The Electric Circuit Simulation Development Report

The Electric Circuit Simulation is an interactive learning tool designed to help students understand how electrical circuits function by exploring concepts such as conductivity, resistance, and circuit behavior through hands-on experiments.

How the Simulation Works:

The simulation is fully interactive and operates directly in a web browser, allowing students to manipulate circuit components and observe real-time changes in electrical behavior.

Students can define their own circuit configurations, selecting different materials, wire types, and electrical components, which dynamically impact how the circuit functions.
They can:
- Add electrical devices and decide whether they should be turned on or off.
- Insert resistors to observe how resistance affects current flow.
- Modify the width and length of the conductor, demonstrating how these factors influence electrical conductivity and overall circuit efficiency.

To make the simulation accessible to a wider audience, all in-game text has been translated into multiple languages, including:
- English (default language).
- Hebrew (LTR, right-to-left adaptation).
- Arabic (RTL, using ArabicSupport for proper text rendering).

When switching to Arabic, the system correctly adjusts text alignment, word order, and UI layout, ensuring proper readability and usability.

The simulation was developed using the following technologies:

- **Unity 2021.3.30 (LTS)** – main game engine.
- **C#** – scripting language for logic implementation.
- **TextMeshPro** – for rendering high-quality text descriptions.
- **Unity WebGL** – for web deployment.
- **Optimized PNG textures** (RGBA Crunched DXT5, BS3) – used to reduce build size while maintaining image quality.
- **Particle Systems** – used for various visual effects, such as:
  Steam from the kettle to demonstrate heat energy transformation.
  Flowing electrical particles to visually represent current movement.
- **Multi-language support** (English, Hebrew, Arabic) – implemented using Arabic Support for proper RTL text rendering
- **External API Communication** – The simulation interacts with an external API written in TypeScript.
  Uses System.Runtime.InteropServices to send messages between C# (Unity) and JavaScript/TypeScript.
  Receives instructions from the API, allowing dynamic simulation behavior based on user-defined parameters.

The main interactive features of the simulation include:

- The simulation includes several interactive features that allow students to experiment with electrical circuits dynamically. Key interactive elements include:
- Custom Circuit Building – Students can freely place and connect components such as wires, batteries, resistors, and electrical devices to create unique circuit configurations.

- Power Control – Students can turn electrical devices on and off, observing how circuit behavior changes in real time.
- Resistor Adjustments – Users can add and modify resistors, changing resistance values to see their effect on current flow.
- Wire Customization – The length and thickness of conductors can be adjusted, demonstrating how these factors influence electrical resistance and current strength.

- Real-Time Particle Effects – Flowing electric particles visually represent the movement of the current, while sparks and steam effects illustrate energy transformation.
- Live Circuit Reactions – Changes in circuit components instantly update voltage, resistance, and current flow, providing immediate feedback for students.

Challenges and Solutions

1. **Text Readability and RTL Support (Localization)**

Problem:

When adding **multi-language support (English, Hebrew, Arabic**), text in **Arabic** was displayed incorrectly, causing word order and alignment issues.

Solution:

- Integrated the **ArabicSupport** plugin to ensure **proper Arabic text rendering.**
- Dynamically adjusted the **UI between LTR (left-to-right) and RTL (right-to-left).**
- Optimized fonts using **TextMeshPro**, improving **text clarity in WebGL**

2. **Problem:**

One of the key challenges in developing the simulation was **designing a system that allows the gradual introduction of circuit components and features** without overwhelming students. The simulation needed to provide **progressive complexity**, ensuring that in the initial lessons, only essential components were accessible, while advanced functionalities could be introduced later.

Additionally, **various parameters** needed to be adjustable, such as:

- **Which objects are visible and interactive** in each lesson?

- **Initial circuit conditions** (e.g., whether the circuit starts as open or closed).
- **Resistor properties** (initial resistance values and whether they can be modified).
- **Wire dimensions** (initial width and length and whether students are allowed to change them).

Without a structured system, manually controlling these parameters for different lessons would be inefficient and difficult to manage.

Solution:
To solve this issue, a **flexible configuration file system** was implemented. This system enables **dynamic lesson adaptation** by using a structured set of **Boolean values and numerical parameters**, allowing fine-grained control over simulation settings.

- **Object Visibility and Interactivity:** Educators can specify which components (e.g., **kettle, TV, fan**) are available for interaction in each lesson.
- **Circuit Properties:** The configuration defines whether the circuit starts **open or closed**, ensuring structured lesson flow.
- **Resistor Settings:** The **initial resistance value** of resistors can be pre-configured, and the system determines **whether students can modify resistance** during the lesson.
- **Wire Properties:** The initial **length and thickness of conductors** can be predefined, with the option to restrict modifications or allow students to experiment with different wire configurations.

Conclusion

By implementing this **configurable system**, the simulation provides a **scalable and customizable learning environment**, ensuring that students engage with new concepts progressively while maintaining control over lesson complexity.