Portfolio 2D Game

Theng Yuan Yuan

Title: Portfolio 2d game

Description: A portfolio works like a 2d game, like Nintendo games in 90s.

Framework: Vanilla

Language(s): JavaScript

Start Date: 5/13/2024

Done Date: -

Source: [Build a Dev Portfolio as a 2D Game – JavaScript Course (youtube.com)](https://www.youtube.com/watch?v=wy_fSStEgMs)

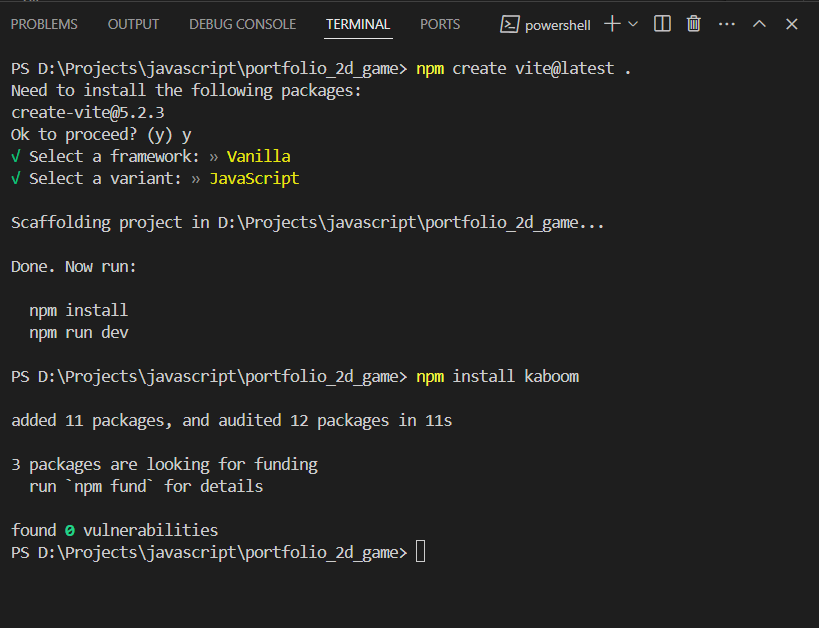
[](https://www.youtube.com/watch?v=wy_fSStEgMs)

# 1. Setup

## Create project

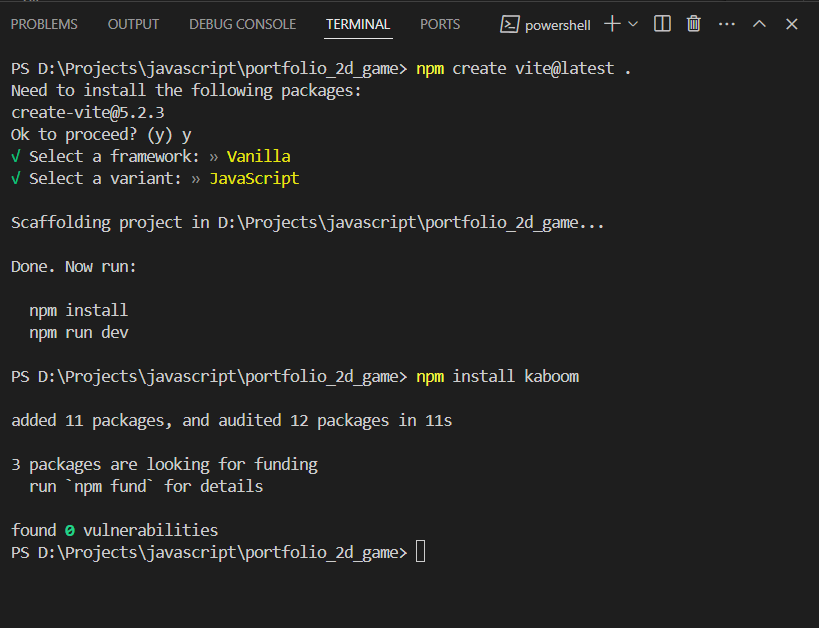
npm create vite@latest .

