

# How to play with DMN

by Guilherme Carreiro



github/karreiro

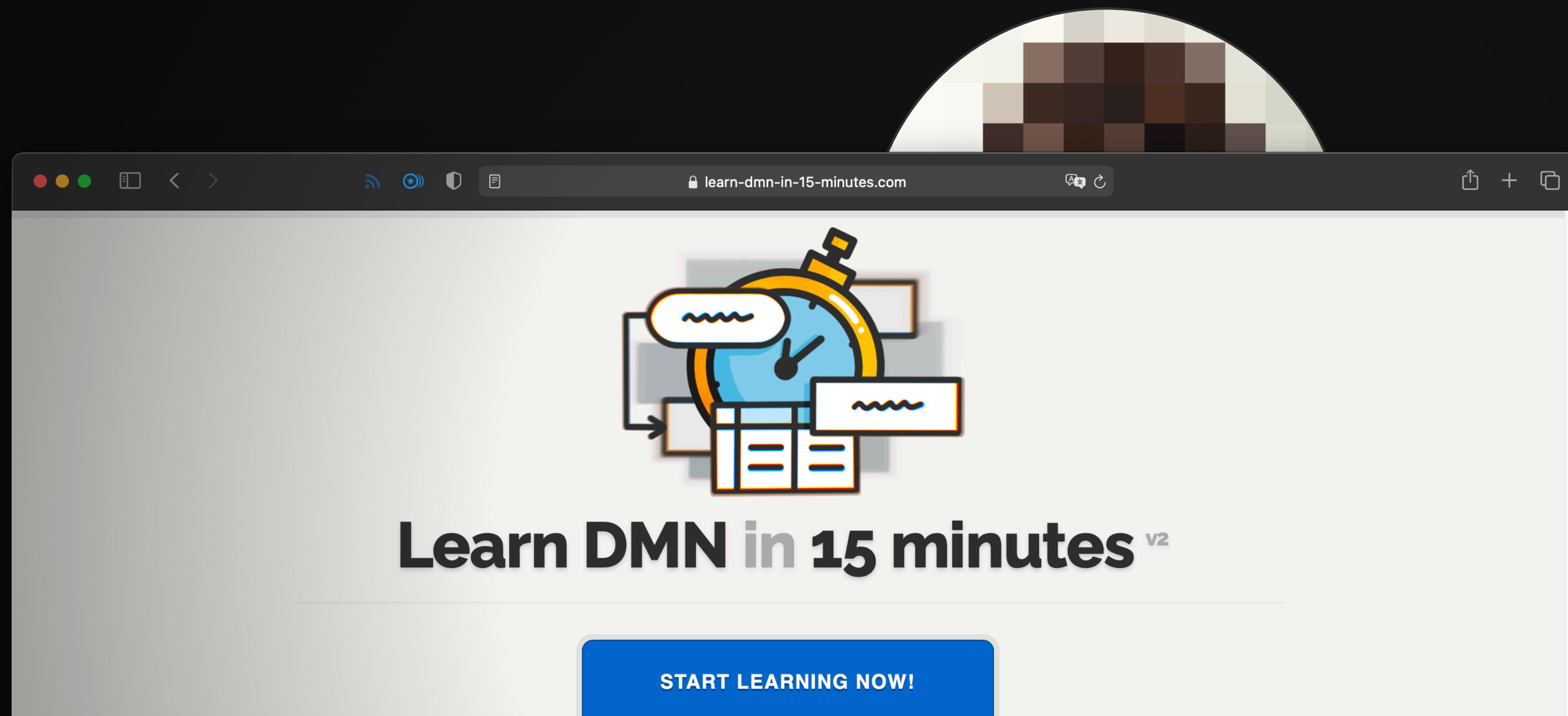
[karreiro.com](http://karreiro.com)

[karreiro.com/wiki](http://karreiro.com/wiki)

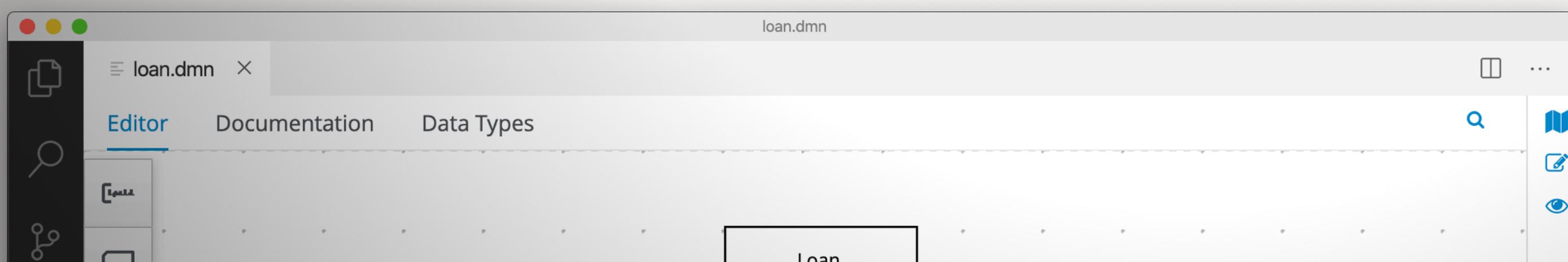
[karreiro.com/talks](http://karreiro.com/talks)

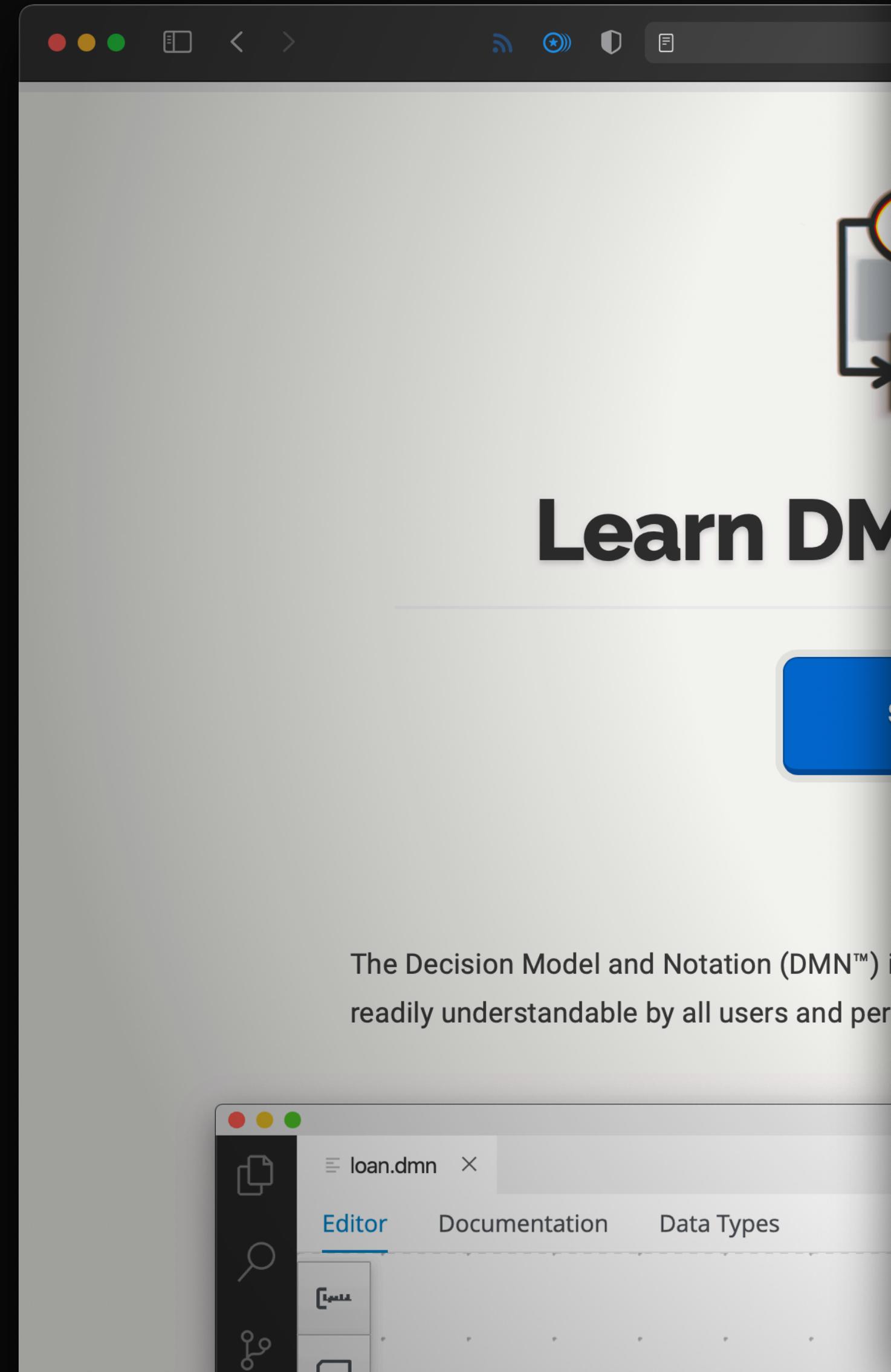
DMN tooling lead

[blog.kie.org](http://blog.kie.org)



The Decision Model and Notation (DMN™) is a Standard by [OMG®](#), providing a conventional and visual notation readily understandable by all users and personas.





dmn-cli.com

Home Installation Usage Report issues Github

# dmn-cli

A command-line interface tool which allows you to easily manage, inspect, and preview DMN models

```
~ $ dmn-cli inspect ~/my-dmn-model.dmn
- Version: DMN 1.2
- DMN nodes:
  L Should be suspended? (Decision)
  L Fine (Decision)
  L Violation (Input Data)
  L Driver (Input Data)
- DMN data types:
  L tDriver (Structure)
    L Name (String)
    L Points (Number)
  L tFine (Structure)
    L Amount (Number)
    L Points (Number)
  L tViolation (Structure)
    L Type (String)
    L Actual Speed (Number)
    L Speed Limit (Number)
~ $ dmn-cli preview ~/my-dmn-model.dmn
Processing DMN model preview... [=====] (10/10)
~ $
```

The screenshot captures a Mac desktop environment with two open browser windows. The primary window in the foreground is a YouTube video titled "[KieLive#11] DMN for Developers, by Guilherme Carreiro". The video player interface includes a thumbnail of the speaker, a progress bar at 21:03 / 52:16, and standard video controls. To the right of the video player, a sidebar displays the text "easily" above a large blue owl logo with the letters "KIE" next to it. The background window is a web page from dmn-cli.com, featuring a dark theme with a sidebar on the left containing links such as "Installation", "Usage", "Report issues", and "Github". A vertical scroll bar is positioned on the right side of this window. The overall layout is characteristic of a Mac OS X desktop.



# How to create a **snake game** with DMN

```
let state = { ... };

let gameLoop = () => {
    state = executeDmn(state);
    paint(state);
}

setInterval(gameLoop, 40);
```

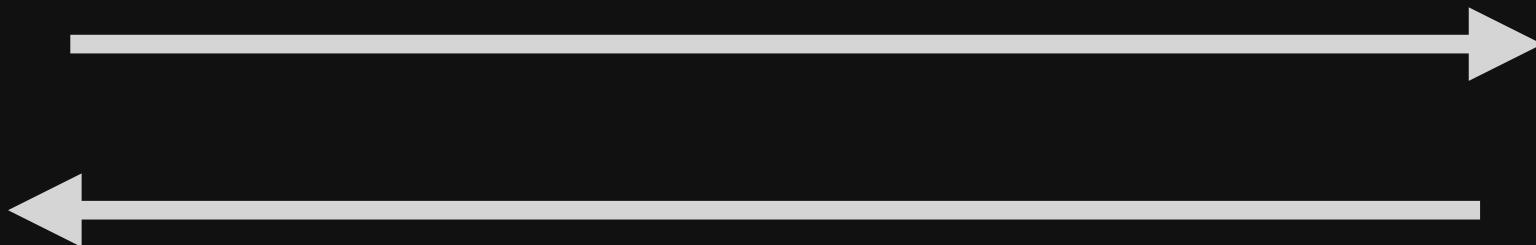
```
let state = { ... };

let gameLoop = () => {
    state = executeDmn(state);
    paint(state);
}

setInterval(gameLoop, 40);
```



Snake game client  
TypeScript



Snake game server  
DMN

```
let state = { ... };

let gameLoop = () => {
  state = executeDmn(state);
  paint(state);
}

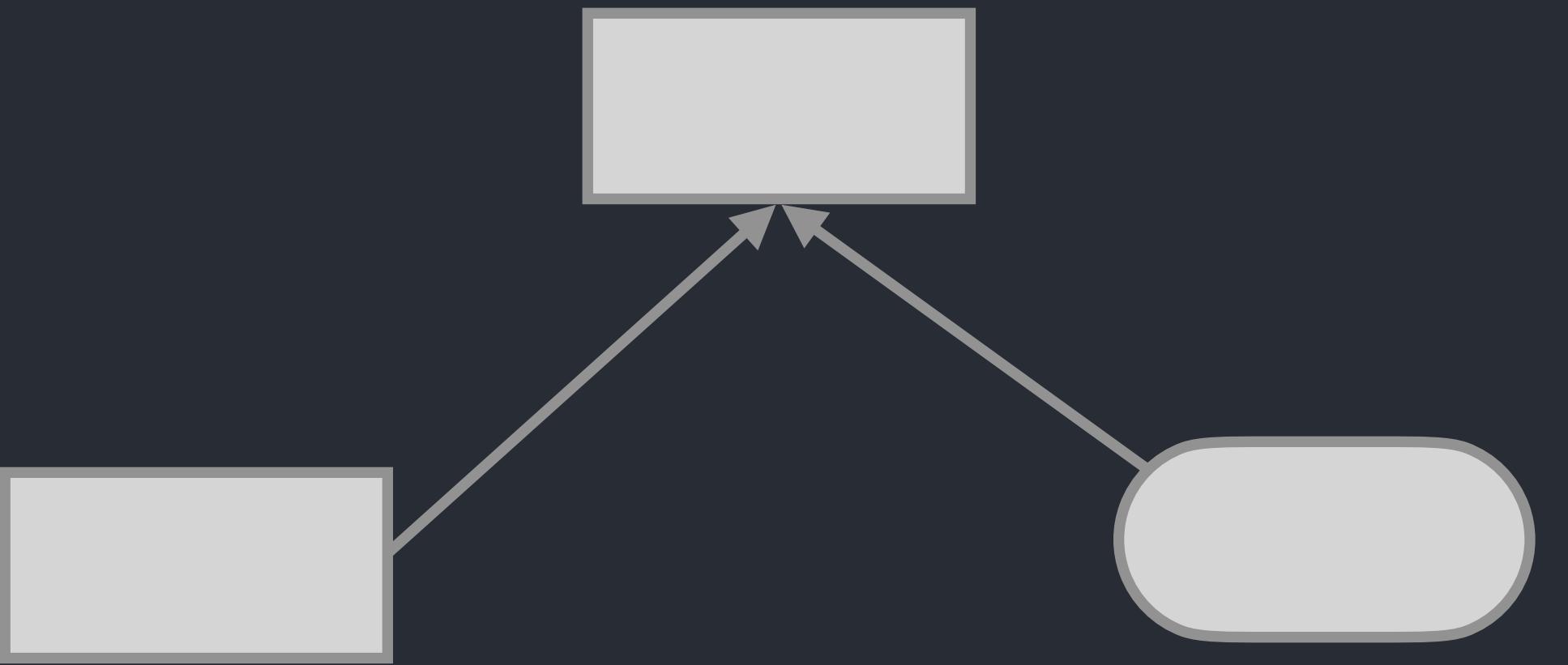
setInterval(gameLoop, 40);
```



