UNIVERSITY OF INFORMATION TECHNOLOGY AND SCIENCES

Department of Computer Science and Engineering

Simulation and Modeling Lab

CSE 413

Project Report

Haunted House Simulation

Submitted To:

Audity Ghosh

(Lecturer, Department of CSE)

Submitted By:

	Member: 1	Member: 2
Name :	Sazzad Hossain	Samin Yeaser Rafid
ID :	0432220005101103	0432310005101101

Batch/Section : 52-7B1

Date : 20-11-25

Haunted House Simulation

1. Introduction

This project creates an animated haunted house scene that transitions smoothly from day to night. Using mathematical models and real-time rendering, the simulation features dynamic elements including a bobbing ghost, moving clouds, glowing windows, random lightning flashes, and appearing stars. The animation demonstrates linear interpolation for color transitions and physics-based movements to create an immersive spooky atmosphere through computational graphics.

Features :

- Smooth day <-> night transition using a control variable transition in [0,1]
- Sun/Moon color interpolation
- Clouds that move horizontally (cloudOffset)
- Stars that fade in as night approaches
- House with windows that glow at night
- Ghost bobbing using a sine function
- Random lightning flashes during deep night
- Graveyard, fence, and spooky trees

Math / Modeling notes :

1. Day-Night transition: a scalar $transition(t) \in [0,1]$ that is linearly advanced toward a target (day or night) with a constant rate transitionSpeed. This interpolates scene colors: color = (1 - transition)*day_color* + *transition*night_color. (This is

linear interpolation — LERP.)

2. Ghost vertical bobbing: modeled with a sinusoid y_ghost(t) = y0 + A * sin(ω t) where A is amplitude and ω controls speed. This demonstrates simple harmonic motion (no restoring force in the simulation, purely aesthetic).
3. Lightning: modeled as a random impulsive alpha(t) that decays exponentially per frame when triggered. We use a Poisson-like chance each frame to trigger a flash, then decrease its alpha by a decay factor.
4. Clouds: horizontally translated by a slow cloudOffset(t) = cloudOffset + v_cloud * dt.
5. Windows glow intensity tied to transition (night), giving a visible coupling between environment state and lit objects.

2. Methodology

- Software/Tools: List the primary libraries used (e.g., numpy, matplotlib.animation, ffmpeg-python).
- Simulation Parameters: Detail the initial conditions and constants. You can copy relevant parameters from the code:
  - FPS, TOTAL_TIME, FRAMES
  - transition and transition_speed
  - ghost_amplitude, ghost_omega
  - lightning_decay, lightning_prob_per_frame
  - cloud_speed
- Core Algorithms/Formulas: Explain the mathematical models behind the dynamic elements (as noted in the code comments):
  - Day-Night Transition (LERP): color = (1 - transition)*day_color +

  transition*night_color

  - Ghost Vertical Bobbing (SHM): y_ghost(t) = y0 + A * sin(ω t)
  - Lightning: Random impulsive alpha with exponential decay.
  - Cloud Movement: Linear translation based on cloud_speed.
- Scene Construction: Briefly describe how static and dynamic elements are drawn using matplotlib.patches and artists.

3. Results

- Observation of Dynamics: Describe how each animated element behaves over the 10-second simulation.
  - *Day-Night Cycle:* When does it transition? How do colors change?
  - *Ghost:* Describe its movement and appearance relative to the day-night cycle.
  - *Lightning:* How frequently does it strike? How does it fade?
  - *Clouds:* Describe their movement.
  - *Windows/Spiderweb:* When do they become active?
- Visual Output: Reference the generated haunted_house_sim.mp4 video. You might consider including screenshots at different times (e.g., day, dusk, night with lightning).
- Quantitative Data (if any): If you were to track specific metrics (e.g., average lightning frequency), this section would be where you'd present them. For this simulation, it's mostly qualitative visual results.

4. Discussion

- Effectiveness of Models: Evaluate how well the mathematical models (LERP, sine wave, exponential decay) achieved the desired visual effects.
- Parameter Impact: Discuss how changing certain parameters (e.g., transition_speed, ghost_amplitude, lightning_prob_per_frame) might alter the simulation's aesthetics or realism.
- Strengths & Limitations:
  - *Strengths:* What aspects of the simulation are particularly effective or well-implemented?
  - *Limitations:* What could be improved? (e.g., more complex cloud shapes, more varied ghost movement, interactive elements).
- Unexpected Behaviors (if any): Did anything in the simulation behave in a way you didn't initially expect?

5. Conclusion

- Summary of Findings: Briefly summarize the main observations and the success of the simulation in meeting its objectives.
- Future Work: Suggest potential enhancements or extensions to the simulation (e.g., adding more characters, dynamic camera, environmental sounds, more realistic physics).

