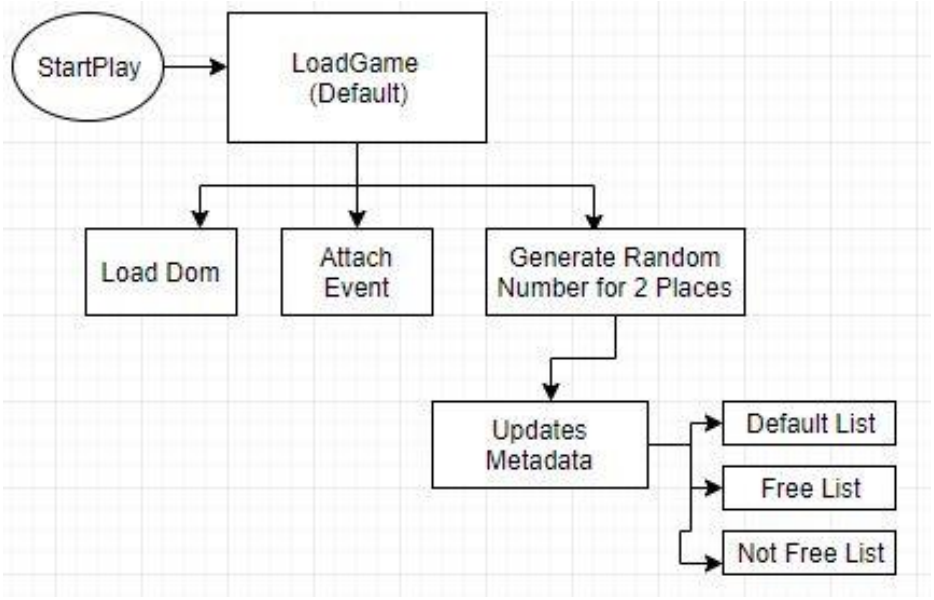
Hand Book – 2048 Game Dev

- **Requirements Analysis:**
  - Build a 2048 game with 4*4 using JS.
  - Operate from console using 1,2,3,4 for left, right, up and down.
  - Move titles accordingly and merge adjacent title if values are same.
  - Player should be able to reach 2048 using controls.

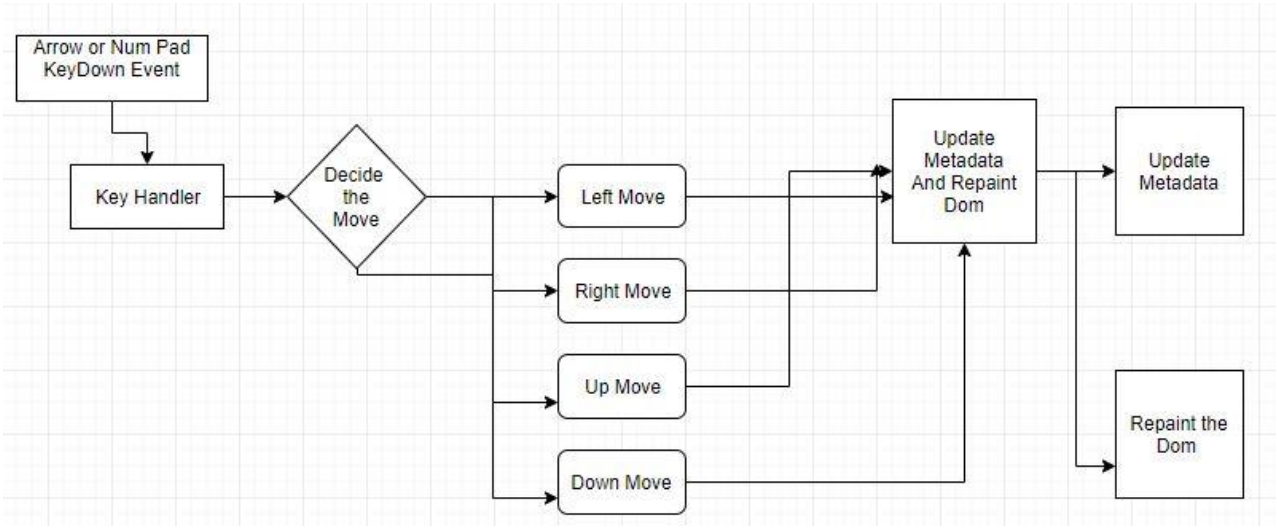
- **Motto:**
  - Build a solution for the requirement which is stable, scalable and with minimalistic UI.