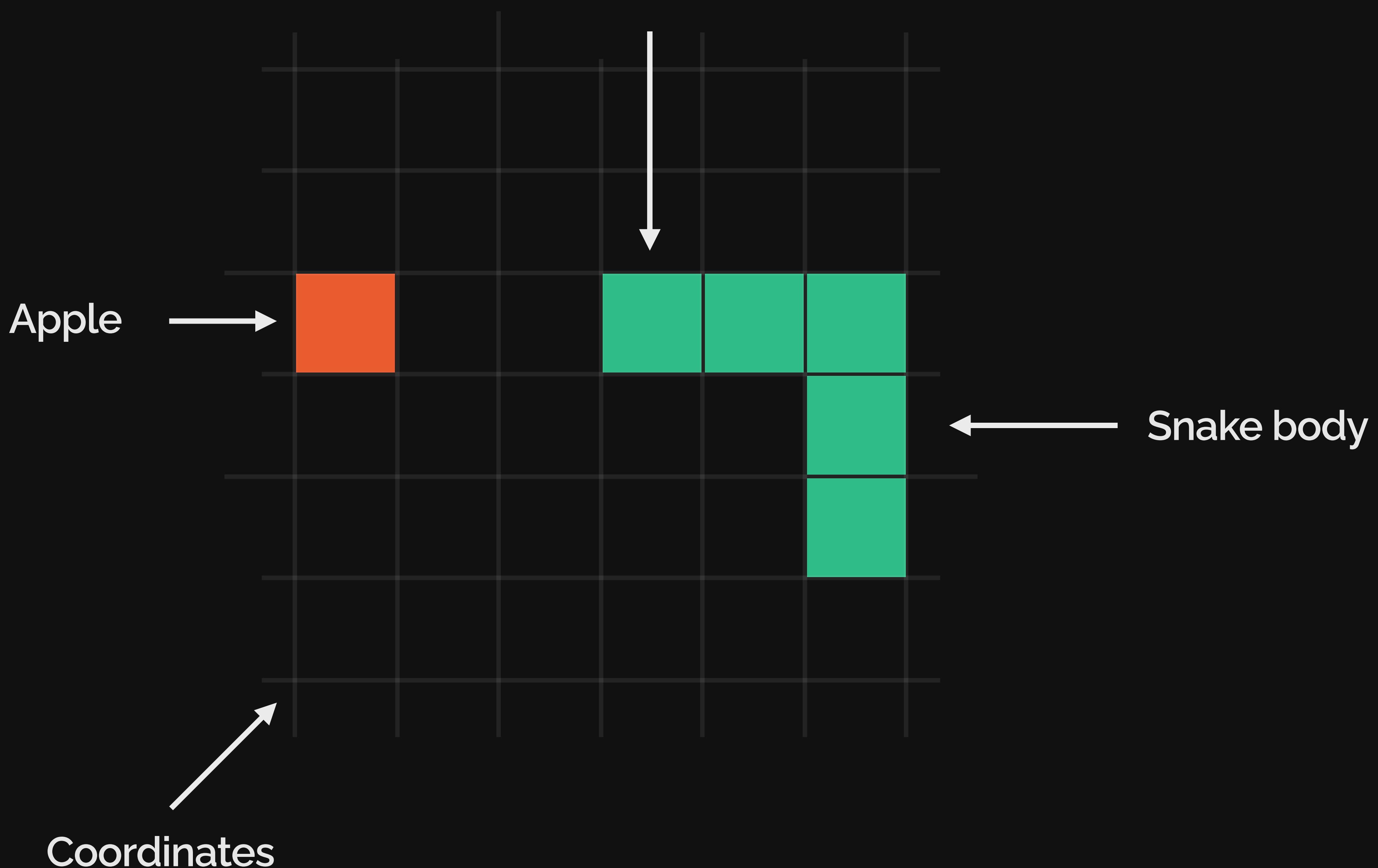
Snake game client  
TypeScript



Snake game server  
DMN



Snake head



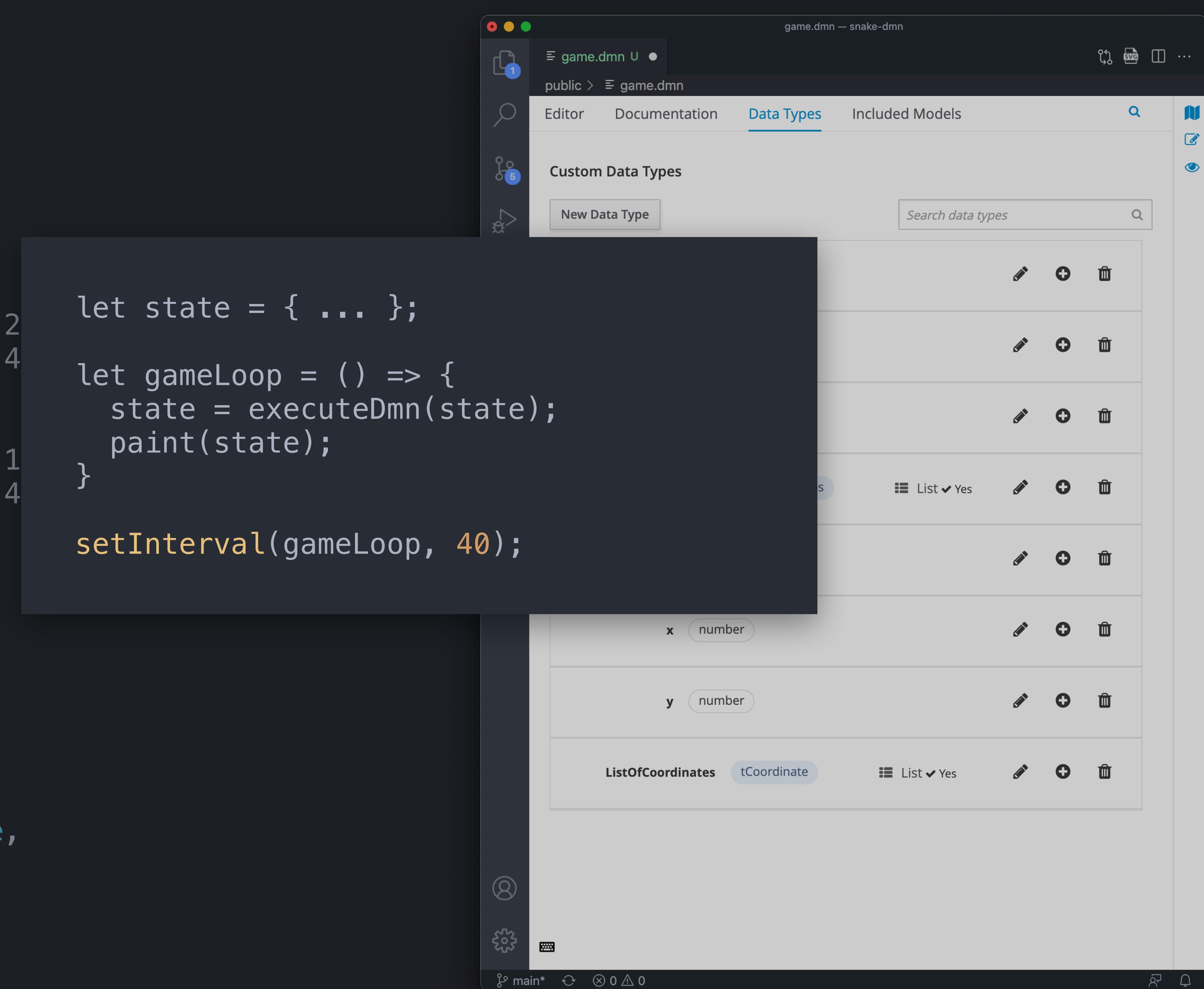
```
"state": {  
  
    "snake": {  
        "head": {  
            "x": 2,  
            "y": 4  
        },  
        "body": [  
            {  
                "x": 2,  
                "y": 4  
            },  
            {  
                "x": 1,  
                "y": 4  
            }  
        ]  
    },  
  
    "apple": {  
        "x": -1,  
        "y": -1  
    },  
  
    "gameOver": false,  
  
    "key": "Right",  
  
    "score": 0  
}
```

The screenshot shows a DMN editor interface with a dark theme. The title bar reads "game.dmn — snake-dmn". The left sidebar has icons for file, search, and other model elements, with a count of 5. The main area has tabs for "Editor", "Documentation", "Data Types" (which is selected), and "Included Models". A search bar says "Search data types".

The "Data Types" tab displays the following custom data types:

- tApple** (tCoordinate): A structure type with no details shown.
- tSnake** (Structure): A structure type with a "head" field (tCoordinate) and a "body" field (ListOfCoordinates). The "body" field is marked as a list and contains a checkmark for "Yes".
- head** (tCoordinate): A structure type with no details shown.
- body** (ListOfCoordinates): A list type containing tCoordinate elements. It is marked as a list and contains a checkmark for "Yes".
- tCoordinate** (Structure): A structure type with fields "x" (number) and "y" (number). Both are marked as numbers.
- ListOfCoordinates** (tCoordinate): A list type containing tCoordinate elements. It is marked as a list and contains a checkmark for "Yes".

```
"state": {  
  
    "snake": {  
        "head": {  
            "x": 2,  
            "y": 4  
        },  
        "body": [  
            {  
                "x": 2  
                "y": 4  
            },  
            {  
                "x": 1  
                "y": 4  
            }  
        ]  
    },  
  
    "apple": {  
        "x": -1,  
        "y": -1  
    },  
  
    "gameOver": false,  
  
    "key": "Right",  
  
    "score": 0  
}
```



# I. The infrastructure side

# The infrastructure side

```
const grid = 20;
const context = canvas.getContext("2d")!;

function paintSnake(snake: Snake) {
    context.fillStyle = "#2ebd88";

    [snake.head, ...snake.body].forEach((part) => {
        context.fillRect(part.x * grid, part.y * grid, grid, grid);
    });
}
```

# The infrastructure side

```
function setKey(event: KeyboardEvent) {  
    switch (event.key.toLowerCase()) {  
        case "arrowup":  
        case "w":  
            key = "Up";  
        case "arrowdown":  
        case "s":  
            key = "Down";  
        case "arrowleft":  
        case "a":  
            key = "Left";  
        case "arrowright":  
        case "d":  
            key = "Right";  
    }  
}  
  
window.addEventListener("keydown", (event) => setKey(event), false);
```

package.json — snake-dmn

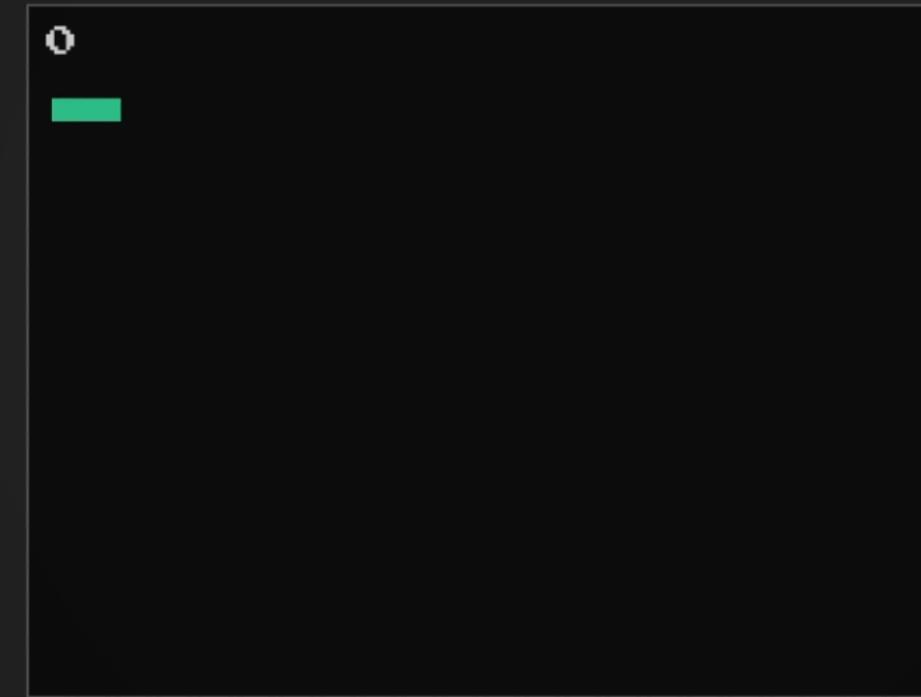
```
{ package.json } X
```

```
{} package.json > ...
```

```
12   "@types/react-dom": "^17.0.0",
13   "react": "^17.0.2",
14   "react-dom": "^17.0.2",
15   "react-scripts": "4.0.3",
16   "typescript": "^4.1.2",
17   "web-vitals": "^1.0.1"
18 },
19 > Debug
20 "scripts": {
21   "start-server": "java -jar server/jitexecutor-runner.jar",
22   "start-client": "react-scripts start",
23   "start": "concurrently \"yarn start-server\" \"yarn start-client\"",
24   "build": "react-scripts build",
25   "test": "react-scripts test",
26   "eject": "react-scripts eject"
27 },
28 "eslintConfig": {
29   "extends": [
30     "react-app",
31     "react-app/jest"
32   ],
33   "browserslist": {
34     "production": [
35       ">0.2%",
36       "not dead",
37       "not op_mini all"
38     ],
39     "development": [
40       "last 1 chrome version"
41     ]
42   }
43 }
```