Vanilla framework (plain JavaScript without any additional libraries or frameworks).



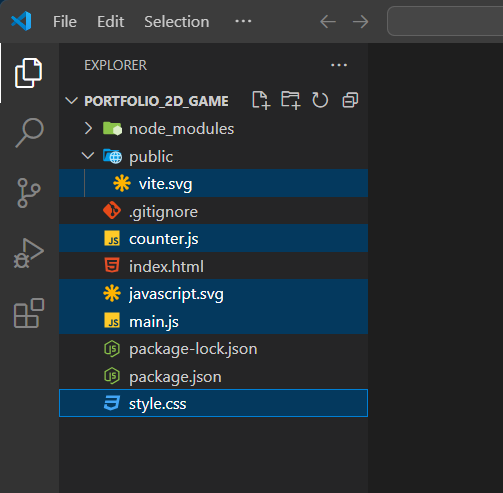
## Install package we need (kaboom)

We will use kaboom.js therefore: npm install kaboom



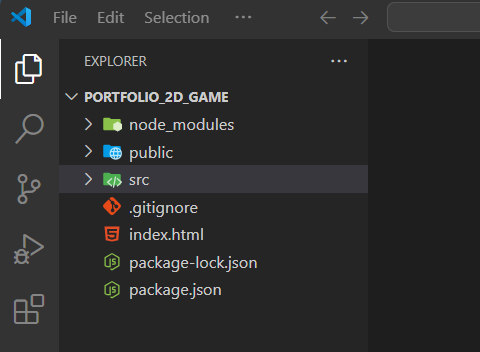
Kaboom.js is a JavaScript framework for making games quickly and easily.

