**DICE GAME HELP**

Introducing a HTML5 Canvas game framework written in Object Oriented JavaScript. This includes intermediate level of a Dice Class, a Sounds Class, Coins Falling animation Class, a basic State Machine Class, and obviously the game logic.

The JavaScript 3d rotated dice class (Object Oriented Development), which allows you to customize its texture/sides and allows a number from you to stop at.

This is very easy to integrate with any casino engine as it takes a number to stop at. You keep the number generation logic at backend and assign the numbers generated by a casino engine to the dices. Later when all dices are stopped, you compare with the random numbers generated from the game engine to pay for any win to avoid game hacks.

This game is purely developed on HTML5 canvas and JavaScript, and no libraries are being used for any purpose.

HTML5 canvas does not allow movements, it only renders each frame. So, we must draw things on each frame by using requestAnimationFrame feature.

**Customizing the dice:**

You must redesign the dice images to give the dice a new look and all other things are being taken care. There are six images for the dice named 1 to 6 in PNG format. You can use any format by the way.

**Button Customization:**

In HTML5 canvas we cannot render a button with all events attached to it. To replicate a simple mouse over button, I have used mousemove and mousedown events of canvas. Button images while generated are being assigned with a callback function for triggering actions on that button.

**Game UI Customization:**

Game UI customization is simple. If you need to integrate a new theme, design individual components of the game in assets/game and assets/gui folder and replace them with the same name. If your components have a different dimension, you must open GameUI.js file and search for that file/image name.

**Adding new button:**

In case you must add a new button, add it in the required folder and create a new entry in the GameUI.js file and use it as per the requirement. You must give a callback method for that button and make that method in the GameEngine.js file.

**Adding sounds:**

The sounds used in this game are being recorded using my android phone. I have created a sound class named Sounds.js. You can initiate the sound and play it whenever needed like below.

this.rollingSound = new sound("assets/audio/rolling.mp3");

this.rollingSound.play();

**Game Features:**

This is the basic game of an Ancient Indian Dice Game Concept.

It has 6 places to bet and maximum bet limit is set to $5 now.

You can set the maximum bet limit to $1 or $10 or any other denominations but you have the design the coin image for the same denomination.

Set the betCoinValue in GameEngine.js file.

Bonus dice roll has been applied when a win has 2 Love or 3 Diamonds and user must have a bet on them to win a bonus game. Bonus game has an coin falling animation with a message to the user.

**Game Play Customization:**

An intermediate JavaScript developer can easily customize this game with basic Object Oriented Development skills.

To adjust the speed of the dice with the rolling sound of your own use “rotIncrement” variable in dice class.

**How the Dice Work:**

Dice class allows few parameters

1. size: This is the size of the dice, you can set any number while initializing the Dice Instances.
2. rand: This is a random number at which the dice has to be stopped. Pass this number from a server generated instance and verify the same from this game to the server instance for any payout to stop illegal hack.
3. rotEnabled: This is to stop and play the dices.
4. img1: This is any kind of image either PNG, JPG for the first side of a dice.
5. img2: This is for the second side of the dice.
6. img3: This is for the third side of the dice.
7. img4: This is for the fourth side of the dice.
8. img5: This is for the fifth side of the dice.
9. img6: This is for the sixth side of the dice.
10. x: Position of the dice on the canvas horizontally in pixels.
11. y: Position of the dice on the canvas vertically in pixels.
12. ctx: Dice takes context of the canvas as the last parameter to render and animate.

It has animate method which calls updateFrameFunc method which generates different sides with transformations applied to each side to make the dice look 3d.

To set delay of rotation, please try with different numbers in GameEngine.js at this line below,

dice.rotations = { x:dice.tempRot.x-10, y:dice.tempRot.y-10, z:dice.tempRot.z-10};

Browsers Supported:

1. Google Chrome
2. Firefox
3. Opera
4. IE (Game must be served from a server otherwise Dices will not be rendered)

**Features Pending for development:**

1. Handling multiple denominations of coins
2. Adding some more animations
3. Integrating advertisement
4. Integrating asset preloader
5. Creating a reusable Button class
6. Integrating Local Storage
7. Score submission to DB with user authentication
8. Fixing issues on Safari for Windows. Safari for MAC not tested.

**Features available on demand**

1. Integration to server (PHP or NodeJs)
2. Payout probability calculation
3. Multi player betting game

**Contact Us:**

For any customization or integration related help you can reach out to me at [Mitrabinda.jvmba@gmail.com](mailto:Mitrabinda.jvmba@gmail.com). We are available for freelance work as well for any web related requirements.

Key words:

Casino, “gudu gudu palli”, craps, dice, “JavaScript dice class”, “dice games” , “Canvas 3d Dice”, “JavaScript Dice”, “casino game”, “dice roll”, “canvas casino game”, “html5 game code”