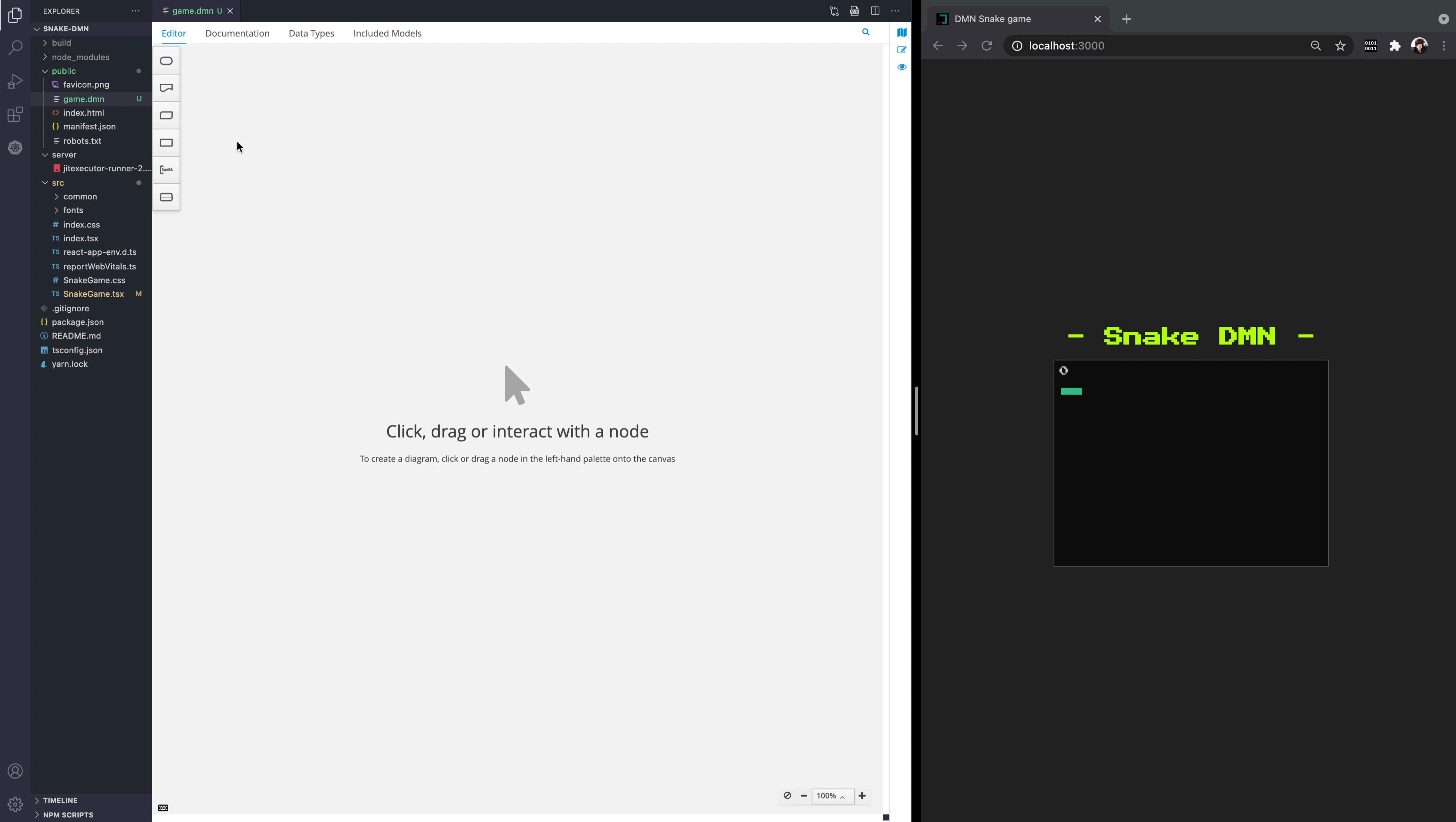
Ln 49, Col 1 Spaces: 2 UTF-8 LF JSON ✓ Prettier

```
"state": {  
  
  "snake": {  
    "head": {  
      "x": 2,  
      "y": 4  
    },  
    "body": [  
      {  
        "x": 2,  
        "y": 4  
      },  
      {  
        "x": 1,  
        "y": 4  
      }  
    ]  
  },  
  
  "apple": {  
    "x": -1,  
    "y": -1  
  },  
  
  "gameOver": false,  
  
  "key": "Right",  
  
  "score": 0  
}
```



```
function paintSnake(snake: Snake) {  
  context.fillStyle = "#2ebd88";  
  
  [snake.head, ...snake.body].forEach(part =>  
    context.fillRect(part.x * grid,  
    part.y * grid));  
}  
  
function paintApple(apple: Apple) {  
  ...  
}  
  
function paintBlackBackground() {  
  ...  
}  
  
function paintScore() {  
  ...  
}
```

## II. The snake state



The screenshot displays a developer's workspace with three main components:

- EXPLORER** (Left): Shows a file tree for a project named "SNAKE-DMN". The tree includes a "public" folder containing "favicon.png", "game.dmn" (which is currently selected), "index.html", "manifest.json", and "robots.txt". It also contains a "server" folder with "jitexecutor-runner-2..." and a "src" folder with various files like "common", "fonts", "index.css", "index.tsx", "react-app-env.d.ts", "reportWebVitals.ts", "SnakeGame.css", and "SnakeGame.tsx". Other files shown include ".gitignore", "package.json", "README.md", "tsconfig.json", and "yarn.lock".
- game.dmn** (Top Center): A code editor window showing the DMN model "Snake (<Undefined>)". A tooltip "Select expression" is visible over the editor area. The interface includes tabs for "Editor", "Documentation", "Data Types", and "Included Models", along with icons for search, copy, and eye.
- DMN Snake game** (Right): A browser window titled "localhost:3000" displaying the "DMN Snake game". The page features a dark background with green text and graphics, including a title "- Snake DMN -" and a small green snake icon.

EXPLORER    ...    game.dmn U X

SNAKE-DMN

- > build
- > node\_modules
- << Back to snake-game

Editor Documentation Data Types Included Models

Snake (Context)

#	Snake (tSnake)			
1	new head (tCoordinate)	U	Key (string)	new head (tCoordinate)
			x (number)	y (number)
1	"Up"	iSnake.head.x	iSnake.head.y - 1	annotation-1
2	"Down"	iSnake.head.x	iSnake.head.y + 1	
3	"Right"	iSnake.head.x + 1	iSnake.head.y	
4	"Left"	iSnake.head.x - 1	iSnake.head.y	

2 new body  
(ListOfCoordinates)

remove(concatenate(iSnake.body, iSnake.head), 1)

<result>

```
{ head: new head, body: new body }
```

DMN Snake game    X    +

localhost:3000

- Snake DMN -

### III. The apple state

EXPLORER

1 SNAKE-DMN

> build  
> node\_modules  
public favicon.png game.dmn index.html manifest.json robots.txt  
server jitexecutor-runner-2...  
src common fonts # index.css TS index.tsx TS react-app-env.d.ts TS reportWebVitals.ts # SnakeGame.css TS SnakeGame.tsx M .gitignore {} package.json README.md tsconfig.json yarn.lock

Editor Documentation Data Types Included Models

game.dmn U

Back to snake-game

Snake (Context)

# Snake (tSnake)

1 new head (tCoordinate)

U	Key (string)	new head (tCoordinate)		annotation-1
		x (number)	y (number)	
1	"Up"	iSnake.head.x	iSnake.head.y - 1	
2	"Down"	iSnake.head.x	iSnake.head.y + 1	
3	"Right"	iSnake.head.x + 1	iSnake.head.y	
4	"Left"	iSnake.head.x - 1	iSnake.head.y	

2 new body (ListOfCoordinates)

remove(concatenate(iSnake.body, iSnake.head), 1)

<result>

```
{ head: new head, body: new body }
```

DMN Snake game

localhost:3000

- Snake DMN -

EXPLORER    ...    game.dmn U X

SNAKE-DMN

- > build
- > node\_modules
- << Back to snake-game
- Apple (Decision Table)
- |   | iApple.x = -1<br>(boolean) | Apple<br>(tApple) |               | annotation-1 |
|---|----------------------------|-------------------|---------------|--------------|
|   |                            | x<br>(number)     | y<br>(number) |              |
| 1 | true                       | 10                | 10            |              |
| 2 | false                      | iApple.x          | iApple.y      |              |
- U
- iApple.x = -1  
(boolean)
- Apple  
(tApple)
- x  
(number)
- y  
(number)
- annotation-1
- 1
- 2
- true
- false
- iApple.x
- iApple.y

Editor Documentation Data Types Included Models

Properties

Id: \_1C2FC3E2-D5B4-4CFB-9082-EABC12F9B4F1

Description:

Documentation Links: Add None

Name: Apple

Question:

Allowed Answers:

Information item

Data type: tApple

Background details

Font settings

DMN Snake game X +

localhost:3000

- Snake DMN -

Timeline NPM Scripts

EXPLORER

1 SNAKE-DMN game.dmn U

Editor Documentation Data Types Included Models

Back to snake-game

Apple (Decision Table)

U	iApple.x = -1 (boolean)	Apple (tApple)		annotation-1
		x (number)	y (number)	
1	true	10	10	
2	false	iApple.x	iApple.y	

build node\_modules public favicon.png game.dmn index.html manifest.json robots.txt server jitexecutor-runner-2... src common fonts index.css index.tsx react-app-env.d.ts reportWebVitals.ts SnakeGame.css SnakeGame.tsx .gitignore package.json README.md tsconfig.json yarn.lock

