Team Grapid Requirements

by

Yoonah Lee, Shah Rafique, Matthew Smitherman

Project Description

The program will generate various objects each with unique textures, movement and data needed to render properly. Additionally the program will correctly render each object based on where the eye is by accepting user input for movement of the eye. User input will also consist of allowing the user to create various predefined objects and place them.

Functional Requirements

- Display cube's, square pyramid's and sphere's.
- Display multiple simulations textures based off the texture assigned to the object at creation.
- Ability to disable textures for a object and instead use colors.
- Rotate objects.
- Animate the rotation.
- Handle user input to change location of the eye.
- Handle user input to create new objects at run time
- Texturing along with correct orientation of textures based on object orientation.

Quality Requirements

- Development process to ensure quality

Our main development process model is rapid prototyping, an approach in which poorly understood requirements are usually
implemented first. While the prototype is constructed, factors like efficiency, maintainability, and documentation are secondary to the creation of functioning models. This approach was selected due to the team's small size and relative inexperience with selected project tools and languages.

The development process ended up as follows: Each member pursued different areas of the program to make prototypes. After each prototype was complete, the prototypes would merged together. Group member Yoonah Lee was tasked with creating a sphere and allowing for rotation of a objected based on user input and loading in multiple textures. Group member Shah Rafique was tasked with reducing copied code, streamlining the pipeline, creating the base object, square pyramid and cube object for rendering, simultaneous textures and rotation and animation. Group member Matthew Smitherman was tasked with user GUI, creation of objects during run time, customization of user placed objects, node.js and usage of Vue.js.

Errors

Reliability is not a critical aspect of this application, as it is not a control application for security or safety-critical systems. However, in the interest of ease-of-use, we want to avoid system failure in the general case as much as possible, with the following error cases being of particular concern:

1. User input: User enters some form of input that is either unrecognized or unaccepted and thus not accounted for.
2. Object constructors are passed a incorrect parameter which happens to be uncaught when assigning that parameter to a field.
3. Stack overflow, divide by 0, logic errors and matrix transformations accuracy degradation.
4. Overtaxing of system resources such as running out of VRAM/RAM or overburdening CPU.
5. Graphical errors due to GPU type.

Maintenance

- Maintenance will only be available for 2nd semester of the 2017-2018 academic year.

Platform Requirements

Operating Systems

- Windows 7, as it still common to see it in use and also because one of the group members is using windows 7.
- Windows 10, latest in Microsoft operating systems and also the operating system of most of the university lab computers.

Tools

- Browser capable of using WebGL and capable of accepting non JavaScript files such as textures.

  - Notepad++
  - Mozilla WebGL API.
  - Browser console.

- Hardware

  Basic CPU to be able to perform calculations.
  Basic GPU, can be in CPU but has to be able to use OpenGL 4.3+ (the latest version of openGL supported by university lab computers).
  Basic storage device to be able to contain program.
  Basic RAM and enough RAM allocation so that the program can run.
  Basic Monitor so that the user can see the graphics.
  Basic keyboard and mouse so that the user can interact with the program.

- Software

  - Firefox, Microsoft Edge or Safari browsers as they allow for non JavaScript files to be loaded.

Process Requirements

- People

  - Programmers
    - Yoonah Lee
    - Shah Rafique
    - Matthew Smitherman
  - Professors/Advisors
    - Chandana Ariyawansa (C.A.), professor for CSC 495: Intro to Graphics.
    - Ike Quigley, professor of CSC 340: Software Engineering.

- Costs

  - Time, as learning graphics and creating matrix libraries will take time.
  - $0, note that the GPU and web browser has to be able to use WebGL and accept non JavaScript files.

- Dates (Deadlines)

  - 4/23/18: Multiple objects with textures or color only
  - 4/25/18: Incorporate sphere into object hierarchy

- 4/26/18: Start combining object classes into the Vue framework to allow those objects to be creatable
- 4/27/18: Implement user control and polish structure of program,present to CSC 340 class
- 4/28/18: Incorporate lighting
- 4/29/18: Add more user controls such as specify position,color and texture of objects.
- 4/30/18: Clean program up.
- 5/1/18: Clean program up, present to CSC 495 class