Credit Sussie - Coding Challenge

- ## Prerequisites

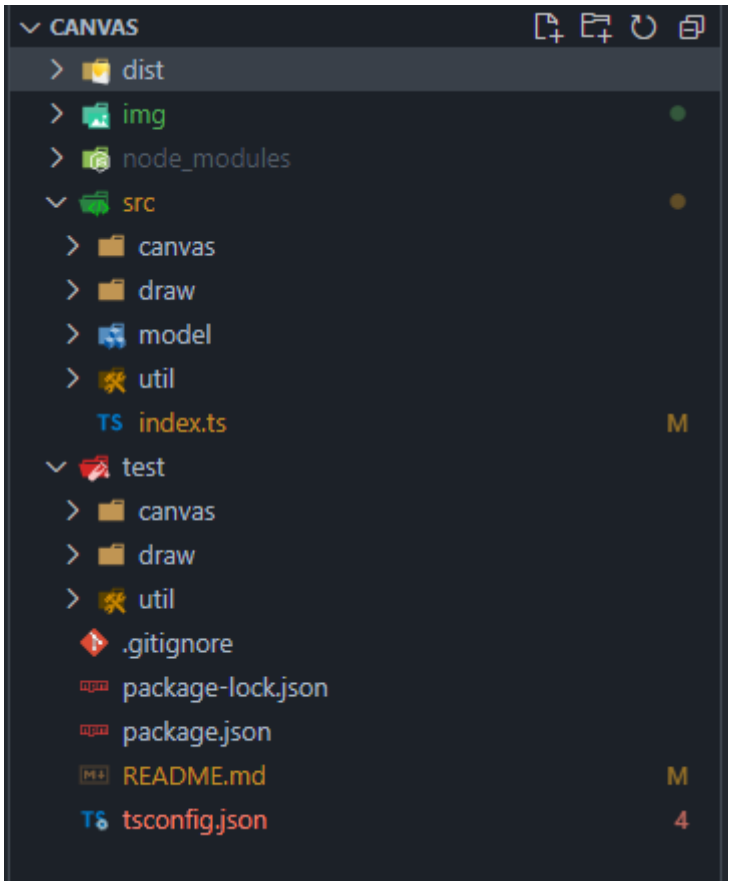
```
* Node.js
```

- ## Project structure

`dist`: Contains the transpiled JavaScript files

`src`: TypeScript files

`test`: Unit-Test files



- ## Initializing the project

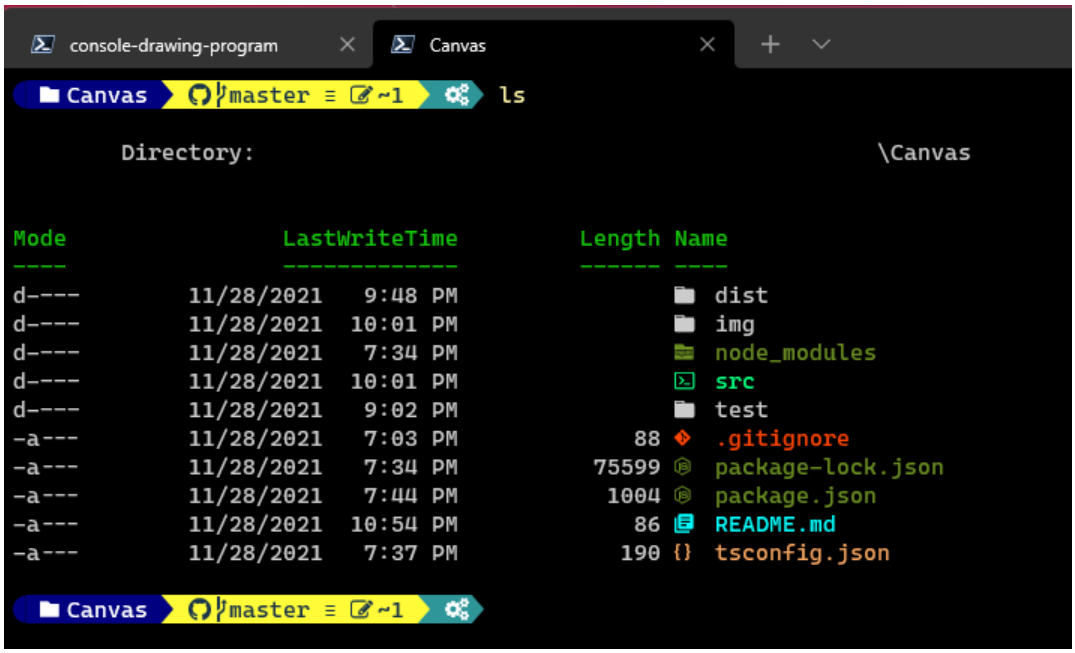
Project execution starts with `init()` function execution

```
  drawNewRectangle,
  fillPalette,
} from "./canvas/palette";

let sessionPaletteData;

init();

function init(): void {
  const command: string = question(Prompt.MSG_ENTER_COMMAND);
  const input: UserInputConfig = validator(command);

  handler(input);
  console.log("\n");

  init();
}

function handler(input: UserInputConfig): void {
  // console.log(input);
  // console.log("\n");

  // Invalid command
  if (input.command == Key.INVALID_COMMAND) {
    console.log(Prompt.MSG_INVALID_COMMAND);
    return;
  }

  // Canvas creation
  if (input.command == Key.DRAW_NEW_CANVAS) {
    sessionPaletteData = createPalette(input);
    if (sessionPaletteData == null) {
      console.log(Prompt.MSG_CANVAS_CREATION_FAILED);
    } else {
      drawInPalette(sessionPaletteData);
    }
    return;
  }

  // Draw a line
  if (input.command == Key.DRAW_LINE && isPaletteSessionAvailable()) {