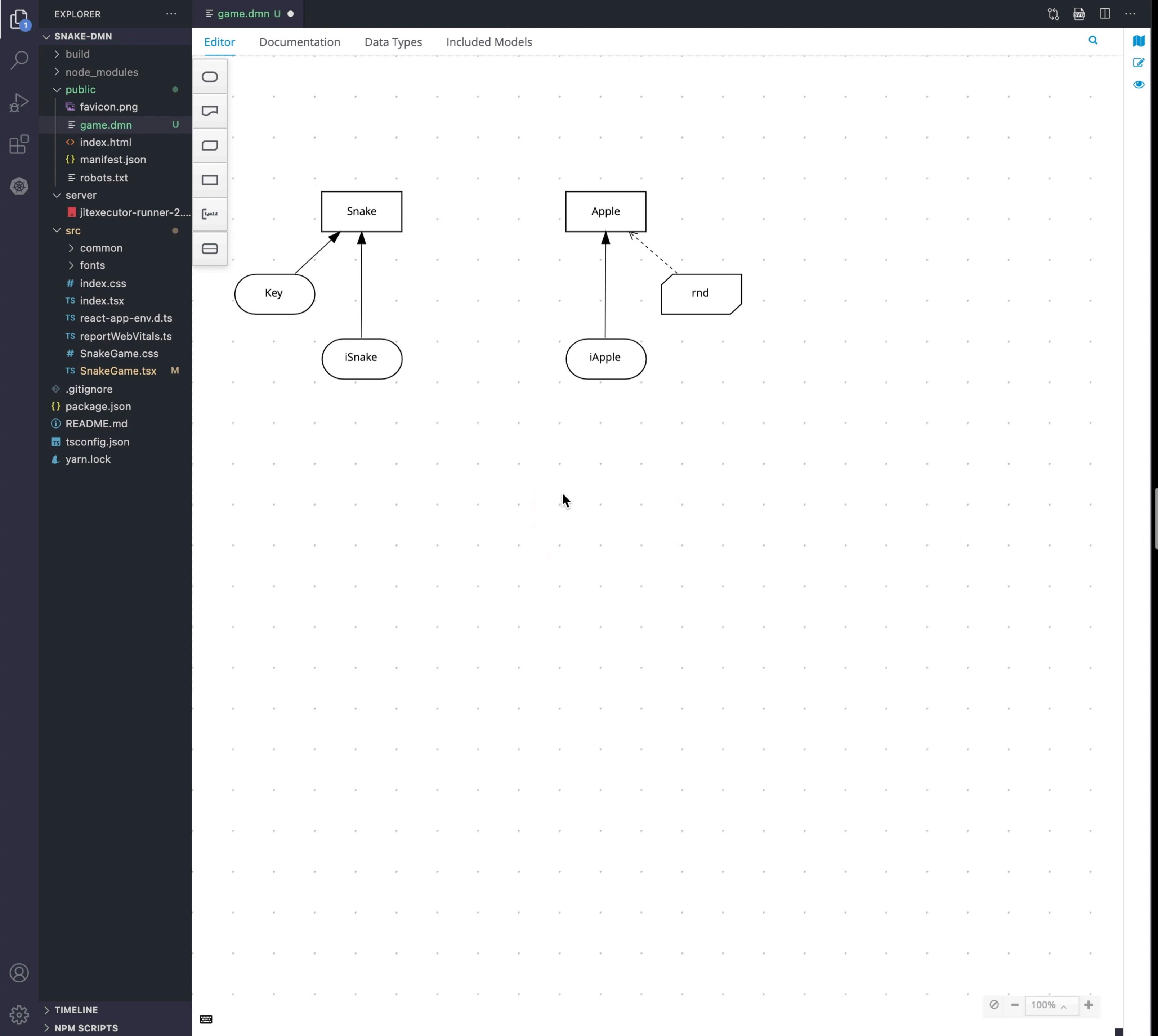
Timeline NPM Scripts

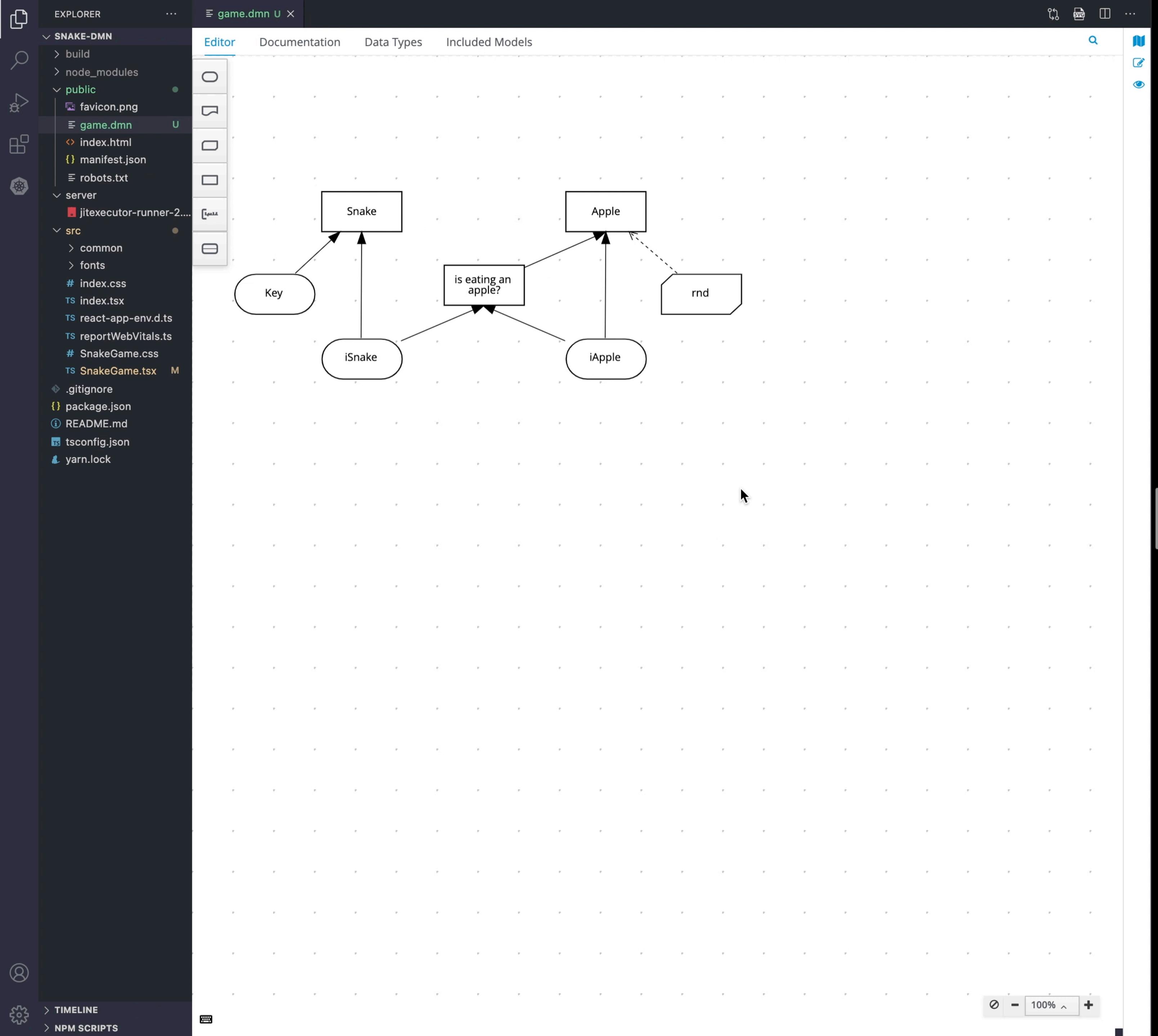
DMN Snake game

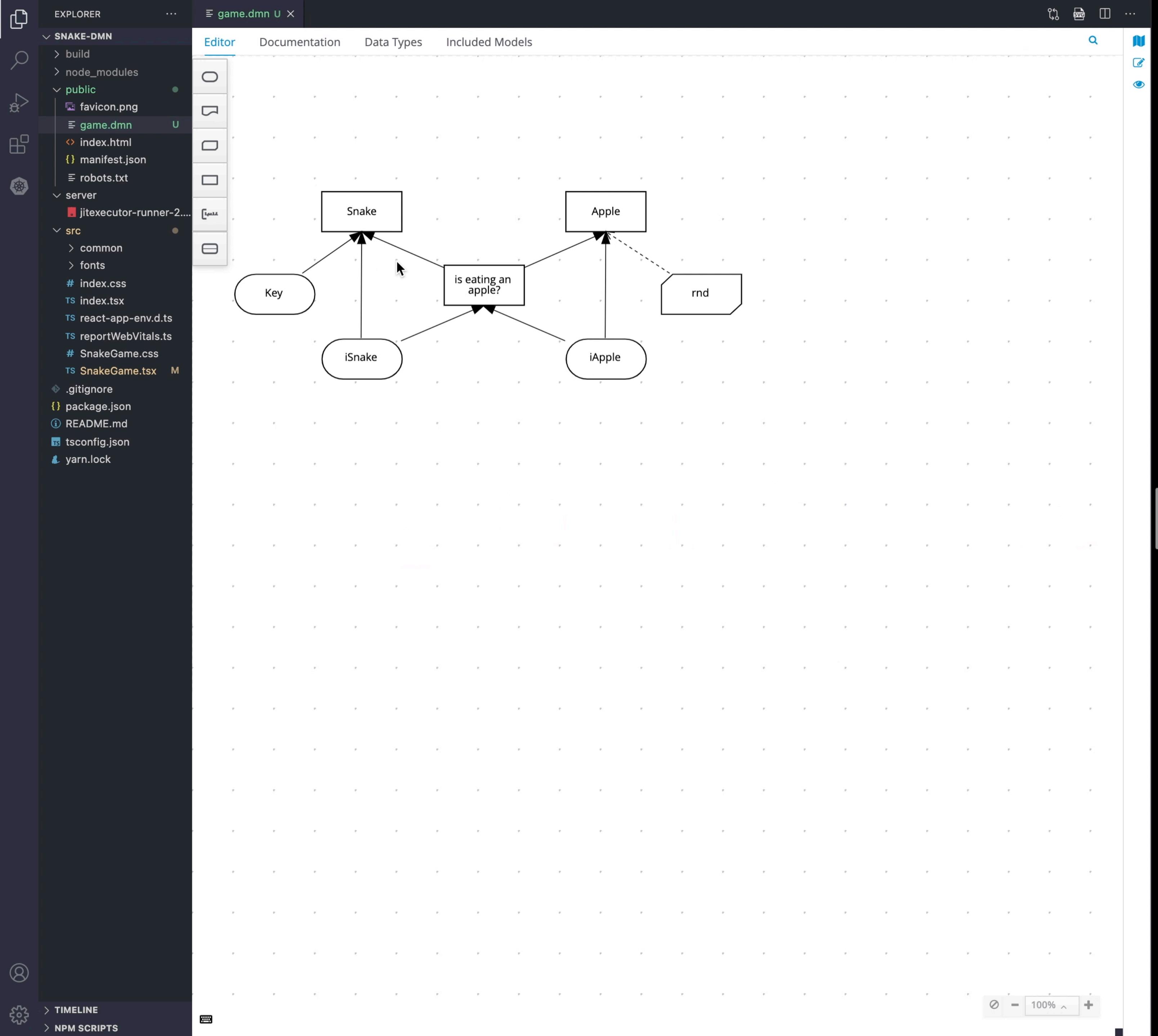
localhost:3000

# - Snake DMN -

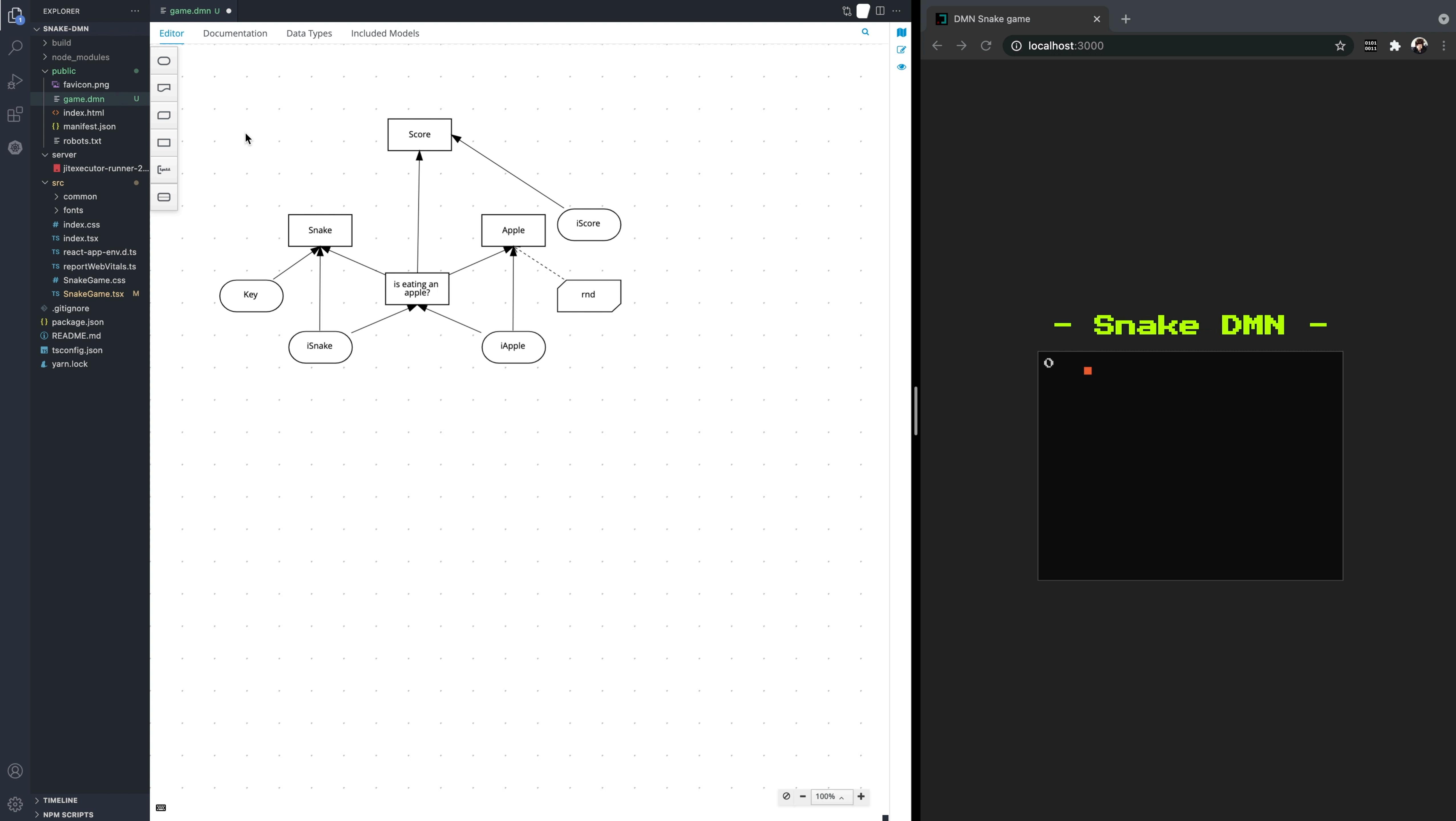
# IV. The **score** state







# V. The game over state



EXPLORER

1 SNAKE-DMN

> build  
> node\_modules  
public  
  favicon.png  
  game.dmn U  
  index.html  
  {} manifest.json  
  robots.txt  
server  
  jitexecutor-runner-2...  
src  
  > common  
  > fonts  
  # index.css  
  TS index.tsx  
  TS react-app-env.d.ts  
  TS reportWebVitals.ts  
  # SnakeGame.css  
  TS SnakeGame.tsx M  
.gitignore  
{} package.json  
 README.md  
 tsconfig.json  
 yarn.lock

Editor Documentation Data Types Included Models

game.dmn U

Back to snake-game

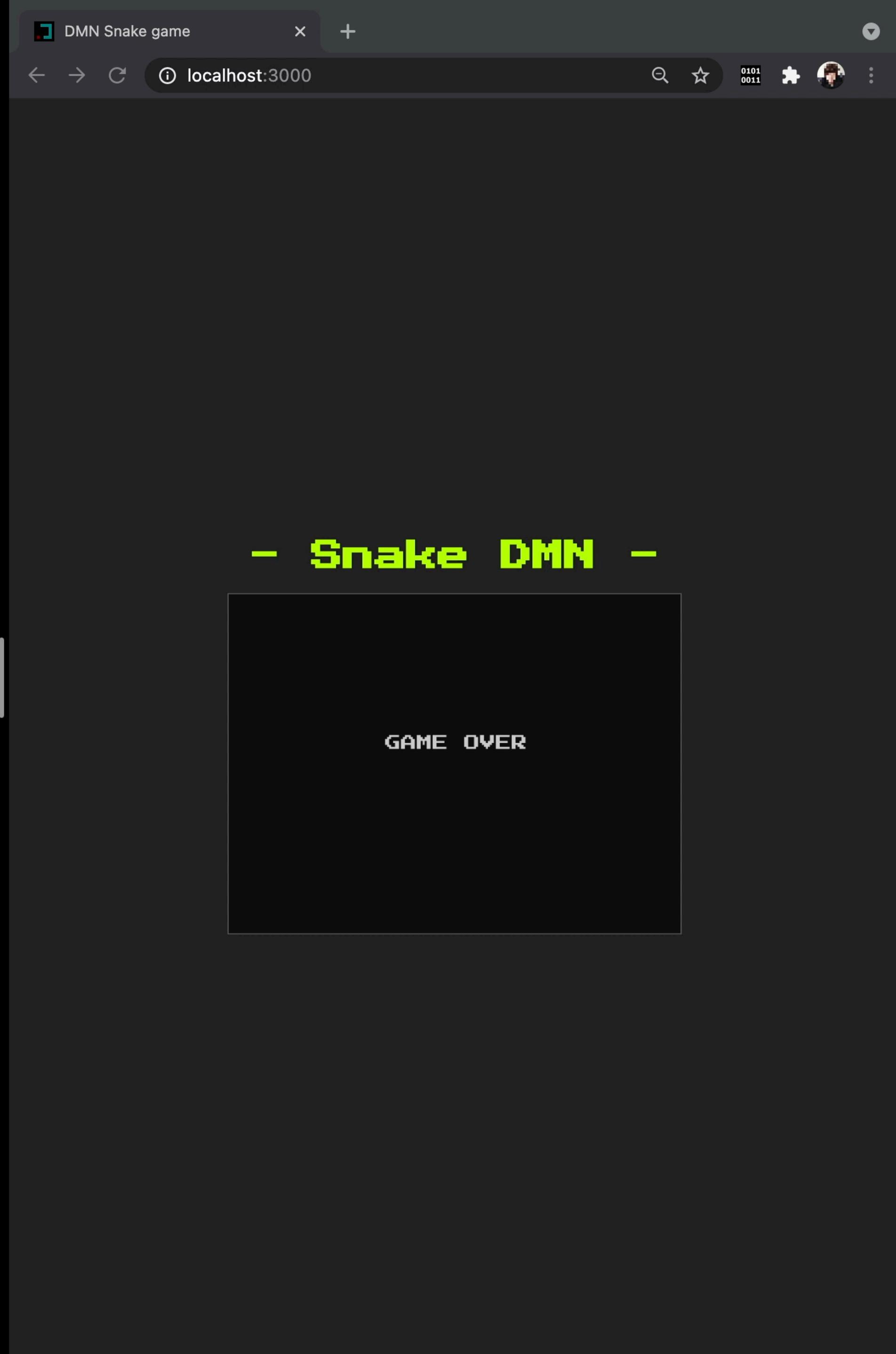
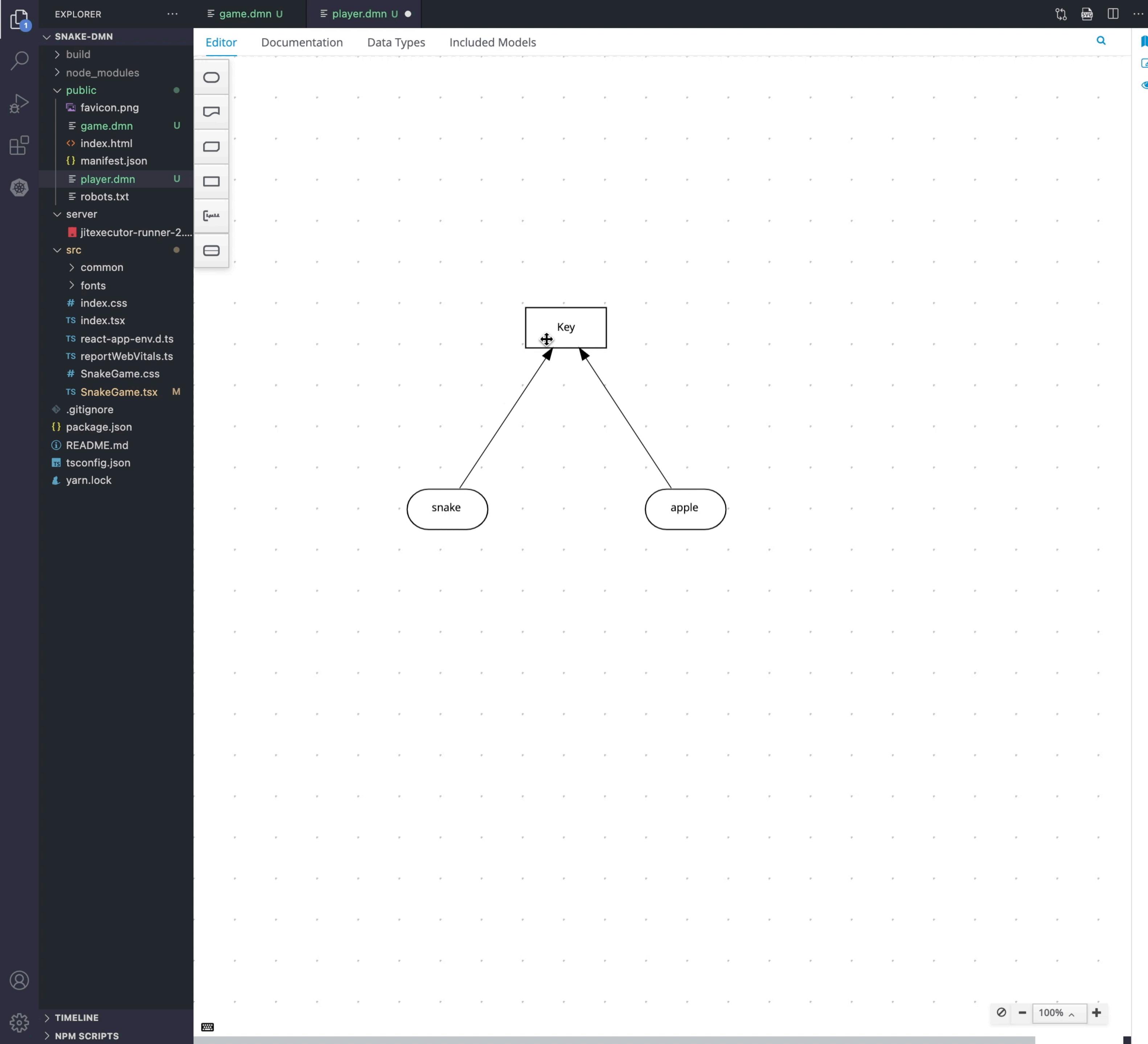
GameOver (Context)

#	GameOver (boolean)	
1	is eating itself? (boolean)	some i in Snake.body satisfies i.x = Snake.head.x and i.y = Snake.head.y
	<result>	iGameOver or is eating itself?





# The player state



EXPLORER    ...    game.dmn U    player.dmn U

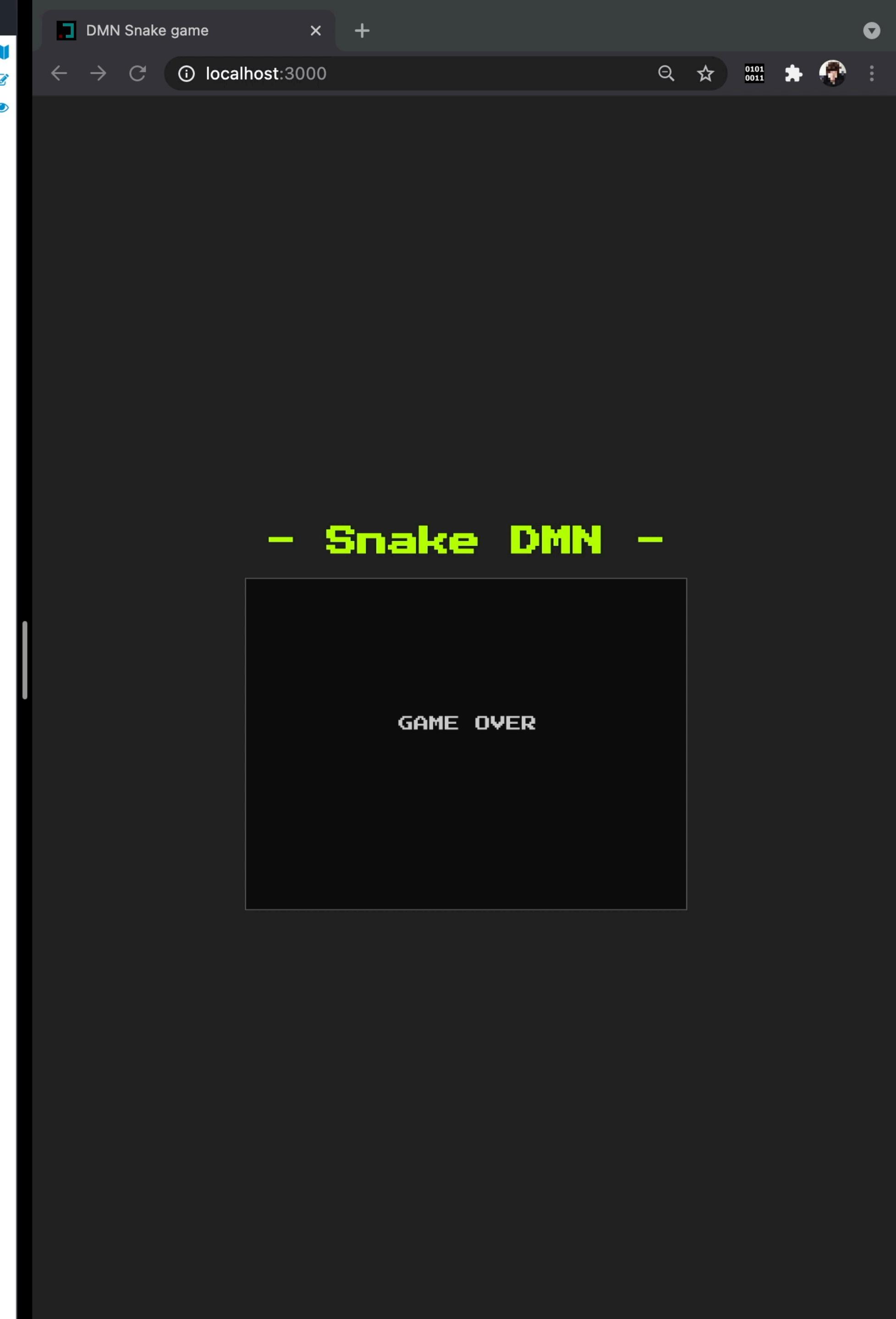
1 SNAKE-DMN

- > build
- > node\_modules
- < Back to snake-player
- < Back to snake-player
- Key (Context)

#		Key (string)
1	keys by distance (Any)	#
1	"Up"	{ x: snake.head.x, y: snake.head.y - 1 }
2	"Down"	{ x: snake.head.x, y: snake.head.y + 1 }
3	"Right"	{ x: snake.head.x + 1, y: snake.head.y }
4	"Left"	{ x: snake.head.x - 1, y: snake.head.y }
2	sorted (Any)	reverse( sort( keys by distance, function(a, b) a.distance > b.distance ) )
3	valid sorted (Any)	sorted[ not(some i in snake.body satisfies i.x = position.x and i.y = position.y) ]
<result>		if count(valid sorted) > 0 then valid sorted[1].k else "Right"

Editor Documentation Data Types Included Models

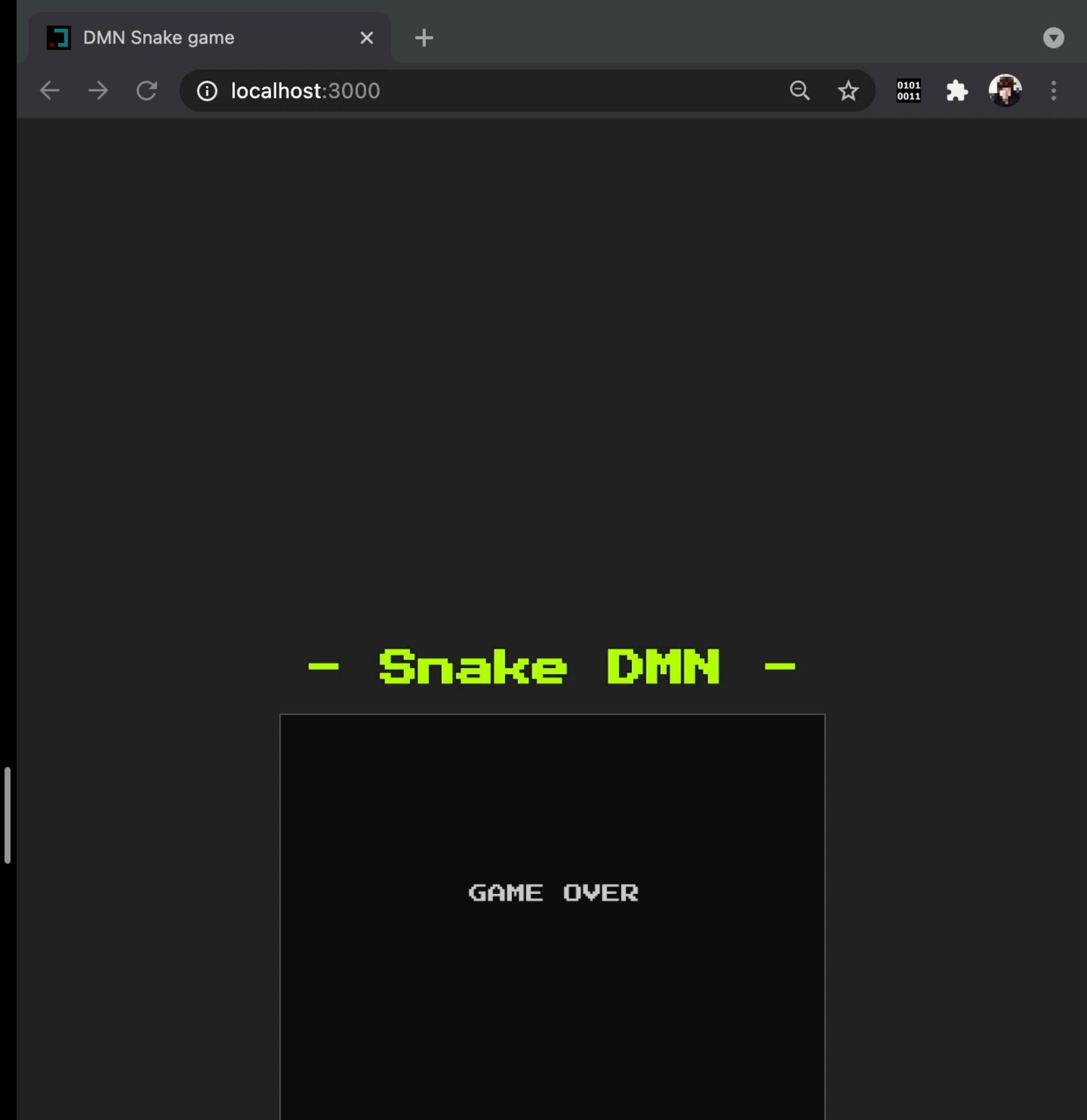
TIMELINE    NPM SCRIPTS



The screenshot shows a DMN (Decision Model and Notation) editor interface. The left sidebar displays a file tree for a project named "SNAKE-DMN". The "player.dmn" file is currently selected. The main area is titled "Key (Context)" and contains a table defining a context key for a snake game.

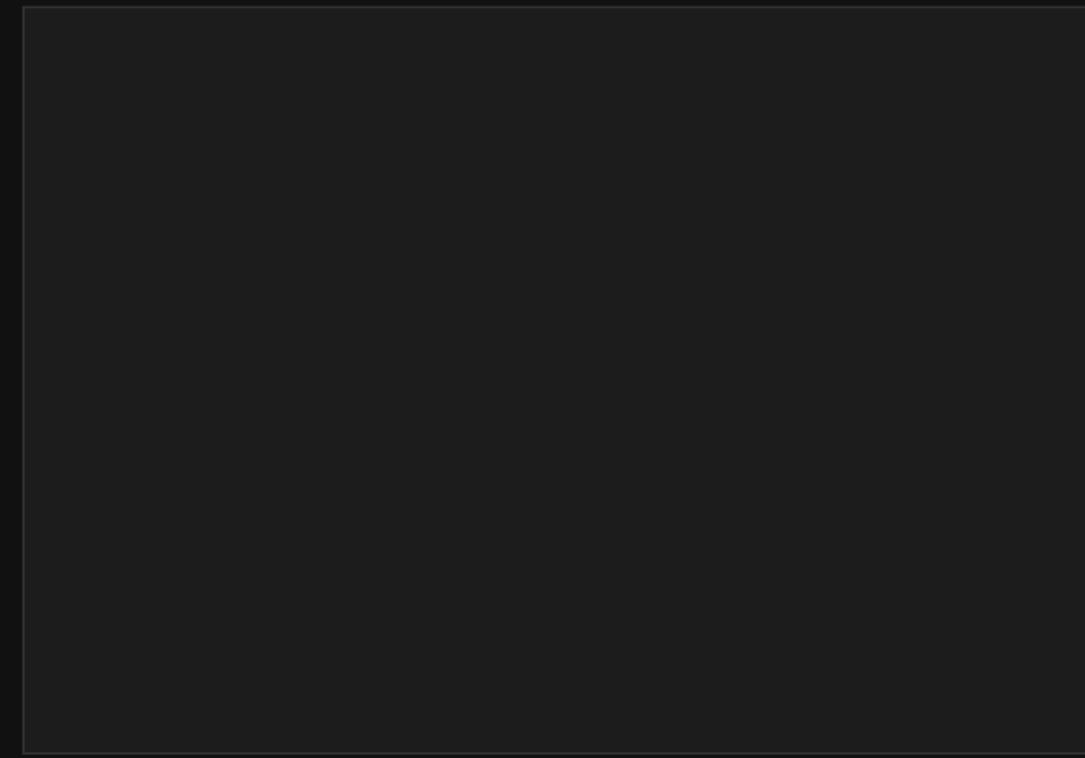
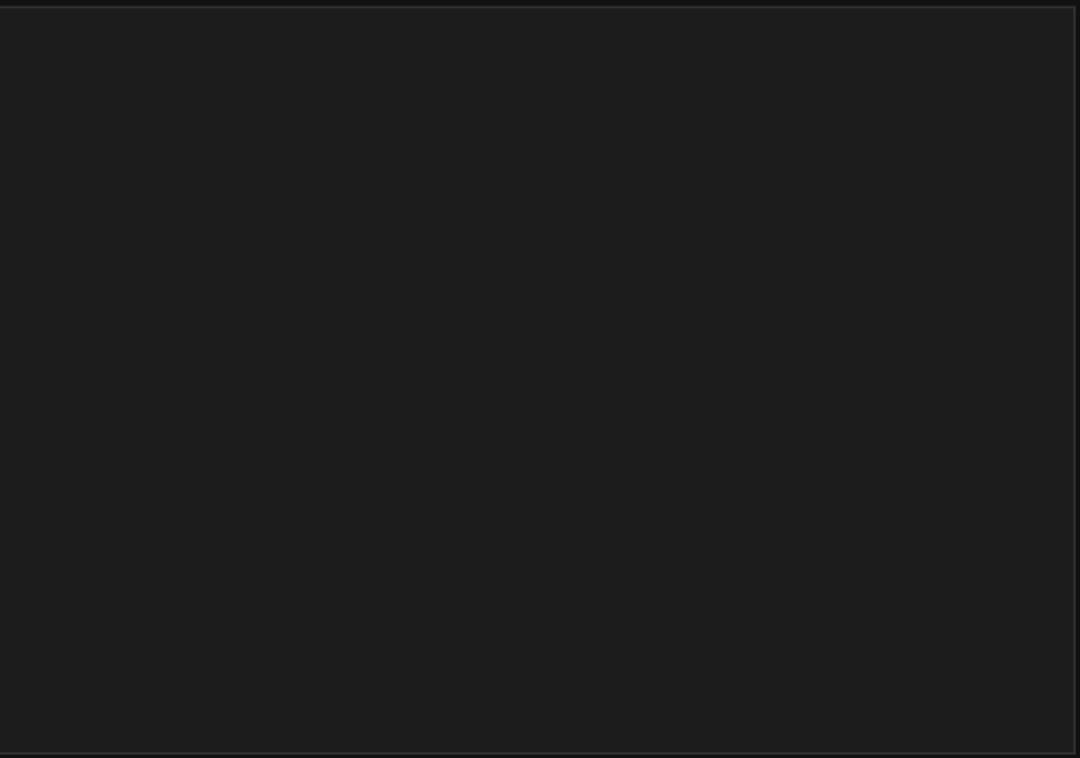
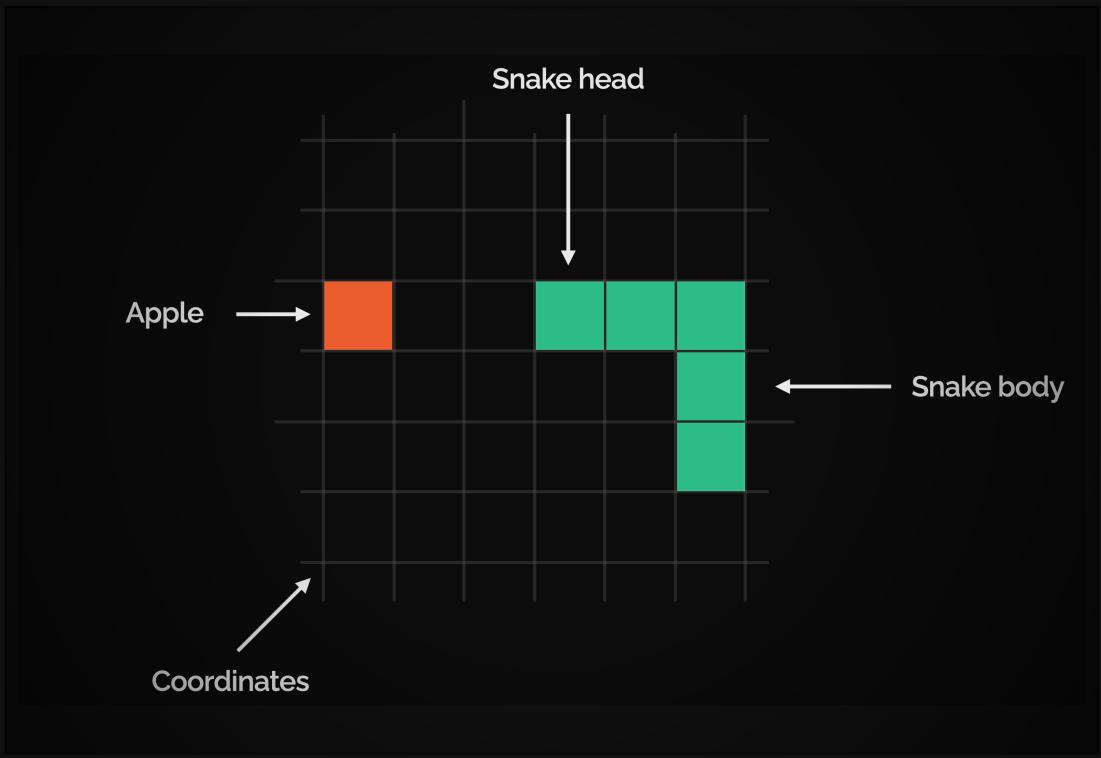
**Key (Context)**

#	Key (string)				
1	keys by distance (Any)	#	k (string)	position (Coordinate)	
		1	"Up"	{ x: snake.head.x, y: snake.head.y - 1 }	sqrt( ((apple.x - snake.head.x) * (apple.x - snake.h:
		2	"Down"	{ x: snake.head.x, y: snake.head.y + 1 }	sqrt( ((apple.x - snake.head.x) * (apple.x - snake.h:
		3	"Right"	{ x: snake.head.x + 1, y: snake.head.y }	sqrt( ((apple.x - (snake.head.x + 1)) * (apple.x - (:
	4	"Left"	{ x: snake.head.x - 1, y: snake.head.y }	sqrt( ((apple.x - (snake.head.x - 1)) * (apple.x - (:	
2	sorted (Any)	reverse( sort( keys by distance, function(a, b) a.distance > b.distance ) )			
3	valid sorted (Any)	sorted[ not(some i in snake.body satisfies i.x = position.x and i.y = position.y) ]			
	<result>	if count(valid sorted) > 0 then valid sorted[1].k else "Right"			

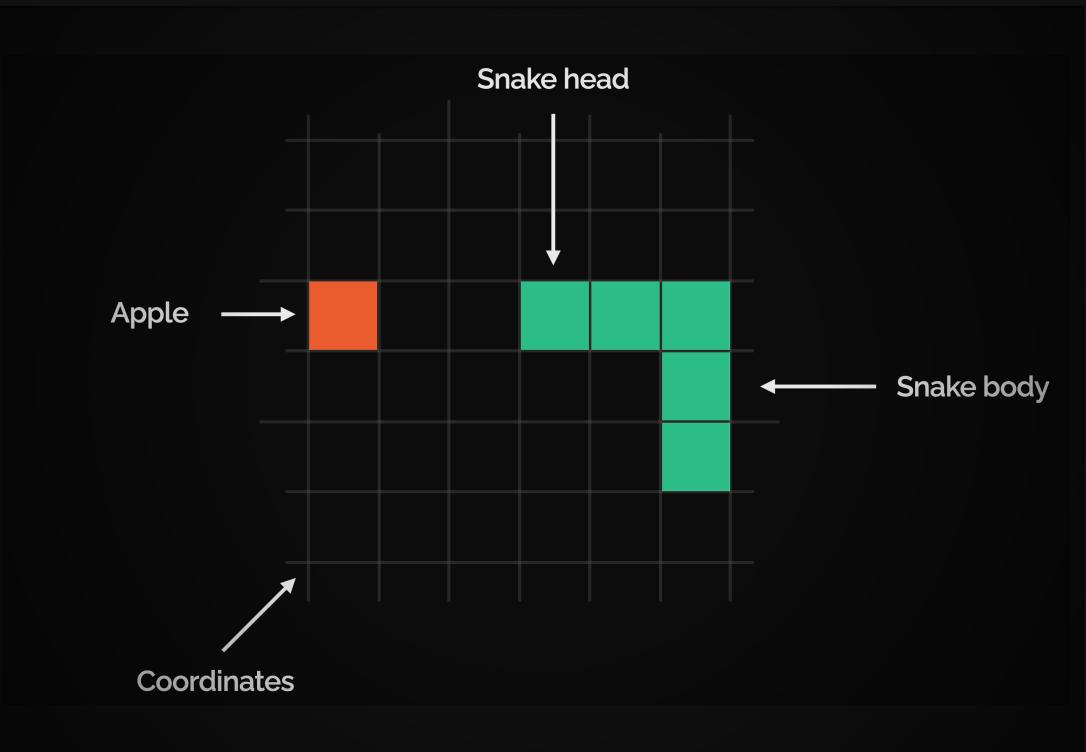








## GUI challenge



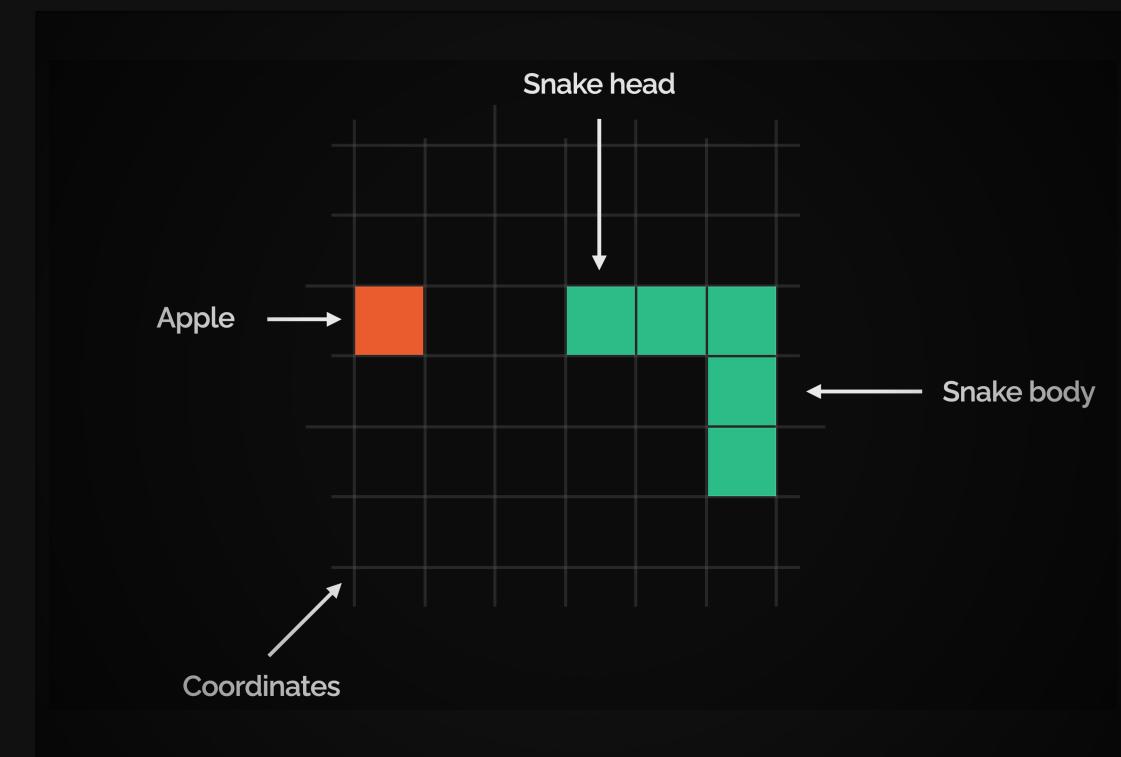
## GUI challenge

### The infrastructure side

```
function setKey(event: KeyboardEvent) {
  switch (event.key.toLowerCase()) {
    case "arrowup":
    case "w":
      key = "Up";
    case "arrowdown":
    case "s":
      key = "Down";
    case "arrowleft":
    case "a":
      key = "Left";
    case "arrowright":
    case "d":
      key = "Right";
  }
}

window.addEventListener("keydown", (event) => setKey(event), false);
```

## TypeScript



## GUI challenge

### The infrastructure side

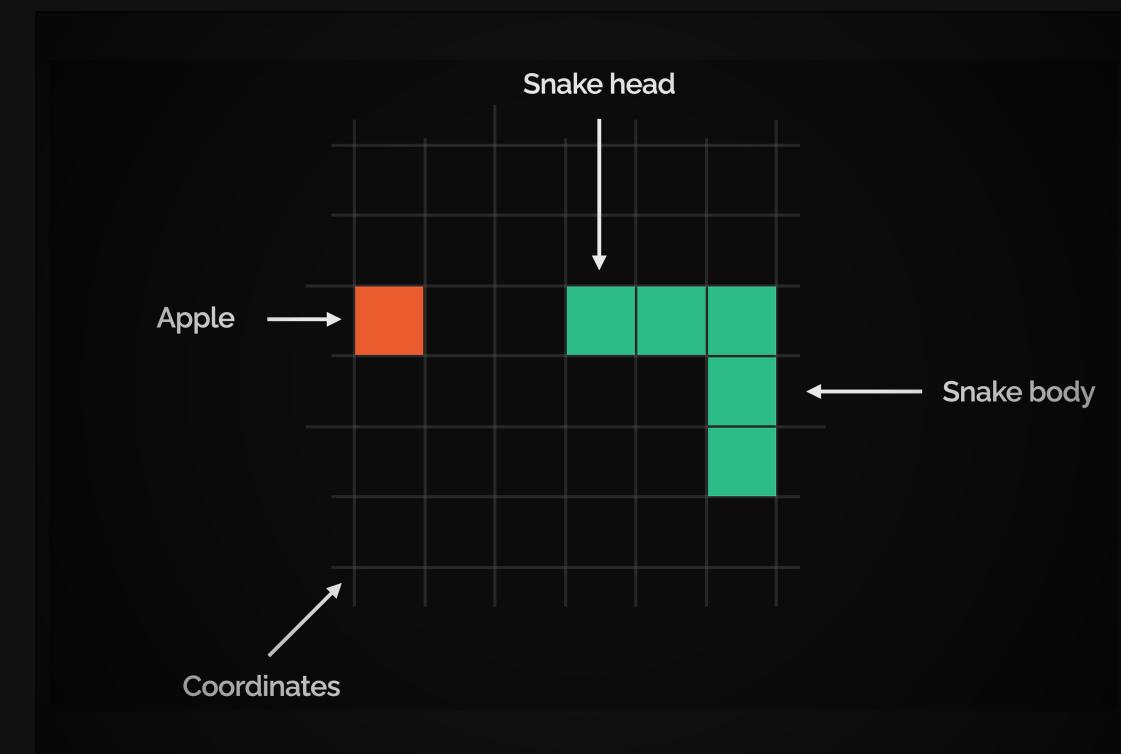
```
function setKey(event: KeyboardEvent) {
  switch (event.key.toLowerCase()) {
    case "arrowup":
    case "w":
      key = "Up";
    case "arrowdown":
    case "s":
      key = "Down";
    case "arrowleft":
    case "a":
      key = "Left";
    case "arrowright":
    case "d":
      key = "Right";
  }
}

window.addEventListener("keydown", (event) => setKey(event), false);
```

## TypeScript

```
package.json
{
  "name": "snake-dm",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "start-server": "java -jar server/jitexecutor-runner.jar",
    "start-client": "react-scripts start",
    "start": "concurrently \"yarn start-server\" \"yarn start-client\"",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ],
    "browserslist": {
      "production": [
        ">=0.2%",
        "not dead",
        "not op_mini all"
      ],
      "development": [
        "last-1 chrome version"
      ]
    }
  },
  "dependencies": {
    "@types/react": "16.0.0",
    "react-dom": "16.0.2",
    "react-scripts": "2.0.2",
    "web-vitals": "0.1.0"
  },
  "scripts": {
    "start-server": "java -jar server/jitexecutor-runner.jar",
    "start-client": "react-scripts start",
    "start": "concurrently \"yarn start-server\" \"yarn start-client\"",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ],
    "browserslist": {
      "production": [
        ">=0.2%",
        "not dead",
        "not op_mini all"
      ],
      "development": [
        "last-1 chrome version"
      ]
    }
  },
  "dependencies": {
    "@types/react": "16.0.0",
    "react-dom": "16.0.2",
    "react-scripts": "2.0.2",
    "web-vitals": "0.1.0"
  }
}
```

## Unusual integrations



## GUI challenge

The infrastructure side

```

function setKey(event: KeyboardEvent) {
  switch (event.key.toLowerCase()) {
    case "arrowup":
    case "w":
      key = "Up";
    case "arrowdown":
    case "s":
      key = "Down";
    case "arrowleft":
    case "a":
      key = "Left";
    case "arrowright":
    case "d":
      key = "Right";
  }
}

window.addEventListener("keydown", (event) => setKey(event), false);

```

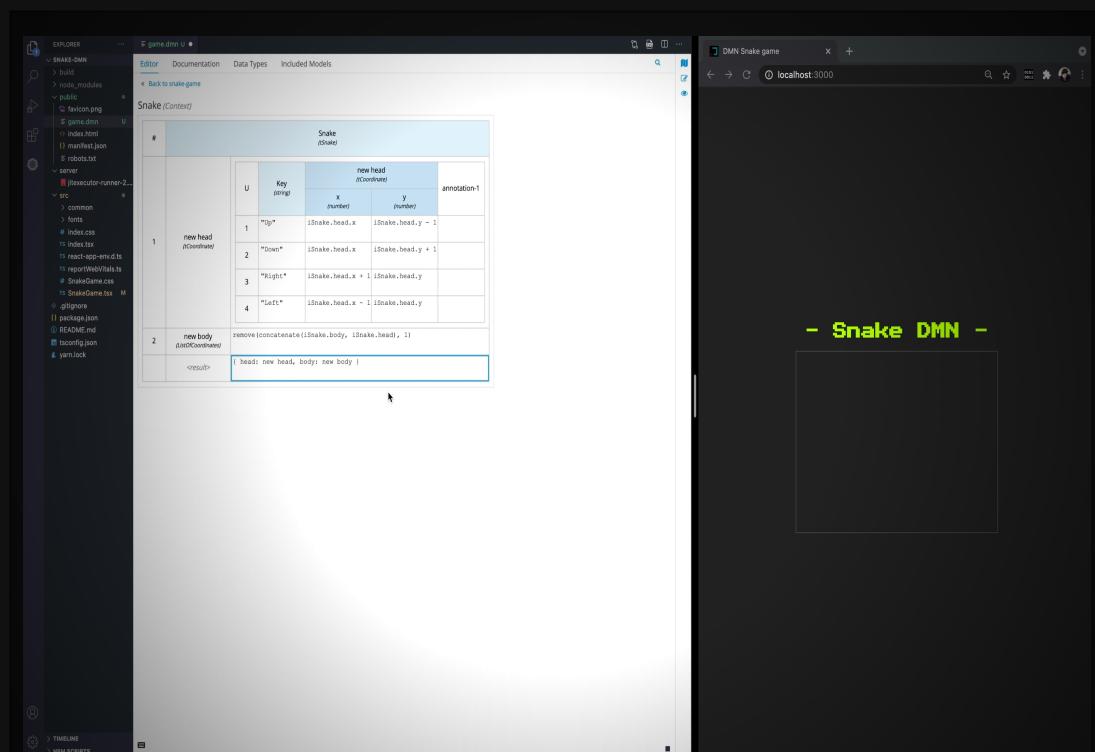
## TypeScript

```

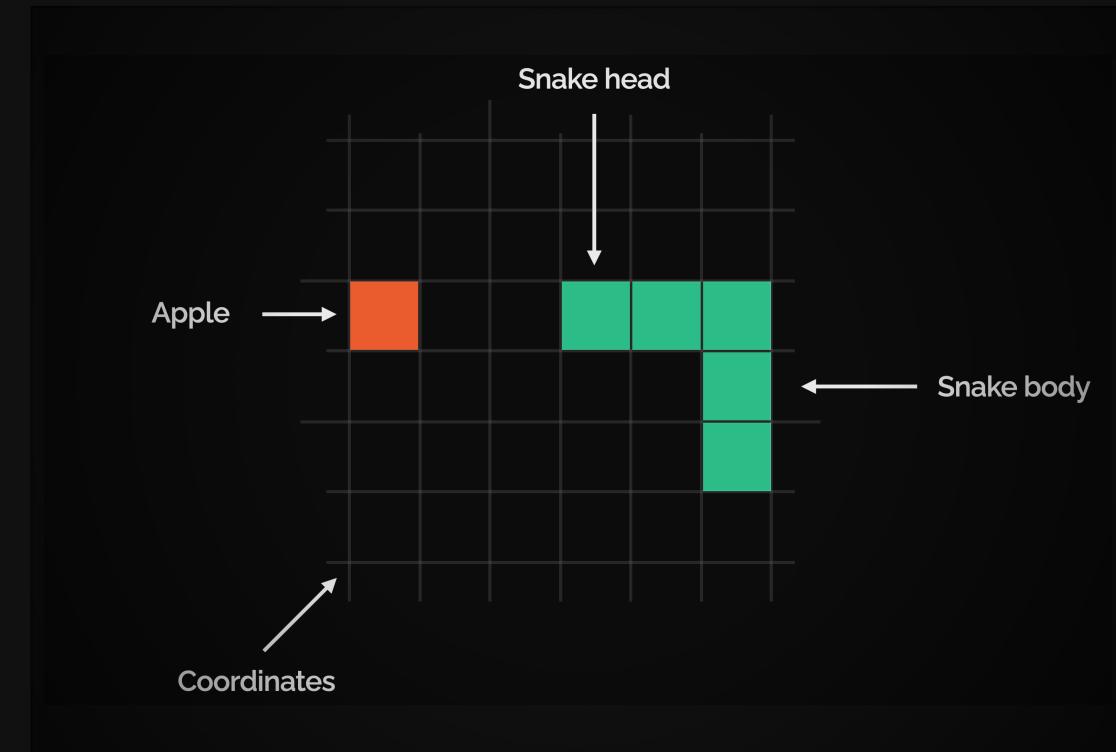
{
  "name": "SNAKE-DMN",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "start-server": "java -jar server/jitexecutor-runner.jar",
    "start-client": "react-scripts start",
    "start": "concurrently \"yarn start-server\" \"yarn start-client\"",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">=0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last-1 chrome version"
    ]
  }
}

