Project Title: Interactive robot builder with WebGL

Group Member:

Bryce Hendricks & Xiaofeng Qu

Description:

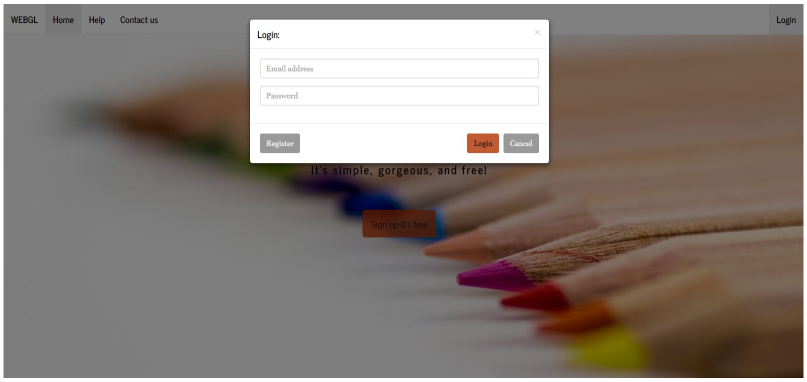
Our application is a an interactive texture mapping platform. It allows users to upload images and apply them directly to a WebGL shape.

Work Divided:

Bryce Hendricks- My tasks included the building shapes in WebGL and mapping textures to them respectively. WebGL is code written in JavaScript that utilizes the GPU for the creation of 2D, 3D, shapes and animations. Unfortunately I was unable to make much progress on generating spheres in WebGL. This was a major setback for my part of the project. I attempted generate a sphere similar to the cube that is working but I had no success. I was only be to complete a texture cube that users can add to the canvas and apply textures.

Xiaofeng Qu: Xiaofeng's developed the UI, texture uploading function, user registering and logging function, and drag and drop function.

Firstly, Xiaofeng developed the Web UI using HTML, CSS, JavaScript, and jQuery. Users can create their user accounts, login into their accounts, upload their textures, and apply their textures to the animated cubic. The Web UI is fully responsive achieved by Bootstrap.



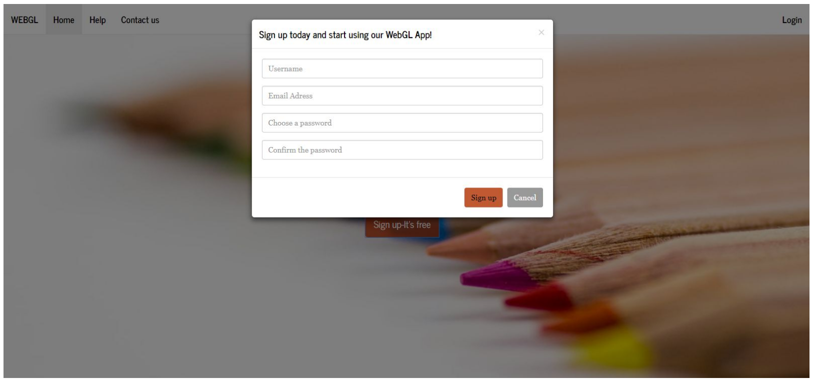
Login:

Email address

Password

It's simple, gorgeous, and free!

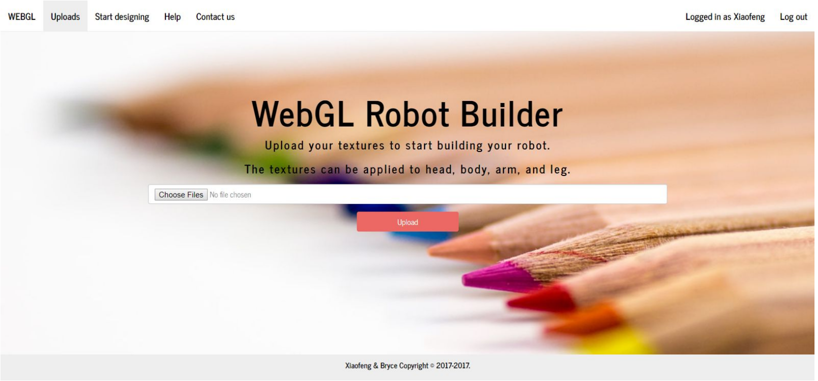
Fig 1. Login page.



Sign up today and start using our WebGL App!

Fig 2. Sign up page.

Secondly, Xiaofeng developed the functions using PHP and MySQL for users to create accounts, login accounts, upload textures to the server, and store their textures' information in database. The users' textures will display in their design page when they logged in.



WebGL Robot Builder

Upload your textures to start building your robot.

The textures can be applied to head, body, arm, and leg.

Xiaofeng & Bryce Copyright © 2017-2017.

Fig 3. Upload form page.

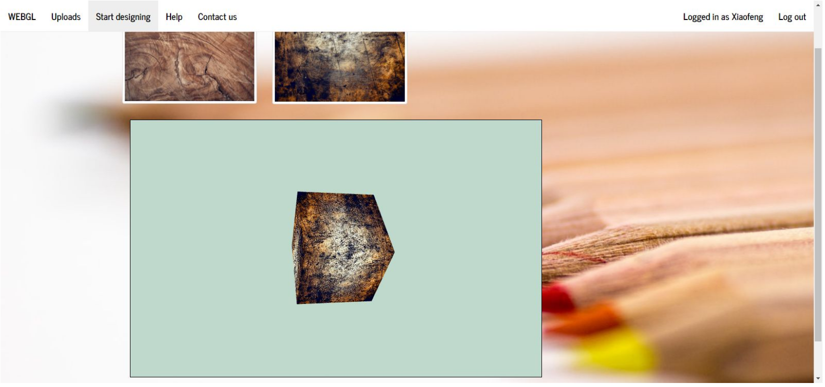


Fig 4. Design page.

Thirdly, Xiaofeng developed the function that users can drag their textures into a canvas and the url of the picture is sent to Bryce's part to apply to the animated 3D cubic.

Architecture:

Our application consists of HTML, JavaScript, CSS, jQuery, Bootstrap, PHP, MySQL and WebGL.

In our project, the user GUI is written in HTML/CSS with customized and Bootstrap styles; and the events are handled via JavaScript and jQuery.

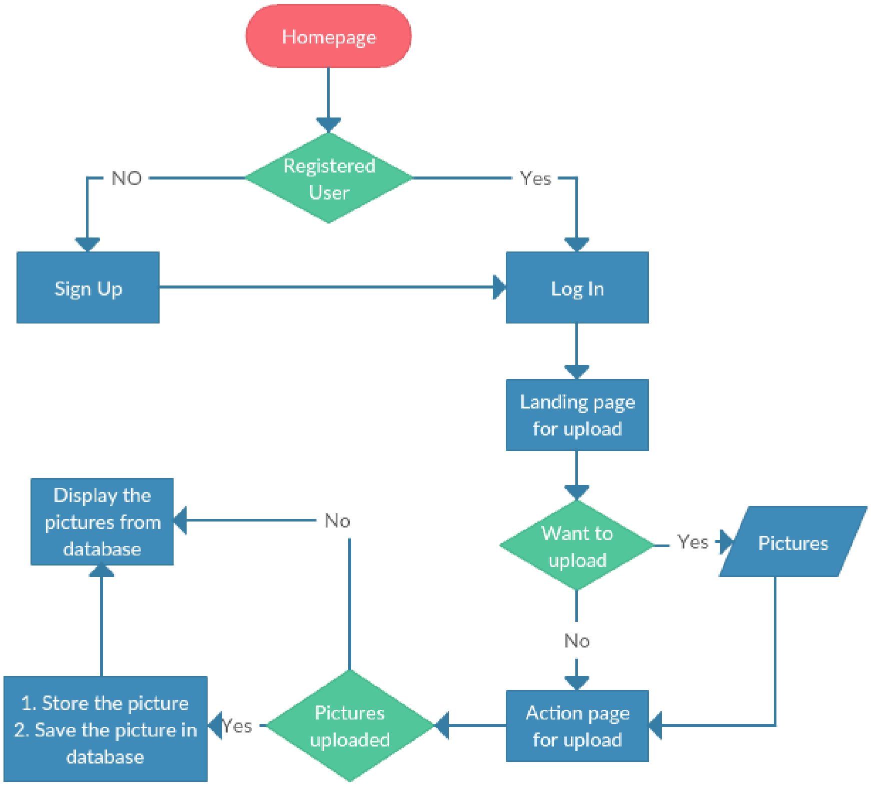


Fig 5. Diagrams of system design.

As shown in the above graph. The user first goes to the homepage (index.php). If the user is registered, he can directly login the system. If the user is not registered, he can register an account and then login. After the user logins, the website will direct the user to the upload landing page, so the user can upload textures. If the user does not want to upload any texture, he can click the Start designing link and go to his design page. The design page will load all the user's textures based the database's records and display them in the design page. If the user upload some textures, the design page will store the textures into the server's file system and store the textures information in database. Then, the design page will load all the user's textures and display them in the design page. The user can drag their textures into the canvas

and the texture will be apply to the 3D WebGL objects. The web application also has the ability to detect whether the user is logged in or not. If a user is not logged in, he can only visit the homepage. If a user is logged in, the homepage will always redirect the user to the upload page. In the back-end, JavaScript is handling events and WebGL is generating shapes, textures, and animations. The application is a highly interactive event driven construct.

Conclusion:

1. **We created a 4-tier web application using PHP and MySQL.**
2. **Functions including register, login, upload and save textures, and applying textures to an animated 3D WebGL cubic.**