## Remove unneeded files

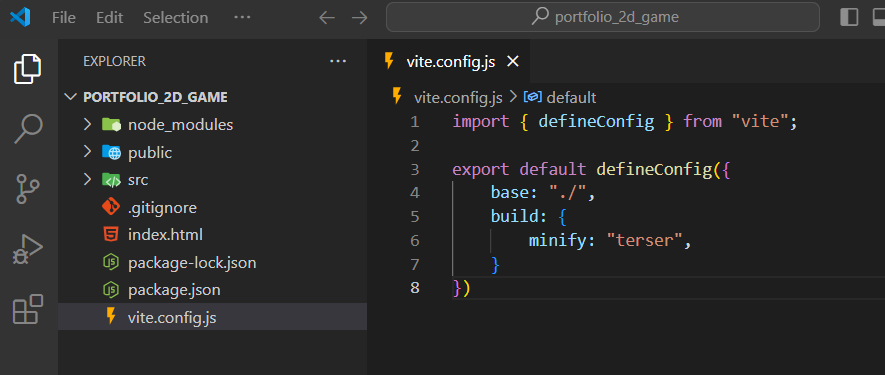


## Add src directory

src: store the source codes



## Add vite.config.js file



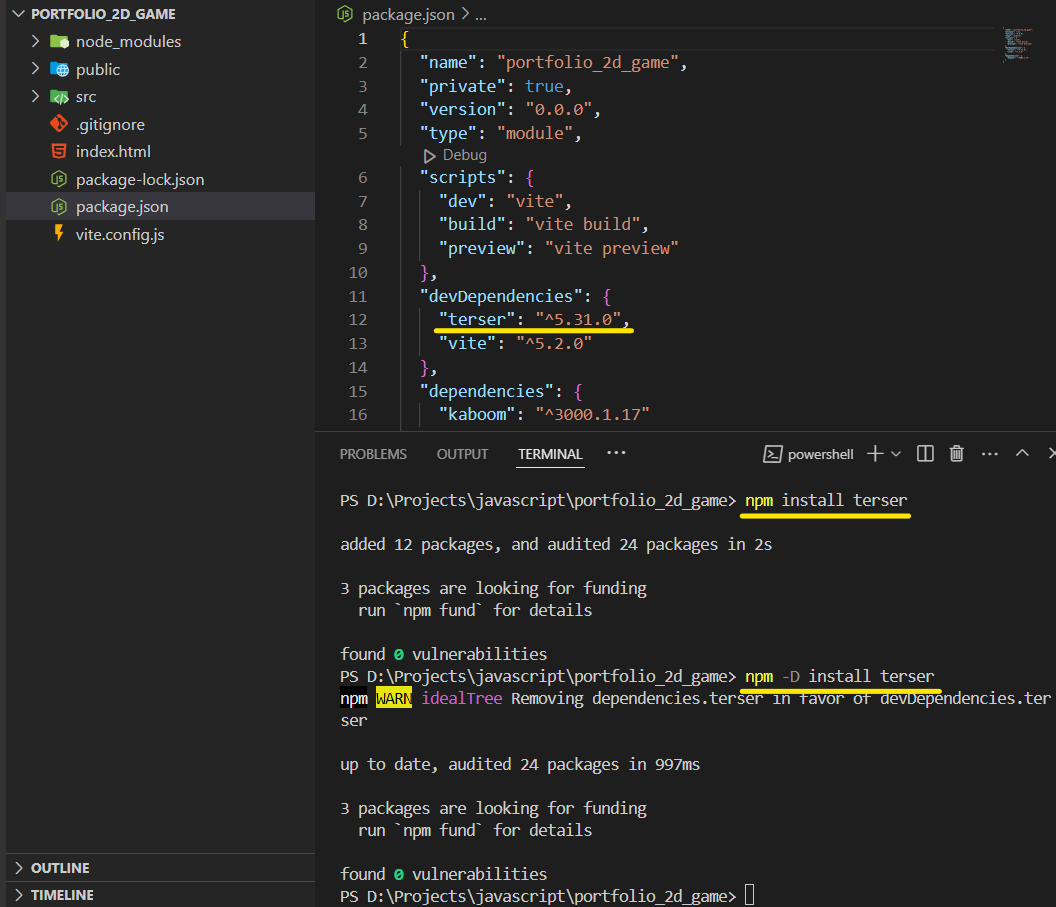
vite.config.js: export/compiles/transpile javascript to run in the browser, optimizations minifications to make the code compact and take least amount of space.

The minify used is “terser” instead of the default, because a bug with the way kaboom is made. If using default minify, the code doesn’t work. Therefore, the alternative “terser” is used. Here, “terser” will make the code smaller, take lesser space.

## Install terser in devDependencies

npm install terser

npm –D install terser



## Create main.js file in src

A screenshot of a computer

Description automatically generated

## Add some codes in index.html

A screen shot of a computer program

Description automatically generated

Style can be in separate folder, but in this project, the author mentioned it is not css-heavy.

Change the src to “src/main.js” as we have changed the path.

## kaboomCtx.js

kaboom context dot js

A screen shot of a computer

Description automatically generated

import kaboom function from kaboom.js.

call the kaboom function to initialize the game and configure it with specific options.

Some explanation from chatgpt:

* **global: false**
  + By default, Kaboom attaches its functions (like **add**, **sprite**, **pos**, etc.) to the global scope so that you can use them anywhere in your code without needing to import them explicitly. However, setting **global** to **false** prevents this global attachment. Instead, you will use the instance **k** to access these functions. This is useful for avoiding global namespace pollution and keeping your code more modular and organized.
* **touchToMouse: true**
  + This option allows touch events (like taps on a touchscreen) to be translated into mouse events. This is particularly useful for making your game playable on both desktop and mobile devices without needing to write separate handling code for touch events.
* **canvas: document.getElementById("game")**
  + This option specifies the HTML **<canvas>** element where the game will be rendered. By using **document.getElementById("game")**, you are telling Kaboom to render the game inside the canvas element with the ID **"game"**. This allows you to control exactly where on your webpage the game appears.

This is example of how to use the global k in the project later.

A black screen with white text

Description automatically generated

Extra: Without Explicit k Initialization

A screen shot of a computer program

Description automatically generated

## Create these files

constants.js, utils.js in src directory

A screenshot of a computer

Description automatically generated

constants.js holds constants information/values (naming, scaling, specific dialogues in the game).

utils.js holds functions to make the code look nicer (? I don’t understand what the author mean)

## assets (fonts/images)

Add the font (monogram.tff) and the image (spritesheet.png) in public directory.