```

## Unusual integrations



## DMN



## GUI challenge

The infrastructure side

```

function setKey(event: KeyboardEvent) {
  switch (event.key.toLowerCase()) {
    case "arrowup":
    case "w":
      key = "Up";
    case "arrowdown":
    case "s":
      key = "Down";
    case "arrowleft":
    case "a":
      key = "Left";
    case "arrowright":
    case "d":
      key = "Right";
  }
}

window.addEventListener("keydown", (event) => setKey(event), false);

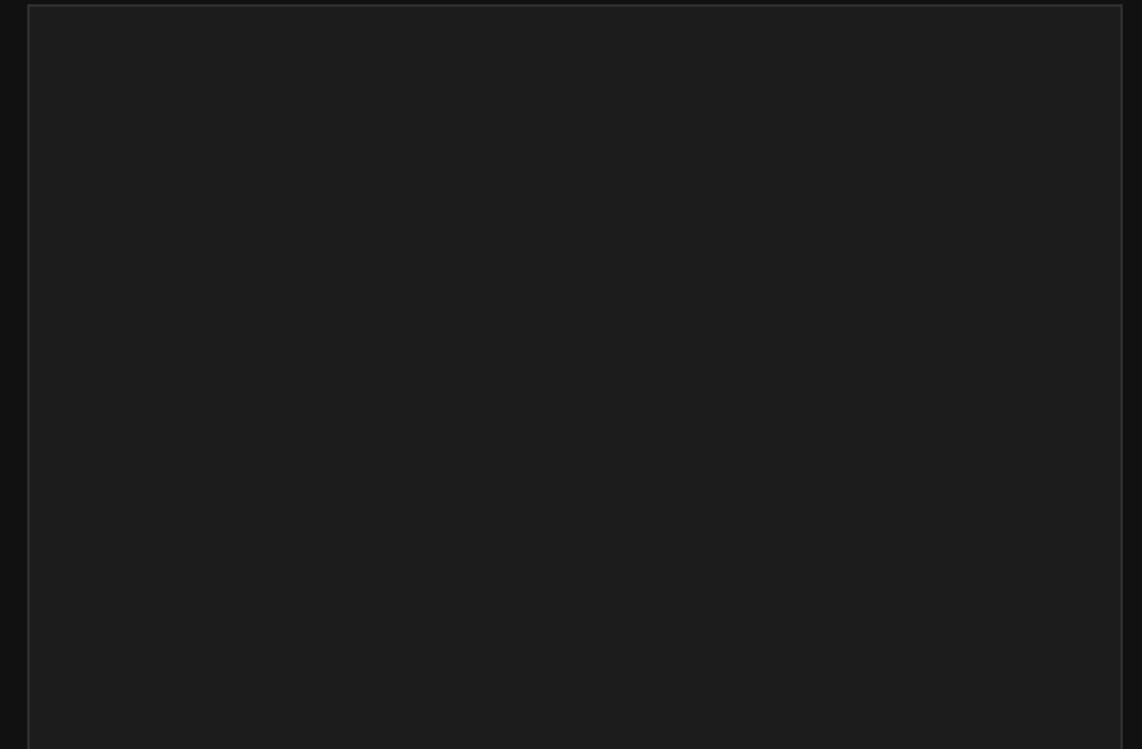
```

```

{
  "name": "snake-dmn",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "start-server": "java -jar server/jitexecutor-runner.jar",
    "start-client": "react-scripts start",
    "start": "concurrently \"yarn start-server\" \"yarn start-client\"",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ]
  },
  "browserslist": {
    "production": [
      ">=0.2%",
      "not dead",
      "not op_mini all"
    ],
    "development": [
      "last 1 chrome version"
    ]
  }
}

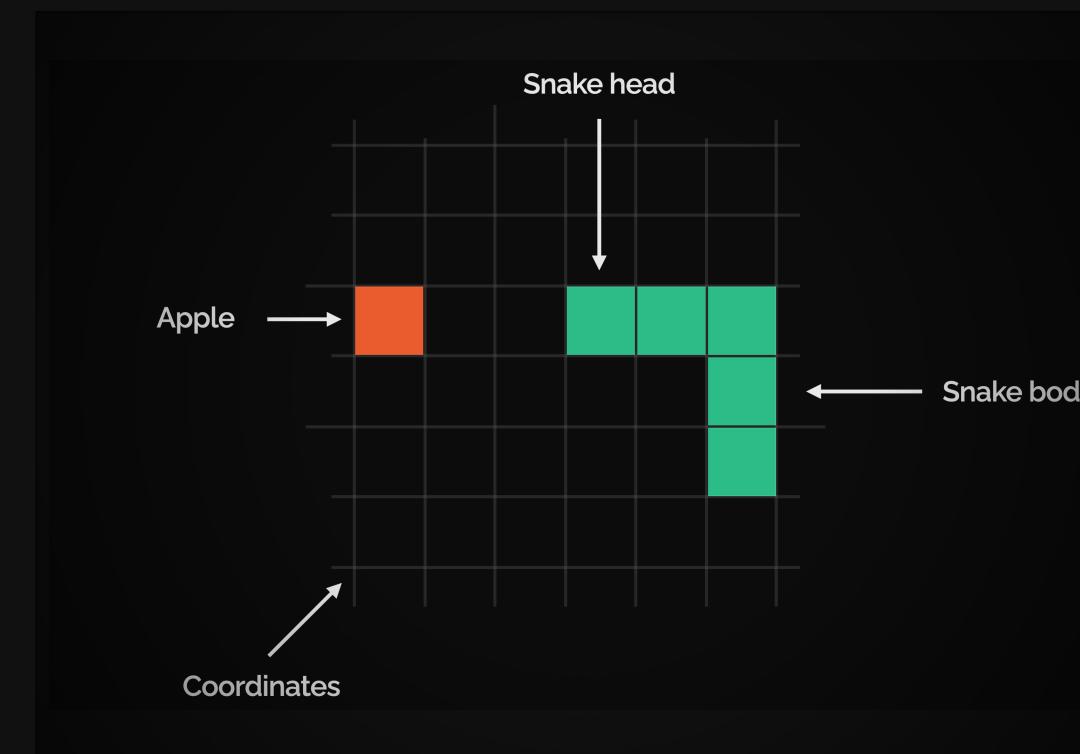
```

## Unusual integrations



## DMN

## Interesting algorithm



## GUI challenge

The infrastructure side

```

function setKey(event: KeyboardEvent) {
  switch (event.key.toLowerCase()) {
    case "arrowup":
    case "w":
      key = "Up";
    case "arrowdown":
    case "s":
      key = "Down";
    case "arrowleft":
    case "a":
      key = "Left";
    case "arrowright":
    case "d":
      key = "Right";
  }
}

window.addEventListener("keydown", (event) => setKey(event), false);

```

## TypeScript

package.json

```

{
  "name": "SNAKE-DMN",
  "version": "0.0.1",
  "private": true,
  "scripts": {
    "start-server": "java -jar server/jitexecutor-runner.jar",
    "start-client": "react-scripts start",
    "start": "concurrently \"yarn start-server\" \"yarn start-client\"",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": [
      "react-app",
      "react-app/jest"
    ],
    "browserslist": {
      "production": [
        ">=0.2%",
        "not dead",
        "not op_mini all"
      ],
      "development": [
        "last-1 chrome version"
      ]
    }
  }
}

```

## Unusual integrations

- Snake DMN -

- Snake DMN -

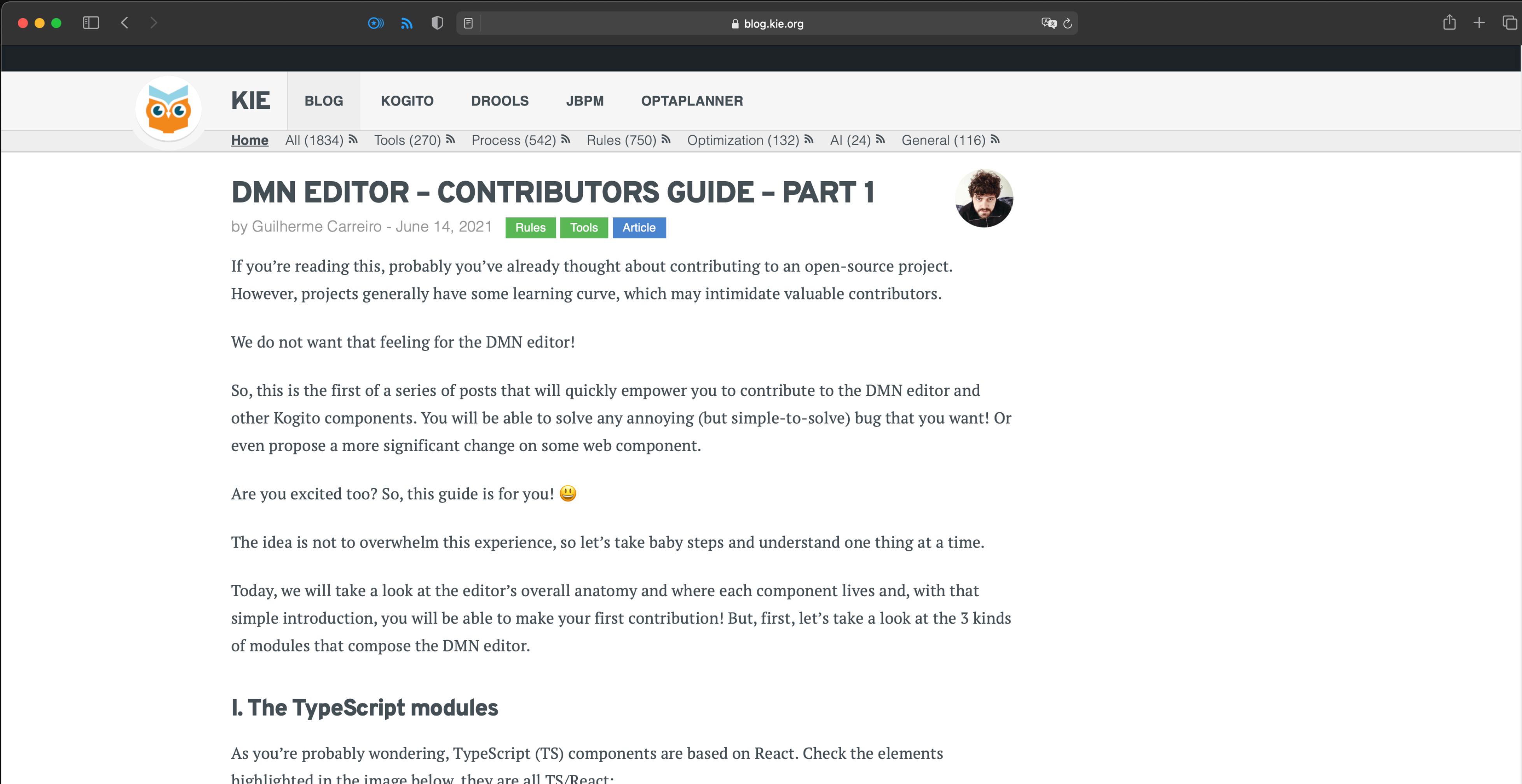


## DMN

## Interesting algorithm

## Ready!

# DMN contributors guide



The screenshot shows a web browser window with the URL [blog.kie.org](https://blog.kie.org) in the address bar. The page title is "DMN EDITOR – CONTRIBUTORS GUIDE – PART 1". The header includes the KIE logo, navigation links for BLOG, KOGITO, DROOLS, JBPM, and OPTAPLANNER, and a search bar. Below the header, there's a navigation menu with links to Home, All (1834), Tools (270), Process (542), Rules (750), Optimization (132), AI (24), and General (116). The main content area features a portrait of the author, Guilherme Carreiro, and the text: "If you're reading this, probably you've already thought about contributing to an open-source project. However, projects generally have some learning curve, which may intimidate valuable contributors. We do not want that feeling for the DMN editor! So, this is the first of a series of posts that will quickly empower you to contribute to the DMN editor and other Kogito components. You will be able to solve any annoying (but simple-to-solve) bug that you want! Or even propose a more significant change on some web component. Are you excited too? So, this guide is for you! 😊 The idea is not to overwhelm this experience, so let's take baby steps and understand one thing at a time. Today, we will take a look at the editor's overall anatomy and where each component lives and, with that simple introduction, you will be able to make your first contribution! But, first, let's take a look at the 3 kinds of modules that compose the DMN editor.

## I. The TypeScript modules

As you're probably wondering, TypeScript (TS) components are based on React. Check the elements highlighted in the image below, they are all TS/React:

learn-dmn-in-15-minutes.com

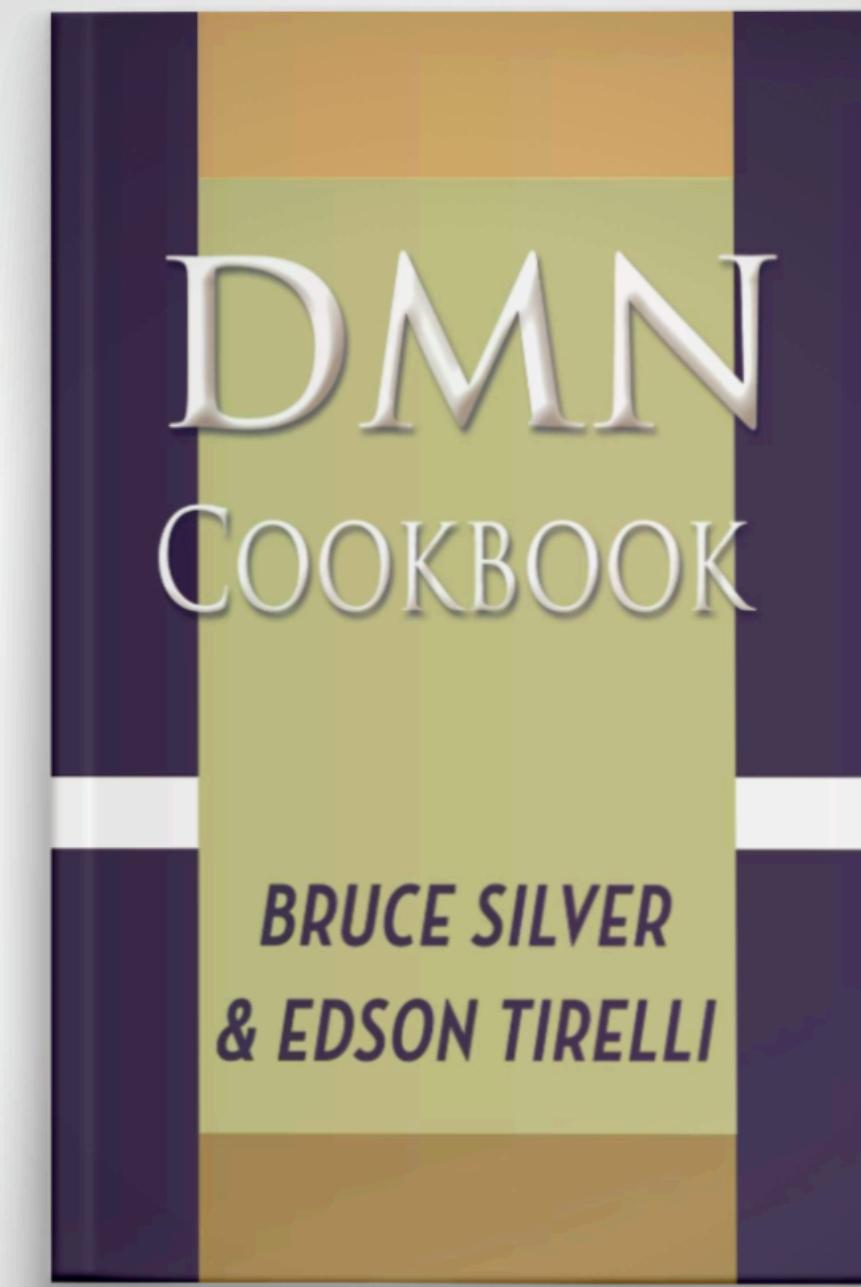
# Learn DMN in 15 minutes v2

START LEARNING NOW!

— or —

[Try the DMN online editor](#)

The Decision Model and Notation (DMN™) is a Standard by [OMG®](#), providing a conventional and visual notation readily understandable by all users and personas.



<https://bit.ly/dmn-cookbook>

# Thank you

Slides: [karreiro.com/talks](http://karreiro.com/talks)