Project Requirements Document

CSE 481v AR/VR Capstone

Fall 2018

Electric Field Visualizer

Hi 5

Candice Miao
Jeff Ranhao Xu
Phoenix Meadowlark
Ryan Smith

Summary

Electric Field Visualizer is a 3D vector visualizer for Jared Canright's 'Physics Not Included' EM Physics engine. We will mainly be using his work in Electromagnetic Field calculations to feed values to our vector visualizations.

Our visualizer will be a 3D grid of 3D points and visual vectors. Each visual vector will represent the strength of the electric field at it's given position, which will be influenced by an placed charges in the scene. The length of the visual vector will represent the strength of the electric field and the direction will point towards the direction of the field at that point. The vector will change color at a maximum and minimum size to denote that the field is actually stronger/weaker at this point but we must limit the visualization. We also will have showable text values that can be turned on and off with a boolean value. The grid will be customizable in the script for easy changing of density of vectors in a scene and width, height, length of the field.

We will also have a simple app experience that will use the visualizer. Users may place kinematic charges in the scene, adjust the magnitude of the charges, and be able to teleport around the scene. We will also allow the user to show/hide the text values of the field vectors.

Deliverables

- Electric Field Visualizer
    - Takes in values from 'Physics Not Included' but will be generic enough to be able to access values from other API's with a different callback
    - Vector visuals on a grid, representing magnitude and direction of the field at each position in the grid
    - Grid size in both point density and dimensions of the field are configurable in the script for easy changing and reuse.
    - Visualizer will indicated maximum and minimum sized vector to both bound the visualization and allow user to know when the magnitude isn't fully represented by the vectors
    - Visualizer will have optional text display of the magnitude
- VR application
    - Simple VR application using our Electric Field Visualizer
    - User may place kinematic charges of negative and positive charge in the scene and the Visualizer will automatically update based on charges in the scene, in real-time
    - User may also teleport around the scene and view text values for magnitude
    - A stretch goal will be to allow users to add charges of varying charge strength with charges clearing showing charge either by size or by color. If by color, we will also need a legend / key
    - Potentially have a magnetic mode with magnets added to scene
- Kickstarter style website tracking progress and demoing the current product
- Kickstarter style hype demo video

Critical Features

Minimum Viable Product

- Electric Field Visualizer is readable and responds to charges in a 3D environment, in real time
- User can place charges and move around a scene using the EFV
- User can see text values of each vector(magnitude) when desired

Final Product

- Vector visuals are attractive and well designed
- Visuals indicate when a vector is at max or minimum magnitude.
- Vector field can have varying density and dimensions

Stretch Goals

- Users can place charges with varying magnitude, and that magnitude is well represented in the scene.
- Users have direct control over the density of the vectors that are visualized so the user can see more granularity vs easier to see through the field without obstruction
- Equipotential Service, from a position, create an equipotential orb connected around the nearest charge or a series of 2d potential planes
- Ability to connect the ends of the vectors with lines to view surfaces based on magnitude?

Performance Metrics

- Functionality - Is the user able to complete the following tasks
  - Add positive and negative charges to the scene
  - Tell the differences in magnitude and direction of the field vectors
  - Teleport around the scene and place charges where desired
- Enjoyability - Is the user able to enjoy interacting in the app? Is the demo app easy or frustrating to use figure out?
- Teaching Moment - Is a user able to develop a sense of electric fields around a charge and the interaction multiple charges has on a surrounding field?
- Total Field Visualization - Does the user have a sense of a seamless electrical field or are each point on the grid felt more as a stand-alone magnitude?

Milestones

Week 1 (Oct 1 - Oct 5)

Team interaction, scheduling, initial brainstorming

- Full Team

Week 2 (Oct 8 - Oct 12)

Project proposal and presentation, set up DevOps and UW-Gitlab repo

- Full Team

Week 3 (Oct 15 - Oct 19)

Initial Kickstarter style website, Project Requirements Document, build bare bones Unity project on Acer Headset

- Candice Miao - project management, assist Unity build
- Jeff Ranhao Xu - building project to Acer Headset with controllers API's downloaded
- Phoenix Youngman - website creation and design
- Ryan Smith - first draft PRD, assist website creation

Week 4 (Oct 22 - Oct 26)

Explore Unity 3D Physics, Colliders, Rigidbodies, and Physics Forces. Demo - Working block interaction with Dell VR Controllers. First meet with physics department to adjust direction to fit departmental needs.