- **Project Structure:**
  - 2048-GAME-OOJS-DEV (Parent Directory)
    - StartPlay.html
    - Resources directory.
      - CSS - Contains the styles for the game.
      - GameConfig - Contains the game configuration.
      - GameEngine - Contains modules for Matrix operations and Move Handling.
      - GameLib - Contains the module for random number generation.
      - GameMetaData - Contains the Data Accessor and Game metadata.
      - OtherLib - Contains other Libaries (Jquery and HandleBars for templating).

- **Flow Diagram**
  - **Onload Phase**



- **OnKey Down Phase**



- **Metadata Object Structure:**
  - DefaultList : List of all Position , during initial load.
  - Free : Holds list of positions free after every movement.
  - Not Free : Holds list of positions not free positions.
  - Master Data : Holds the latest data/values of matrix after every moment.

- **Free and DefaultList** : List holds the position which are free and available for random position generation. Random Number generation logic uses this list to introduce new number, with this approach the iteration are reduced.

Reference:

```
▼{defaultList: Array(14), free: Array(0), notFree: Array(2), masterData: {…}, weights: {…}}
  ▶defaultList: (14) ["R0C0", "R0C1", "R0C2", "R0C3", "R1C0", "R1C1", "R1C2", "R1C3", "R2C0", "R2C1", "R2C2", "R3C0", "R3C1", "R3C2"]
  ▶free: []
  ▼masterData:
    ▼R0: Array(4)
      ▶0: cell {value: 0}
      ▶1: cell {value: 0}
      ▶2: cell {value: 0}
      ▶3: cell {value: 0}
       length: 4
      ▶__proto__: Array(0)
    ▶R1: (4) [cell, cell, cell, cell]
    ▶R2: (4) [cell, cell, cell, cell]
    ▶R3: (4) [cell, cell, cell, cell]
    ▶__proto__: Object
  ▶notFree: (2) ["R2C3", "R3C3"]
  ▶weights: {0.55: "R0C1"}
```

Steps to Start game:

Note:

- By Default game is configured to 4*4, configuration can be changed in GAME_CONF.js
- Default controls are set to numpad, this configuration can also be changed in GAME_CONF.js (as shown below).

Game Size:

```
const GAME_CONF = (function(){
    let gameConf =
    {
        "gameSize": 4,//Default set to 4, can be extended to 8 or N
```

Controls Conf:

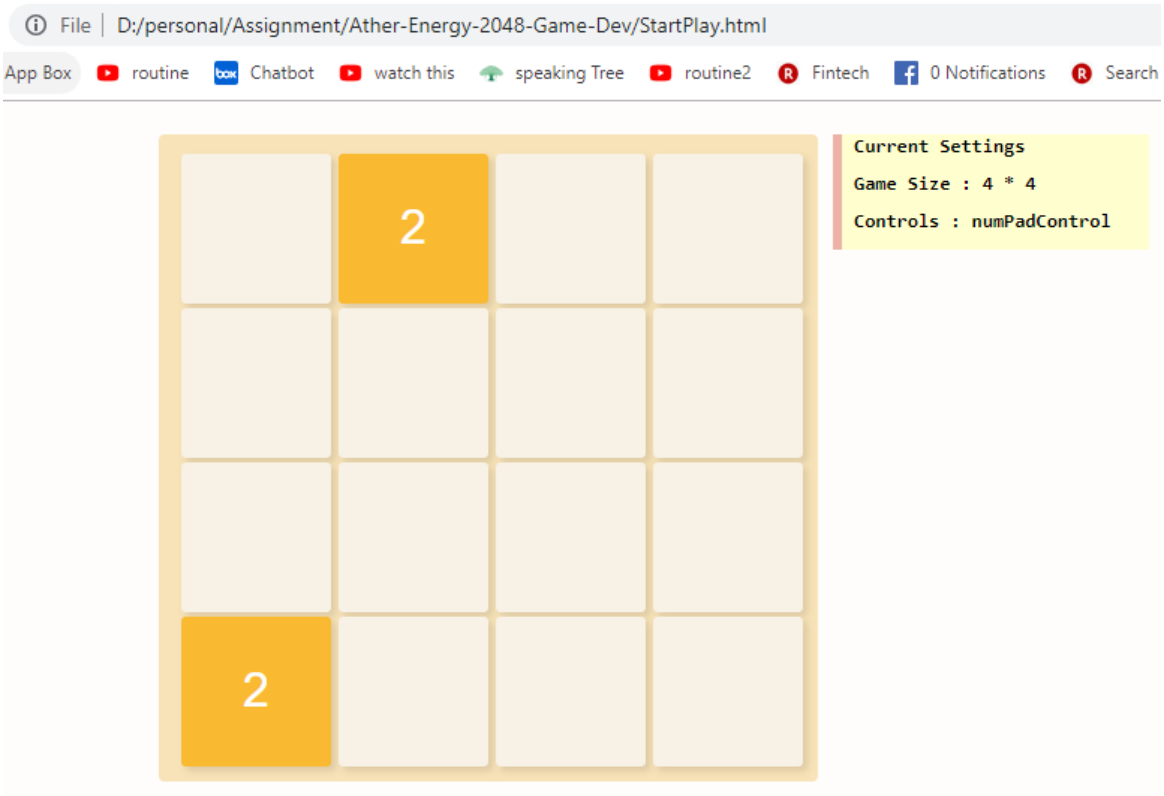
```
"movementControl": "numPadControl",
```

Available Settings:

i. “arrowsControl” for Arrow movement.
ii. “numPadControl” for number movement.

1. Launch “StartPlay.html” in browser (Best viewed in Chrome).
2. Enjoy the playing using the controls.

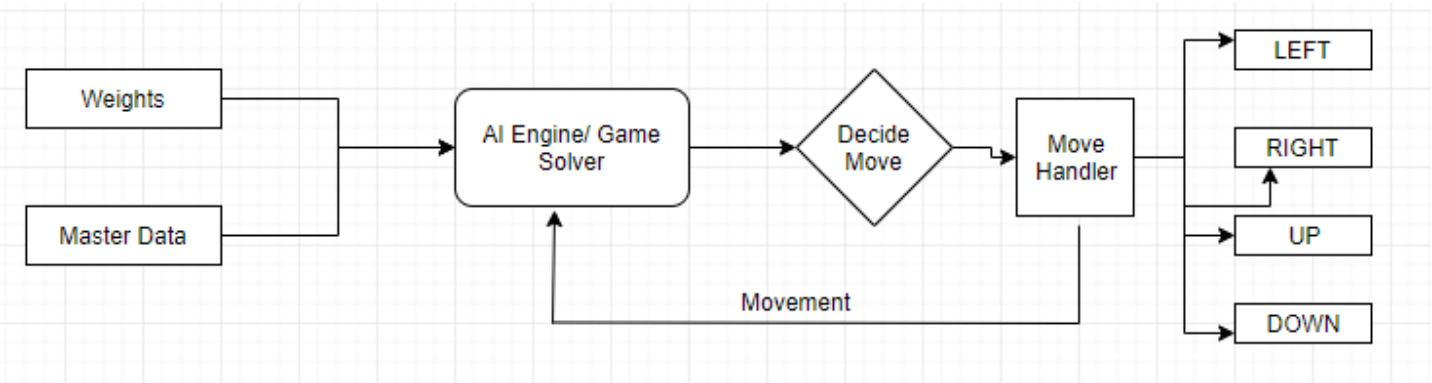
For better user experience , game interaction is from UI.



Future Enhancements:

"Weights" structure in Metadata , can be programmed to store the highest weight/value against the position. Along with the masterdata , weights can be sent to AI/Computer solver to make the corresponding move.

With the decide move , movement can be sent back to the AI Engine for further training.



Flow of Game using AI Engine