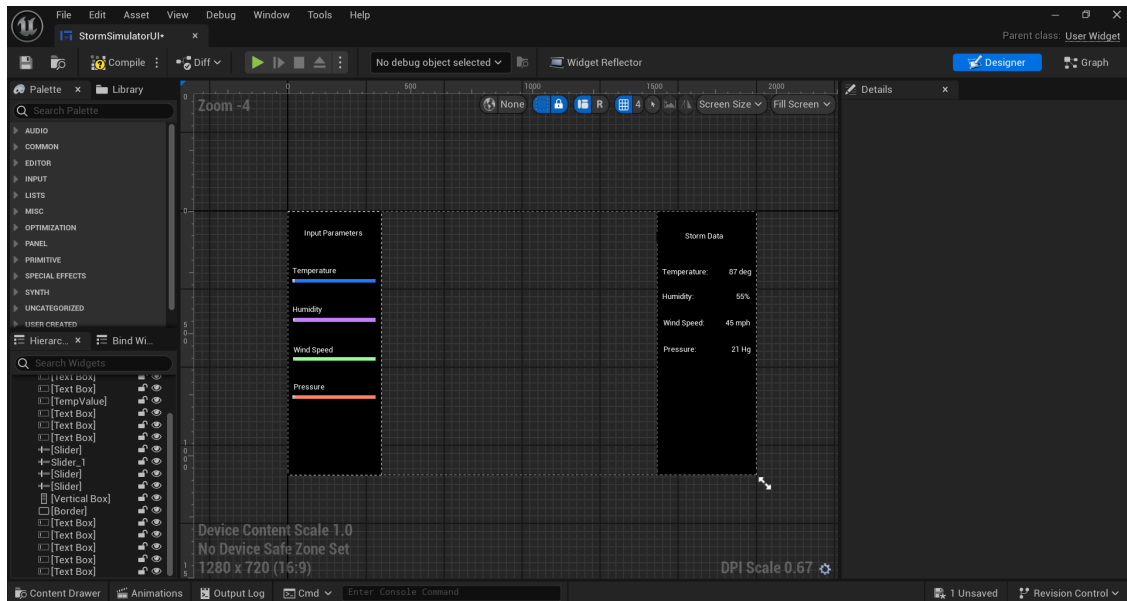
- **Left Panel:** A set of sliders for user input (e.g., wind speed, temperature).
 - **Right Panel:** Text blocks to display real-time data based on the slider values.
-

1.3.1 2.1. UI Design Strategy:

- **Left Panel (User Input):**
Contains sliders that allow the user to adjust parameters for the storm simulation.
 - **Sliders:** Linked to variables such as wind speed and temperature.
 - **Right Panel (Real-Time Data Output):**
Displays data such as the current wind speed, which updates in real time as the user interacts with the sliders.
 - **Text Blocks:** Bound to the slider values to show live updates.
-

1.3.2 2.2. Creating the UI Elements:

1. **Canvas Panel:**
 - This was the root container used to organize all the elements.
 - **Vertical Boxes** were used to align the sliders and text blocks vertically within each panel.
2. **Adding Sliders:**
 - **Sliders** were added to the left panel to capture user input. These were intended to control parameters like wind speed and temperature.
3. **Adding Text Blocks:**
 - **Text Blocks** were added to the right panel to display the current values of the sliders. These values were dynamically bound to the sliders.
4. **Adding a Background Panel:**
 - To improve the UI's readability, a **black background panel** was added behind both the sliders and text blocks. This was done using a **Border** widget.