[monogram by datagoblin (itch.io)](https://datagoblin.itch.io/monogram)

[2d-portfolio-kaboom/public/spritesheet.png at master · JSLegendDev/2d-portfolio-kaboom (github.com)](https://github.com/JSLegendDev/2d-portfolio-kaboom/blob/master/public/spritesheet.png)

A screenshot of a computer

Description automatically generated

The spritesheet.png from Author:

A screenshot of a video game

Description automatically generated

## Load asset (loadSprite function)

A screen shot of a computer program

Description automatically generated

sliceX – The total number of frames in x-axis

sliceY – The total number of frames in y-axis

anims – specify the frame of the moving animations

Explanation from chatgpt:

* **k.loadSprite Function:**
  + This function is used to load a spritesheet and define how it should be sliced and animated.
  + **"spritesheet"** is the name you assign to this sprite. You can reference this name later when adding the sprite to the game.
  + **"./spritesheet.png"** is the path to the image file that contains the spritesheet.
* **Options Object:**
  + **sliceX: 39**: This means the spritesheet is divided into 39 columns.
  + **sliceY: 31**: This means the spritesheet is divided into 31 rows.
* **Animations:**
  + **anims** is an object that defines various animations using the frames from the spritesheet.
  + Each key in the **anims** object represents an animation name (e.g., **"idle-down"**, **"walk-down"**).
  + The value can be a single frame number or an object that defines a frame range for the animation.
  + Example of a single frame animation:
    - **"idle-down": 936**: This animation consists of just one frame at index 936.
  + Example of a frame range animation:
    - **"walk-down": { from: 936, to: 939, loop: true, speed: 8 }**: This animation plays frames from 936 to 939, loops continuously, and plays at 8 frames per second.

# Learn to use Tiled Software

## new map

New map, set the map size and tile size.

A screenshot of a map

Description automatically generated

## new tile set

Add new tile set, import the tile set author has provided.  
A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Save this map in json file in the public folder

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

# Project development: display map

## add map.json and map.png

<https://github.com/JSLegendDev/2d-portfolio-kaboom/tree/master/public>

The author has drawn the map. Copy the map.json and map.png into public folder.

A screenshot of a computer

Description automatically generated

## load sprite(map) and background

A screen shot of a computer program

Description automatically generated

* **Loading a Sprite:** Use **k.loadSprite** to load a sprite for the game, associating it with a name for later reference.
* **Setting Background Color:** Utilize **k.setBackground** to set the background color of the game canvas, providing a color value.