    let _data = drawNewLine(input, sessionPaletteData);
    if (_data == null) {
      console.log(Prompt.MSG_INVALID_LINE_COORDINATES);
    } else {
      sessionPaletteData = _data;
      drawInPalette(sessionPaletteData);
    }
    return;
  }

  // Draw a rectangle
  if (input.command == Key.DRAW_RECTANGLE && isPaletteSessionAvailable()) {
    let _data = drawNewRectangle(input, sessionPaletteData);
    if (_data == null) {
      console.log(Prompt.MSG_INVALID_LINE_COORDINATES);
    } else {
      sessionPaletteData = _data;
      drawInPalette(sessionPaletteData);
    }
    return;
  }

  // Fill the canvas with colour
  if (input.command == Key.BUCKET_FILL && isPaletteSessionAvailable()) {
    let _data = fillPalette(input, sessionPaletteData);
    if (_data == null) {
      console.log(Prompt.MSG_FILL_FAILED);
    } else {
      sessionPaletteData = _data;
      drawInPalette(sessionPaletteData);
    }
    return;
  }

```

```
84    // Exit command issued
85    if (input.command == Key.QUIT) {
86      console.log(Prompt.MSG_QUIT);
87      process.exit();
88    }
89  }
90
91  function isPaletteSessionAvailable() {
92    if (sessionPaletteData) {
93      return true;
94    }
95
96    console.log(Prompt.MSG_NO_PALETTE);
97    return false;
98  }
```

