

## ② Spin & Win (Arrays & Maths)

Here to build this game we'll use 'phaser' framework.

Things we'll work on in this game:

- Game loop in phaser
- Add Images
- Event Listeners
- Animations
- Arrays
- Mathematics
- Sounds
- Object oriented Prog.

### Basic skeleton of phaser:

(.js)

let config = {  
 type: Phaser.CANVAS,  
 width: 800,  
 height: 600,  
 scene: [ ]  
};

by def. type: Phaser.CANVAS,  
but we get black bgnd.

It fetches canvas API which enables to render game on web browser.

It basically means level of the game.

scene: [ ]  
 preload: yourpreloadfunction(),  
 create: yourcreatefunction(),  
 update: yourupdatefunction(),  
};

now acc. we'll define these fns.  
kinda game loop

let game = new Phaser.Game(config); ①

on Game[config]

↳ W  
↳ H  
↳ etc

↳ scene

↳ load "of Assets"  
↳ create (to create assets and)  
↳ cont. update of diff obj's

music  
img  
// etc

## # load & Add Sprites

predelined

So inside config we have further many objects, like height, width etc. Just like this inside config we've scene, which further contains many pre defined functions like load, create etc. Now further these functn (load, create) have functions to set audio, image etc for our game.

for audio: this.load.audio();  
write this in preload fn().

in preload: this.load.image(key, path);

(ex) this.load.image('background', '../Assets/back.jpg');

↑  
this means go back one folder, then in Assets & select back.jpg

• Now after load it we can display the image in create fn().

to set image  
wrt center  
of screen

let w = game.config.width;  
let H = " " height;

let background = this.add.sprite(0, 0, 'background');  
background.setPosition(w/2, H/2);

background.setScale(0.20); → w/h ke resp me scale / kam krna bg ko.

Suppose we had 2 imgs. ab age konsi

img ko agha & piche kisko, can be decided by 'depth' fn.

pin.depth = 1; → '1' max den & '0' pass.

Suppose pin  
to be another img  
like background! → then this would mean background age to pin  
peche.



to create event listener in phases

\* write in create():

this.input.on ("pointerdown", spinwheel, this);

event, this means  
click down

↑  
click krne pe kya hoga.  
our customized fn.

To obj. ko edit  
or modif. krna hai ake  
click kr.

then we can define our  
customized fn() in main body!

# To display text on screen

\* write in create():

font\_style = {

font: "bold 30px Roboto",

align: "center",

color: "red",

}

this is  
scene. As

create, update()

ye sare for scene obj me call

honge. ∴ this is reference obj for scene!

this.game\_text = this.add.text(10, 10, "welcome to  
spin wheel,

font\_style)

## # Animation & tweens

function spinwheel () {

  tween = this.tweens.add ( {

    target: this.wheel, → is pr elct aega.

this.wheel will  
go from 0 to  
800 angle  
in 3000 ms.

angle: 800, →

ease: "cubic.easeOut", →

duration: 3000,

phrasa elct jisse ing  
disruptly nli rule

onComplete: function () {

  console.log ("You won");

}

});

}

} duratn khtm hone  
ke baad ye fr  
chalega.

```

let prizes - config = {
  count: 12;
  prize - name = (" --- ");
}

```

```

let config = {
  type: Phaser.CANVAS,
  width: 800,
  height: 600,
  scene: {
    preload: preload,
    create: create,
    update: update,
  }
};

```

→ here we can also write "Phaser.AUTO," this automatically determines what's best CANVAS on webgl.

```

let game = new Phaser.Game(config);

```

```

function preload() {
  console.log("Preload");
  this.load.image('background', '../Assets/back.jpg');
  this.load.image('wheel', ...
    " ('pin', ...
    " ('stand', ...
}

```

} load all images



function create() {

★ ~~pls~~ look up the github repo  
for test code!