

# KingKong v0.0.1

## Class Structure

### Main

---

1. init **stage**
2. init Debug Utilities (**Stats, Logger etc**)
3. init **Preloader**
4. init **ServerInterface**
5. init **Game**

Dependency:

1. App dependencies loaded in index.html
2. Main entry point called in index.html

### Preloader

---

1. Load **Setup**
2. Load **Sounds**
3. Load **Spritesheets**
4. Notify **Game** with **LOAD\_COMPLETE** Event
5. Cache loaded Assets in **DJ / Artist / GameState**

Dependency:

1. Initialised by **Main**
2. **Game**
3. Should be aware of Device OS & Browser capability
4. **DJ**
5. **Gallery**
6. **GameState**

### DJ

---

1. Cache loaded **Sounds**
2. Respond to **SOUND\_EVENTS** to play Sounds
3. **Or** provide **callbacks** to play sounds

### Gallery

---

1. Cache loaded Art Assets
2. Provide getters to the rest of the game to access **Art**.

# KingKong v0.0.1

## Game

---

1. Store **GameState**
2. Init **GameComponents**
3. Run stage update **Ticker**
4. Init **UI Events**
5. Init **CommandQueue**

Dependency:

1. **GameState**
2. **GameComponents**
3. stage
4. **ServerInterface**
5. **CommandQueue**

## ServerInterface

---

1. Verify connection to Server
2. Handle connection Success
3. Handle connection Error
4. Handle timeouts
5. Define server requests
6. Handle server responses and Notify **Game**

Dependency:

1. Initialised by **Main**
2. **Game**
3. A Server

## GameComponents

---

1. Component Display
2. Wire UI Events to **ServerInterface**
3. Wire Commands from **ServerInterface**

Dependency:

1. User
2. **ServerInterface**
3. stage

## CommandQueue

---

1. Creates arrays of Commands which can be scheduled

## KingKong v0.0.1

2. Can execute Commands over time
3. Calls out to QueueFactory to generate specific preset queues

Dependency:

1. **QueueFactory**
2. **Command**

### Command

---

1. Is given data to initialise itself (eg. a WinLine Data)
2. Is given a GameComponent to call based on data (eg. a WinLine)
3. Has an execute function which runs necessary functions on GameComponent

Dependency:

1. **GameComponent**
2. **Model**

### QueueFactory

---

1. Creates Preset CommandQueues and returns them to CommandQueue

Dependency:

1. Setup
2. GameComponents
3. Command
4. CommandQueue

## GameComponents

### Reels

---

- Display **Reels** according to setup
- Insert New **Reels** according to **ServerInterface**
- Spin **Reels** with customisable Tweens
- Notify **Game** with **REEL\_SPIN\_COMPLETE** Event

### SymbolWins

---

- Should be able to create DisplayList and initialise **Sprites** upfront

# KingKong v0.0.1

- Run Winning **SymbolWin** animations
- Blink or Frame based Spritesheet Animations.
- Called by **SymbolWinCommand**

## WinLines

---

- Should be able to create DisplayList upfront
- Cache vector drawings
- Display and animate **WinLines** in correct positions
- Called by **WinLineCommand**

## BigWin

---

- Should be able to create DisplayList and initialise **Sprites** upfront.
- Display and animate **BigWin** animations
- Called by **BigWinCommand**

## ParticlesLayer

---

- Should be able to craete DisplayList upfront
- Run celebratory particle animation
- Called by **ParticlesCommand**

## GaffMenu

---

- Should create a **GaffMenu** with a set of buttons based on **Setup** Options
- Should be able to show/hide based on user keyboard/touch Events.
- Should be able to execute customisable actions in the game based on **Setup**.