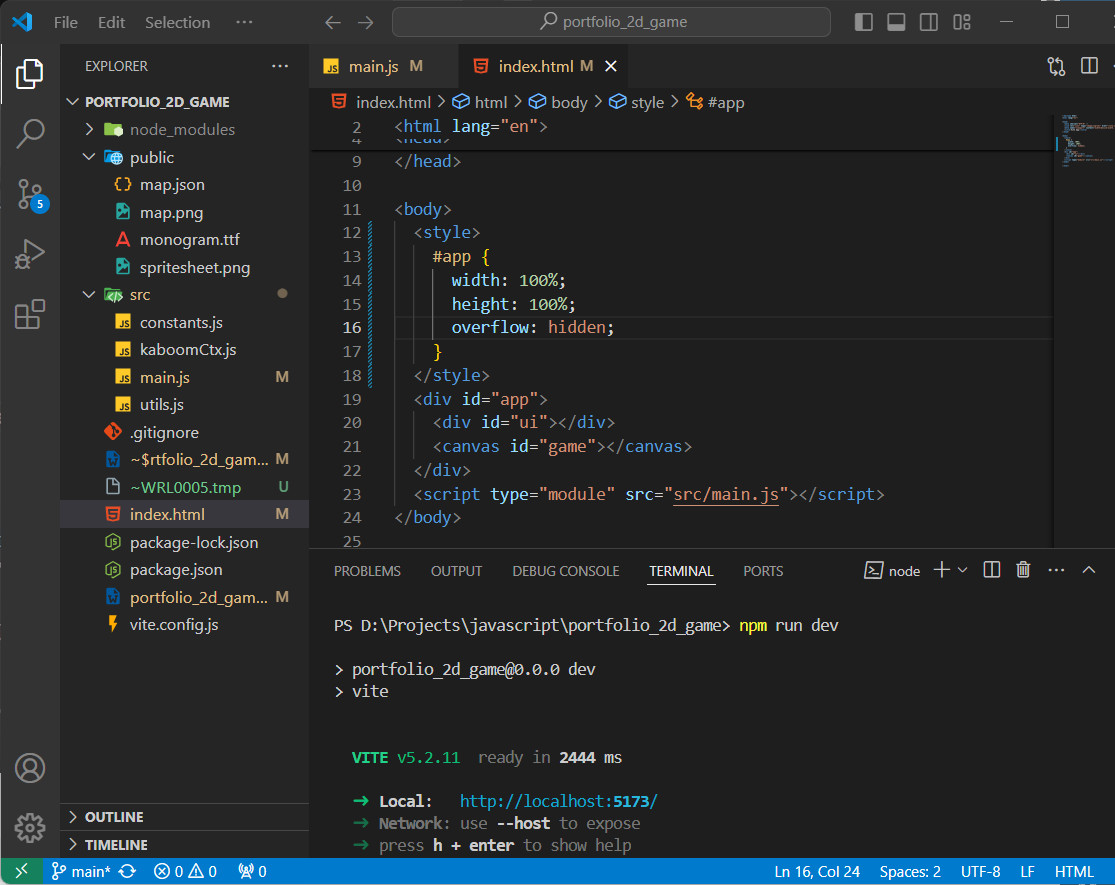
## load in browser.

npm run dev

A screen shot of a computer

Description automatically generated

## Set style in index.html



set 100% for width and height of the <div> with id “app”

## map display, create map object

A screenshot of a computer screen

Description automatically generated

Explanation from chatgpt

k.scene("main", async () => {

// Scene logic goes here

});

k.scene define a Scene “main”

asynchronous function (using **async**) to handle asynchronous operations.

const mapData = await (await fetch("./map.json")).json();

asynchronously fetches map data from "map.json" using the **fetch** API.

fetches the file using **fetch**

extracts JSON data from the response using **.json()**.

const layers = mapData.layers;

This extracts the "layers" property from the fetched map data, include background layers, foreground layers, collision layers, etc.

const map = k.make([

k.sprite("map"),

k.pos(0),

k.scale(scaleFactor),

]);

This creates a map object using k.make. It includes:

k.sprite set A sprite named "map" (loaded earlier with loadSprite).

k.pos set the position at (0, 0).

k.scale sets the scaling factor, “scaleFactor” is defined in constants.js as below).

A screen shot of a computer

Description automatically generated

## create player object

A screenshot of a computer program

Description automatically generated

This creates a player object using **k.make**, it includes:

k.sprite defines the appearance of the player and initial animation (idle-down)

k.area defines the area of the player that can interact with the game environment.

rectangle with a width and height of 10 units, (0, 3) relative to the player's position.

k.body adds physics to the player, allowing it to interact with the game world based on physical properties like gravity, collision, etc.

k.anchor sets the anchor point of the player sprite to its center

k.pos sets the initial position of the player. (no params = 0,0)

k.scale scales the player sprite

custom data to the player object. It includes properties like **speed**, **direction**, and **isInDialogue**, which can be used to control player behavior and state.

“player” is the tag to the player object, used to identify and manipulate it within the game.

## Logic to display

A screen shot of a computer program

Description automatically generated

A screen shot of a computer

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

The rest is not going to be explained.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generated

A screen shot of a computer program

Description automatically generatedA screenshot of a computer screen

Description automatically generated

A screen shot of a computer program

Description automatically generated