6. Appendix :

Small summary of code:

Main Components:

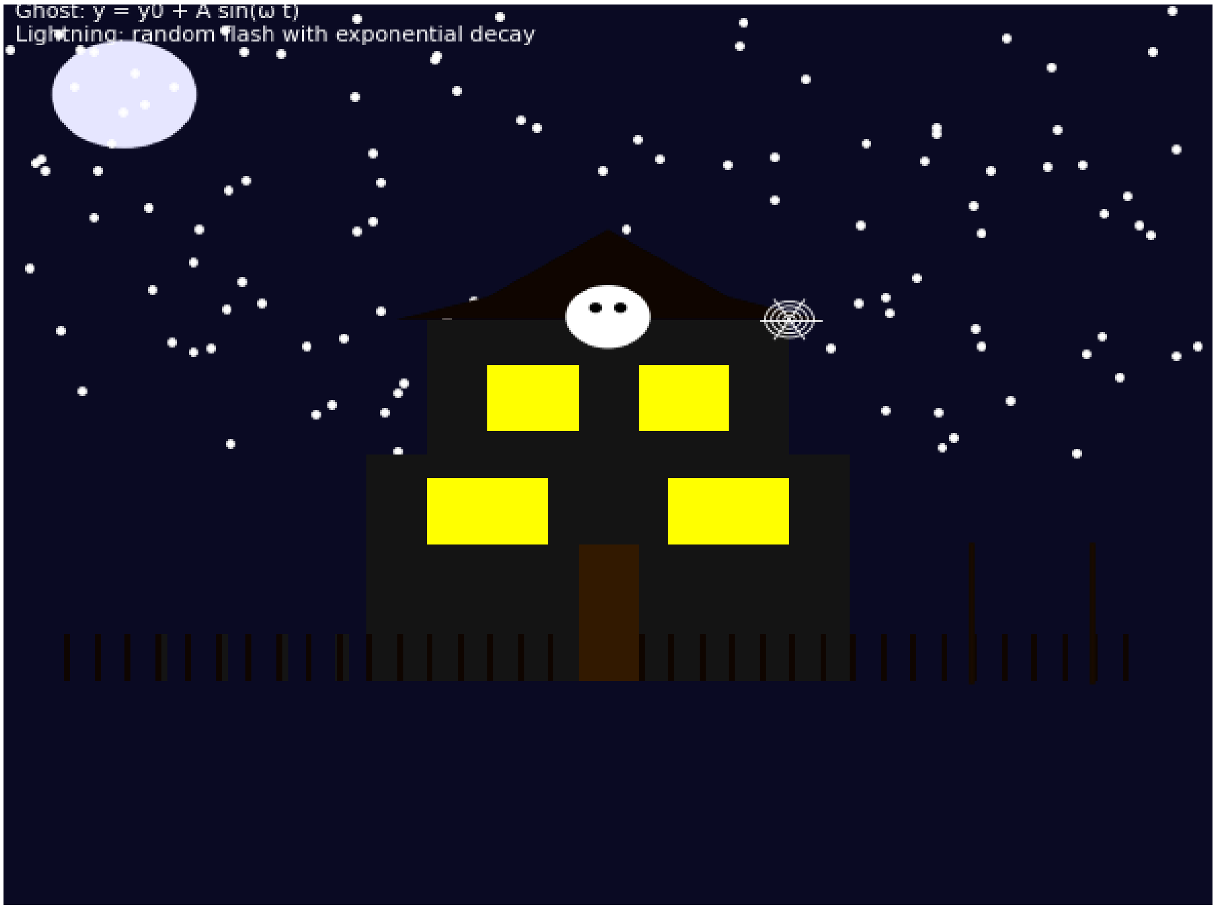
- **Scene Elements**: House, ghost, clouds, sun/moon, stars, trees, fence, graveyard, and lightning
- **Day-Night Transition**: Smooth color interpolation between day and night skies
- **Animations**:
    - Ghost bobbing up/down with sine wave motion
    - Moving clouds that fade out at night
    - Stars that appear gradually
    - Random lightning flashes during deep night
    - Spider web that appears at night
    - Glowing windows that intensify at night

Technical Features:

- Uses matplotlib for rendering and animation
- Linear interpolation (LERP) for color transitions
- Frame-by-frame updates for real-time effects
- Physics-based movements (sine waves, exponential decay)
- Exports to MP4 video format

The simulation runs for 10 seconds at 60 FPS, creating a spooky atmospheric animation that transitions from daytime to nighttime with appropriate visual changes.

Screenshots: Additional visual evidence from the simulation.



Github Link:

https://github.com/SazzadHossain103/SimulationLabProject/blob/main/Team_Shinobu.ipynb