- Candice Miao - Explore SteamVR SDK, Unity Text for HUD
- Jeff Ranhao Xu - Explore SteamVR SDK for working with Dell VR Controllers
- Phoenix Youngman - Blog update, conceptual drawings and ideas for EM demos
- Ryan Smith - GPE demo in Unity/SteamVR with live text update, GPE and KE calculations

Week 5 (Oct 29 - Nov 2)

Meet with Physics Department on Tuesday to brainstorm a EM demo and see what their department is working on in VR internally.

- Candice Miao - Gather resources for vector visualizations, practice with physics-not-included
- Jeff Ranhao Xu - Begin working on an EM demo, practice with physics-not-included
- Phoenix Youngman - Gather resources for vector visualizations, practice with physics-not-included
- Ryan Smith - Begin working on an EM demo, practice with physics-not-included

Week 6 (Nov 5 - Nov 9)

Settle on Demo concept, Begin developing Electric Field Visualizer

- Candice Miao - Display vector from a float value
- Jeff Ranhao Xu - Rework physics controls to remove palette and just have an add tool, research raycasting
- Ryan Smith - Write C# Script for creation and updating of Electric Field Visualizer

Week 7 (Nov 12 - Nov 16)

Begin demo scene that uses EFV

- Candice Miao - User can teleport around scene
- Jeff Ranhao Xu - User can place charges in scene, adjust charges, control options with left hand

- Ryan Smith - EVF updates in real-time based on charges placed in scene. Vectors now have a min/max value indication. Start to work on text display

Week 8 (Nov 19 - Nov 23)

Improving visuals and PEG feedback

- Candice Miao - Meet with PEG and demo current progress / gather feedback
- Jeff Ranhao Xu - Find and implement improved visuals for the vector displays
- Ryan Smith - Make EVF more dynamically configurable, be able to change density of points and dimensions of the grid

Week 9 (Nov 26 - Nov 30)

Respond to PEG feedback, in depth testing

- Candice Miao - Extensive testing of App/Demo experience + feedback
- Jeff Ranhao Xu - Edit 3D scripts based on PEG feedback to help improve student takeaway
- Ryan Smith - Edit 3D scripts based on PEG feedback to help improve student takeaway

Week 10 (Dec 3 - Dec 7)

Finalizing Demo, attempting to make styling and experience more professional

- Candice Miao - Testing, debugging, and design tweaks
- Jeff Ranhao Xu - Testing, debugging, and design tweaks
- Ryan Smith - Testing, debugging, and design tweaks

Materials and External Help

- Extensive use of Unity's physics scripts
- Extensive use of SteamVR SDK for controller interaction
- Extensive use of 'Physics Not Included' to help with charges and electric field calculations
- Reserve our budget for potentially adding more professional graphics, 3D models, or background textures to the demo

Risks

- **We can't properly visualize a 3D vector field because it is too computationally expensive or too difficult to figure out.**
  We can have some fixed, non interactive demos with 3D vector field animations that mock a field instead of resolving one from actual computations
- **Difficulty working with Physics-Not-Limited project and coming up with a shared demo vision with PEG**
  Our Electric Field Visualizer should be as agnostic as possible to 'Physics Not Included' so we can demo the visualization without specifically using Jared's engine
- **The user isn't engaging with physics and just playing without learning**
  We need to take care to direct user attention to the physics aspect of the demo and make sure they realize that our grid is meant to show a continuous field. We might need to play with increasing density while decreasing the vectors visual footprint to achieve this goal
- **The user is unable to relate the changes in the vectors with the changes in an electrical field**
  We may require an introduction to the user at the beginning of the scene to describe the visuals or possible add a HUD key the the user can reference if things seem too confusing