1.4 Implementing Real-Time Data Updates:

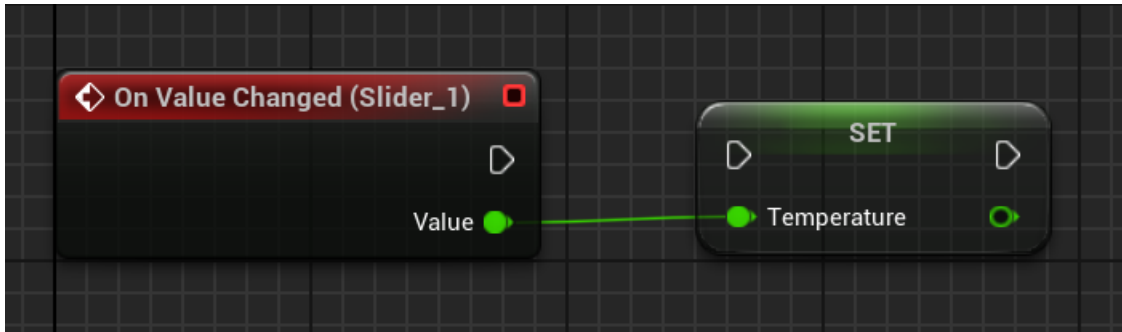
The goal was for the text fields on the right panel to update immediately when the user changes the sliders on the left panel.

1.4.1 3.1. Creating Variables for Real-Time Data:

- **Creating Float Variables:**
In the **Blueprint Graph**, I created variables to store the values of the sliders (e.g., `WindSpeedValue`, `TemperatureValue`).
 - These variables were updated whenever the user interacted with the sliders.

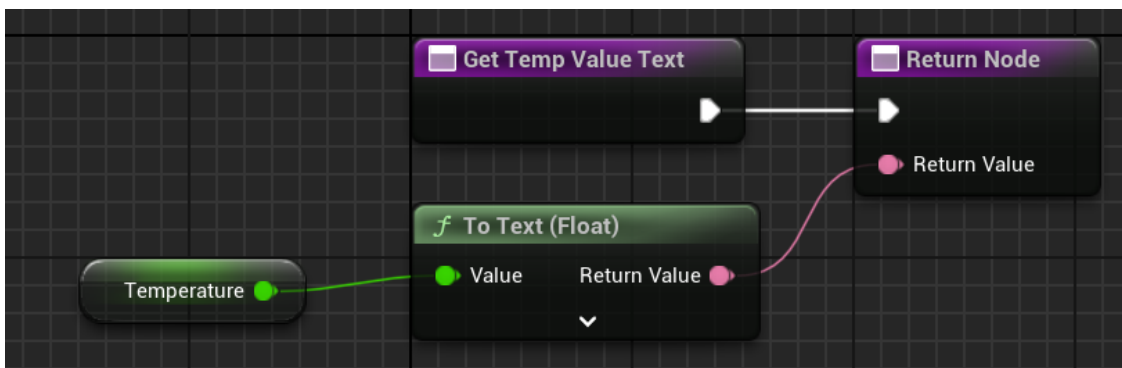
1.4.2 3.2. Binding Sliders to Variables:

- **On Value Changed Event:**
For each slider, I implemented the **On Value Changed** event. This event triggers whenever the user moves a slider.
 - The event was connected to a function that updates the corresponding variable (`WindSpeedValue`, `TemperatureValue`).



- **Binding Text Blocks to Variables:**

The **Text Blocks** on the right panel were bound to the slider variables. Whenever the slider value changed, the text block updated to display the new value in real time.



1.5 Testing and Next Steps:

After implementing the real-time data updates, I tested the UI to ensure that changes in the slider values were correctly reflected in the text fields.

1.5.1 4.1. Testing the UI:

- Ran the Unreal simulation and verified that the slider interactions worked as expected.
- The text blocks successfully displayed the updated values in real time.

1.5.2 4.2. Challenges Faced:

- **Binding Variables:**

Learning how to properly bind variables to UI elements was a new concept. Through trial and error, I figured out how to link the sliders and text blocks via the **Blueprint system**.

- **Solution:** Used the **On Value Changed** event and created binding functions for the text blocks.

2 Conclusion:

- Successfully implemented a basic UI with two panels (input and output).
- Achieved real-time data updates based on user interaction with sliders.

2.1 Next Steps:

- Finalize the design and functionality.
- Add a play/pause button for the simulation.
- Connect the sliders to actual storm simulation data once the Neural Network is in place for the calculations.

[]:

