Liberation

I’m currently lead developer on Liberation, a 3D action shooter game. Developed in UE4

Hey Welcome!

I’m Austin Chiang.

I’m a software developer located in Irvine, California.

Always set environment variables on windows for npm and java

Project for personal website

Installed packages

* NPM
* Node.js
* React.js
* Using NPM
  + Webpack
  + Webpack-cli
  + Babble-loader
  + @babel/core
  + @babel/preset-emv
  + @babel/preset-react
  + React
  + React-dom
  + Jquery
  + Nodemon
  + Express
  + Body-parser
  + @types/react
  + @types/reactDOM
  + --save url-loader
  + **--save style-loader css-loader**
  + React-reveal
  + Pose
  + React-rotating-text

Package Descriptions

* NPM is package manager
* Node.js package of useful things including npm and stuff
* React is for front end framework (builds separate components) for on hover and like mouse over (state configuration)
* Webpack is for bundling react components (1 file)
  + Webpack gets all of the components and makes it into one .js file or css file etc.
  + Webpack-cli is for interacting with webpack (like client) need to install with webpack
* babel is for reading code and changing it live
* @babel/preset-emv
  + Is like vanilla js
* @babel/preset-react
  + Allows babel to read both vanilla and react js
* Nodemon
  + Any changes you make to your folder, will continuously update test environment (convenience)
  + Makes it so you don’t have to run npm start everytime you start up test environment
* Express
  + Server side package
* Body-parser
  + Option that allows you to change string version of data to actual objects (kind of like casting in c++)
* --save url-loader
  + For url parsing in react
* --save file-loader
  + For file parsing
* {
* test: /\.(gif|svg|jpg|png)$/,
* loader: "file-loader",
* }],
  + Throw this after loader in webpack.config.js to let webpack load files
* **--save style-loader css-loader**
  + For css parsing in react
* {
* test: /\.css$/,
* loader: "style-loader!css-loader"
* },

Steps:

1. Installed Node.js, npm (part of node), created git repository
2. Set environment variables
3. Installed all packages above using npm
4. Created client, database, server folders
5. Created webpack.config.js file (empty)
6. Copied webpack stuff into the config file (check jun sub’s email)
   1. This is what runs when you use webpack
7. Go to server folder and make index.js
   1. Make an express server
8. GO to package.json and add script for npm to use

Ex. "start": "nodemon Server/index.js",

1. Use NPM start to start server
2. Now start adding react components (look into files for comments)
3. GOOD LUCK!

To run things

* NPM start
  + Starts the server
* Webpack
  + Updates the packs to refresh page essentially

Npm init -> creates package.json which is for having all the package stuff put in

Npm install -g <package> -> globally means on computer (need to set path variable if so) get rid if just want local

Whenever you run test environment use npm start

Let in js

* new version of var is outdated
* Use let and const