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)

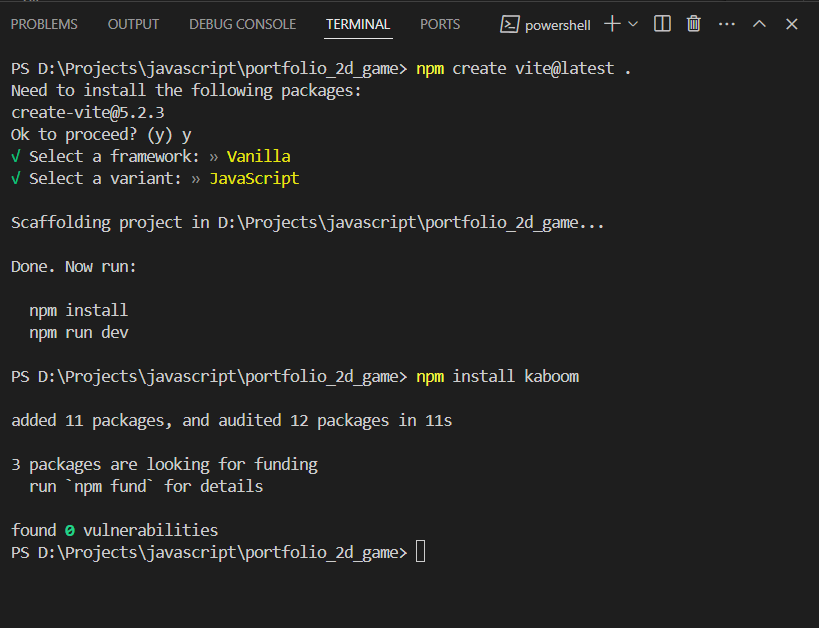
[](https://www.youtube.com/watch?v=wy_fSStEgMs)

# Steps 1: Setup

## Create project

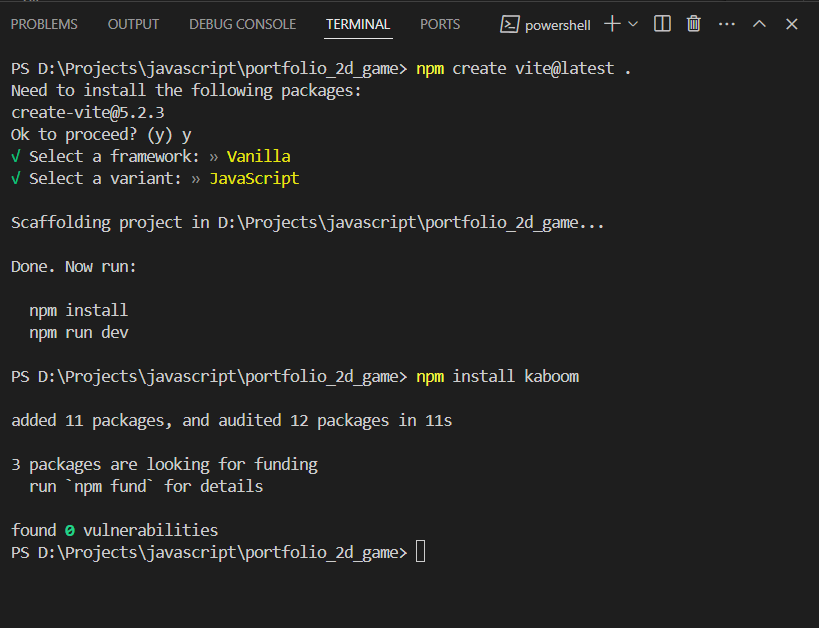
npm create vite@latest .

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



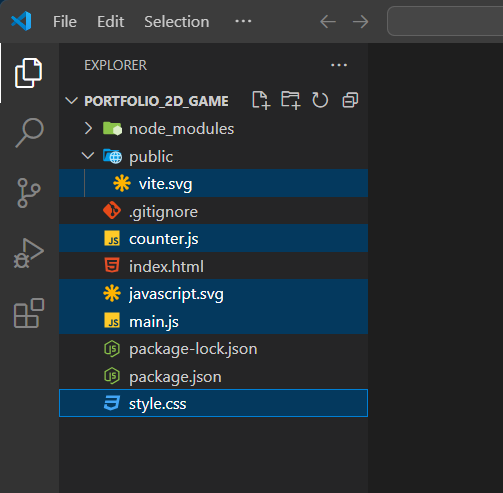
## Install package we need (kaboom)

We will use kaboom.js therefore: npm install kaboom



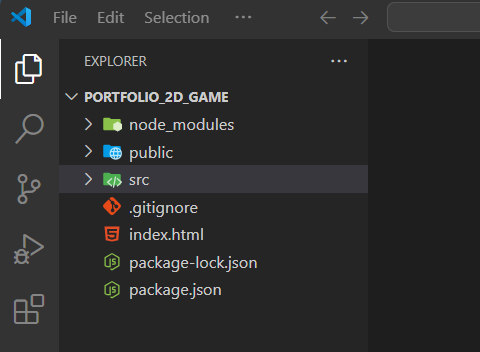
Kaboom.js is a JavaScript framework for making games quickly and easily.

## Remove unneeded files

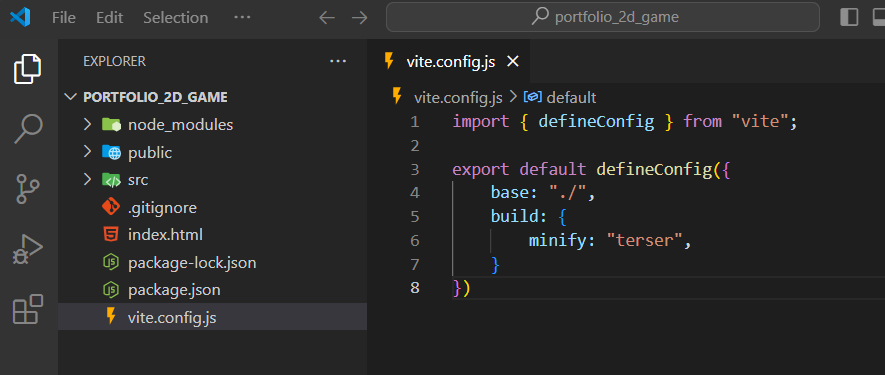


## Add src directory

src: store the source codes



## Add vite.config.js file



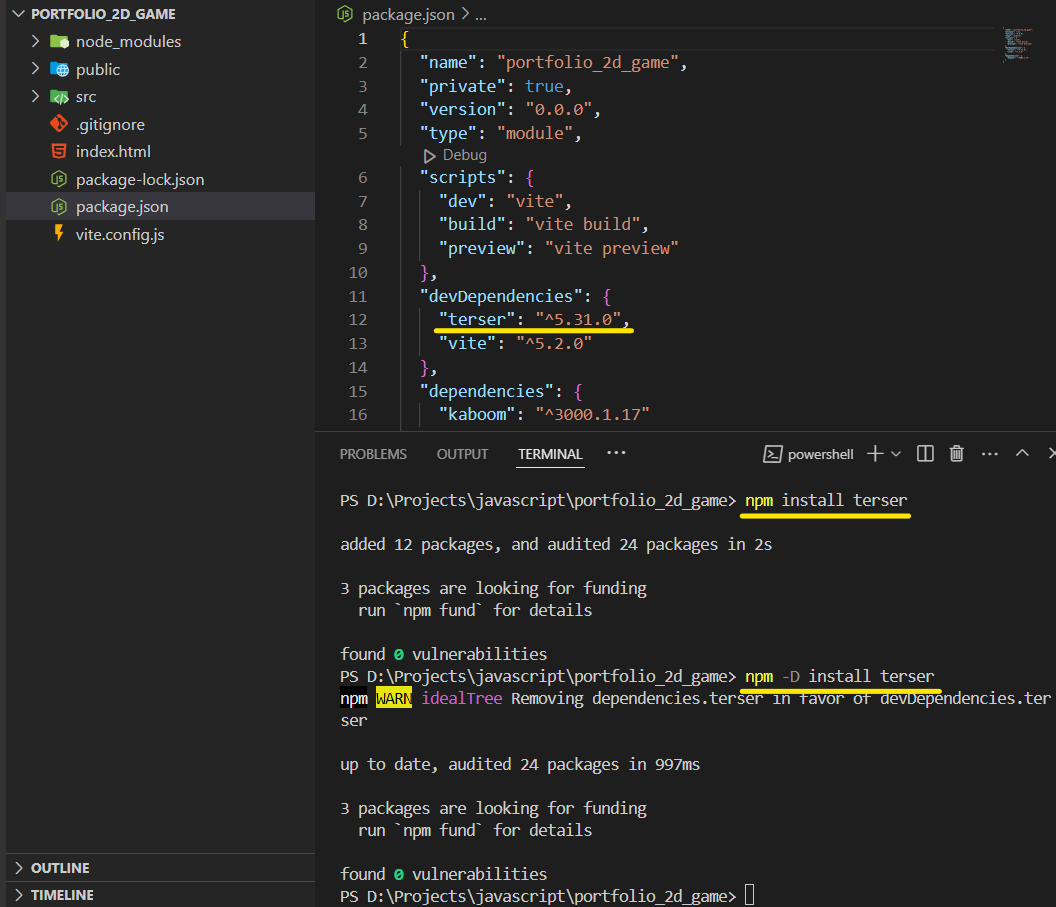
vite.config.js: export/compiles/transpile javascript to run in the browser, optimizations minifications to make the code compact and take least amount of space.

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

# Start Code: Load Assets