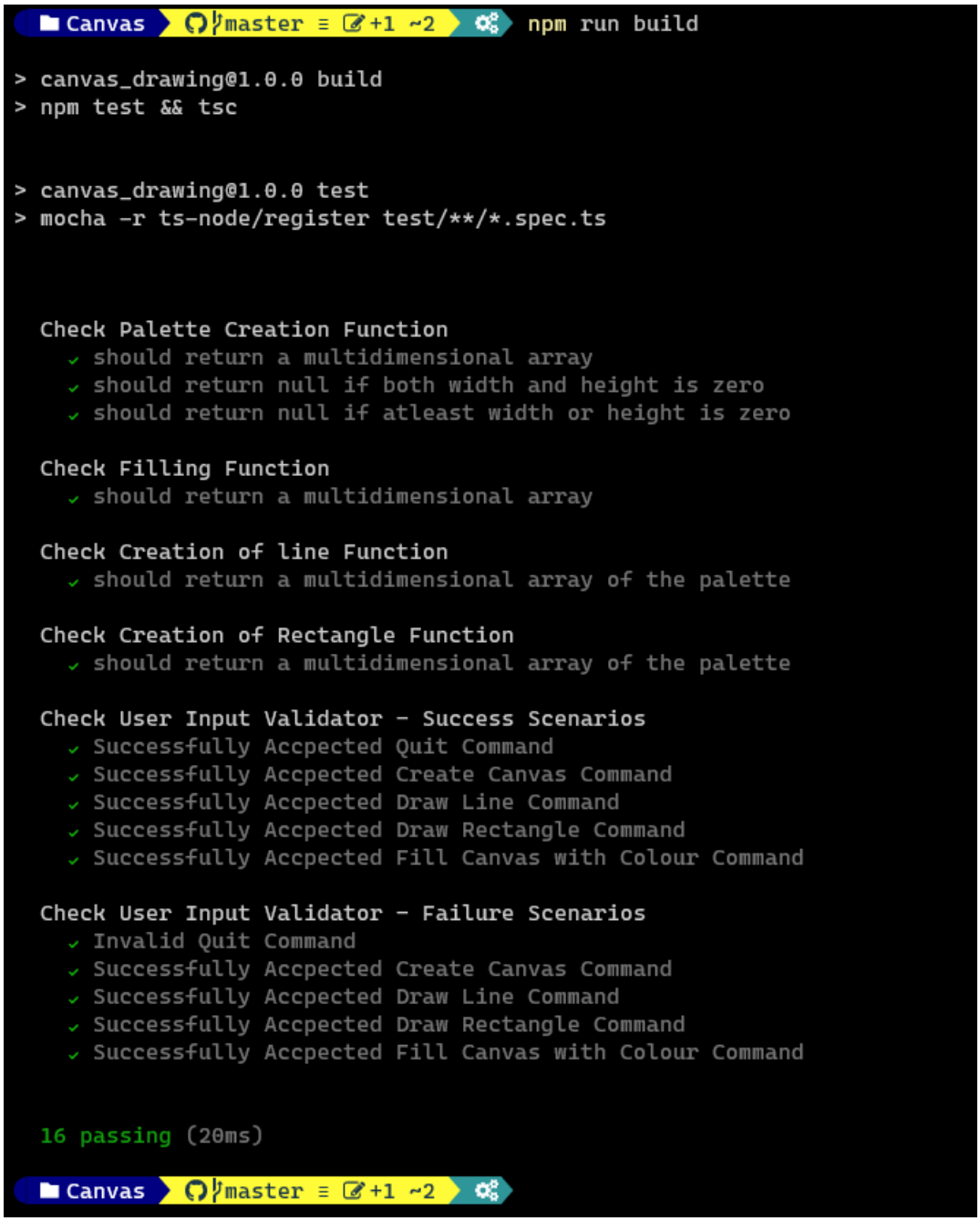
- ## Execution

1. Navigate to project folder



2. To build:

```
> npm run build
```



3. To run the Unit-Tests:

```
> npm test
```

```
Canvas  master ≡ +1 ~2  npm test

> canvas_drawing@1.0.0 test
> mocha -r ts-node/register test/**/*.spec.ts



  Check Palette Creation Function
    ✓ should return a multidimensional array
    ✓ should return null if both width and height is zero
    ✓ should return null if atleast width or height is zero

  Check Filling Function
    ✓ should return a multidimensional array

  Check Creation of line Function
    ✓ should return a multidimensional array of the palette

  Check Creation of Rectangle Function
    ✓ should return a multidimensional array of the palette

  Check User Input Validator - Success Scenarios
    ✓ Successfully Accpected Quit Command
    ✓ Successfully Accpected Create Canvas Command
    ✓ Successfully Accpected Draw Line Command
    ✓ Successfully Accpected Draw Rectangle Command
    ✓ Successfully Accpected Fill Canvas with Colour Command

  Check User Input Validator - Failure Scenarios
    ✓ Invalid Quit Command
    ✓ Successfully Accpected Create Canvas Command
    ✓ Successfully Accpected Draw Line Command
    ✓ Successfully Accpected Draw Rectangle Command
    ✓ Successfully Accpected Fill Canvas with Colour Command


  16 passing (20ms)

Canvas  master ≡ +1 ~2
```

