React game engine

Evobi web development interns

February 26, 2020

1 Introduction

1.1 Highlighting

It is a basic game engine to create a simple game so that you can get to know how the game can be created.

1.2 Getting started

These instructions will get you a copy of the project up and running on your local machine for development and testing purposes.

1.3 Prerequisites

What things you need to install the web app-
- NodeJs
- NPM package manager

2 Installing

To install this web app
- Download the zip folder and extract it.
- Run npm install.This will install all the required libraries.
- Then type npm start.This will run web app locally on server http://localhost:8080/

3 Working at user level

3.1 Objects window

In object window ,there are two components
- a carousel
- elements which shows selected elements.

3.2 Assembly window

All the selected elements are shown in gray div and workspace in yellow. - Objects can be resized only in the gray window. - Each object can be dropped to (yellow) workspace.
- After dropping each element "OK" button needs to be pressed to reset the position x and y to zero.
- In yellow workspace "Camera div" can also be dragged and dropped to provide the first scene of the game .

3.3 Logic window

In this window , you can give properties to the elements and can preview the sequence of actions(move ,jump) given.
- You can either give the movable property or static property to the object.
- For movable object you can give predefined move and jump value or you can give keyboard interactions.
- If you choose keyboard interaction and sequencing together then,only keyboard interaction will work.
- For the static element , there's a overlap property using which element can be allowed to overlap or it can collide.
- In miscellaneous property, there's an option to add sound and give the shooting property to the player and you can use this by pressing shift key.

3.4 Preview window

In this all the actions are shown in sequence and you can press "preview" button to see the preview of those actions. The preview will be opened in a new tab with only player and a platform.

3.5 Game window

After giving properties , press "Play" button to launch the game.

4 Development

4.1 Index.html

This will be the first file which will be executed .In this file, there is a div with id "root" and links for carousel.

4.2 src/index.js

In this file, react,reducers and routers are linked and "root" id is selected and components are displayed there.

4.3 src/router.js

In this file, specified files are routed to specific routes.

4.4 src/Components/App.js

This file is used to display all the containers.

4.5 src/Containers/objects.js

Functions-

1. render
2. renderlist - Shows all the objects in carousel.
3. renderlist2- shows selected items in elements component.

Reducers- It is taken as props in mapStateToProps(state) function.

1. themeselect-contains all the objects with its properties
2. objectselect-contains all selected elements and can be manipulated using object_select action.

3. id1

Actions- It is taken as props and manipulated to dispatch mapDispatchToProps(dispatch) function.

1. Object_select
2. Increment

4.6 src/Containers/choose.js

It is used to show 3 tabs - objects , assembly , logic.

4.7 src/Containers/assembly.js

constructor- It contains x and y for interactjs.

Functions-

1. render-It contains interactjs drag drop snippet for the "cameradiv" and three divs for gray component ,yellow workspace component and camera .
2. renderlist - contains interactjs drag drop snippet for the images and drop zone is yellow div(canvas1).

Reducers-

1. objectselect-it contains all the selected items

Actions-

1. Camera_position - to store final dropped position of the camera.

4.8 src/Containers/logic.js

Functions-

1. render –local storage is used to save camera position and objects with their properties which then can be used in preview.js
2. renderlist - it will display all selected images in the horizontal scroll bar
3. properties - this will return 3 sections -> movable, static and miscellaneous
4. movable - this will return the movable properties .
5. statica - this will return the static properties

6. misc-for sound and bomb shooting.

7. preview_screen-to show sequence and link to preview screen.

Reducers-

1. object - it contains all the selected items

2. objprop - it contains currently selected object

3. camera_Pos - it contains the position of the camera

Actions-

1. obj_prop_select - it will select the clicked object

4.9 src/Containers/preview.js

When the preview link is clicked, this window will be opened in new tab and only the player will be shown.

4.10 src/Components/player.js

- When the play link is clicked , game window will be opened in a new tab. All the functions are written in the constructor.
- "Config" function is used to set up the game screen
- "Arcade" physics is used in the game to give properties to the objects.
- "Preload" function is used to load all the images before executing.
- "Create" function is used to set up the player and objects.
- "anims" function is used to change the frames.

4.11 src/action/index.js

Here all the action creators are written.

4.12 src/reducers.js

Inside this folder all the reducer files are there.

5 Bugs

1. In the concept window , you can select many items.But in assembly window this is not scrollable because if you put scroll then you won't be able to drag the item out of the elements section.
2. The images dropped into the canvas are same images which are displayed in the elements section.Solution-(Try to create a new object when it is dropped into canvas.Don't use the same image as of elements.Make a copy of that and then display it in canvas)
3. You have to press "ok" button to set the position to 0 again.
4. Positions are not accurate
5. You cannot give all the properties to all the objects. Properties are specific to specific objects(example you cannot give movable property to sun).

6 Future Scope

1. You can zoom in and out in the assembly. Before that you should get the positions properly.
2. You can add more properties to the objects and environment.
3. You can change the theme of the game(example underwater)
4. To create more sprites , you have to create frames of the object.
5. You can add the scroll bar to assembly window to extend the environment.
6. You can create different types of games.