4. Enter following command

```
> npm start
```

5. Enter commands to prompt

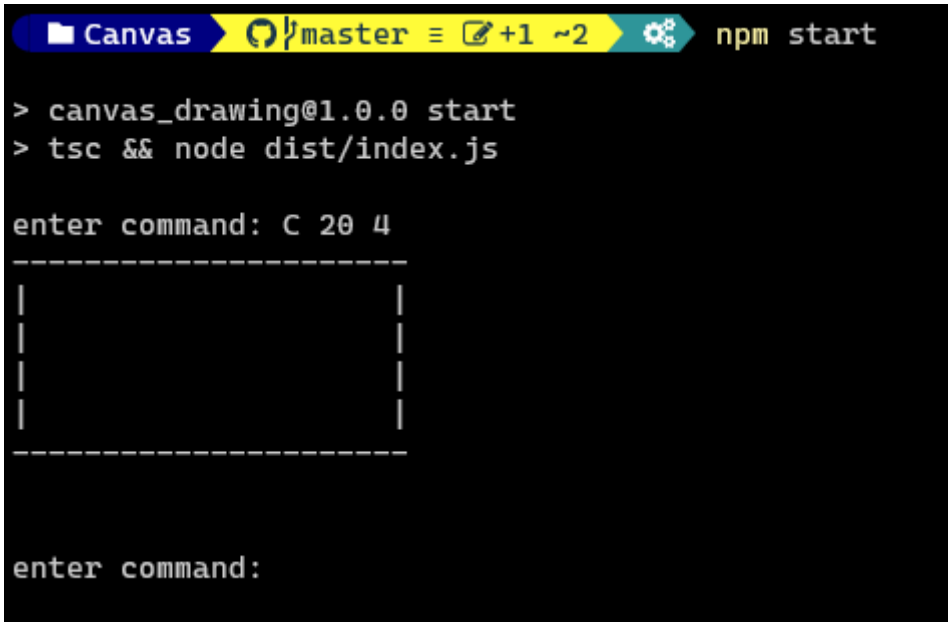
```
Canvas  master ≡ ~1  npm start

> canvas_drawing@1.0.0 start
> tsc && node dist/index.js

enter command:
```

6. To create a canvas:

```
> C 20 4
```

Width: 20

Heigth: 4



7. To draw a line:

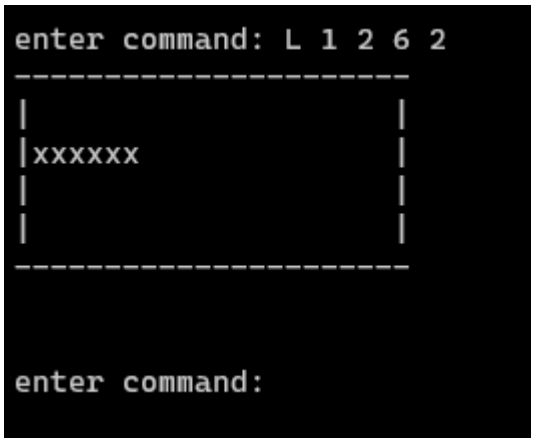
`> L 1 2 6 2`

x1: 1

y1: 2

x2: 6

y2: 2



8. To draw a rectangle:

`> R 14 1 18 3`

x1: 14

y1: 1

x2: 18

y2: 3

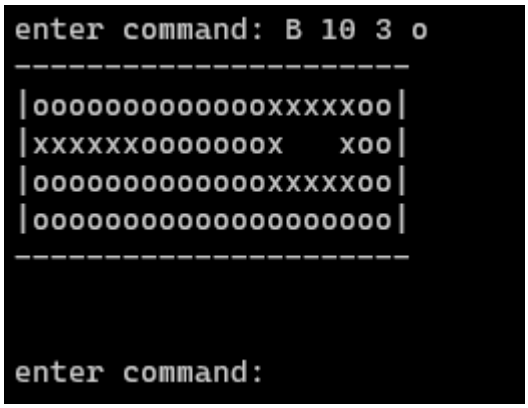
9. To fill the canvas with colour:

`> B 10 3 o`

x1: 10

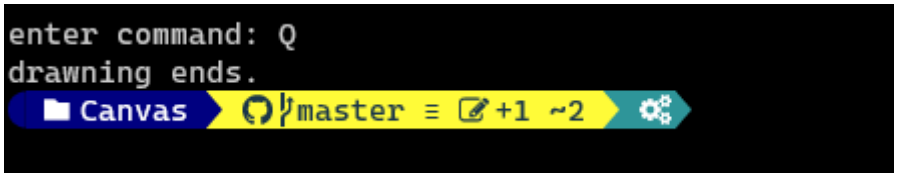
y1: 3

c: o



10. To quit:

`> Q`



Repository