
Game Development Workshop I

MIT S Gwalior

Sanket Singh | Coding Blocks



Welcome!

Welcome to the Game Development Class.

- **Javascript And Phaser Basics**

Understand the basics of JS & Phaser.

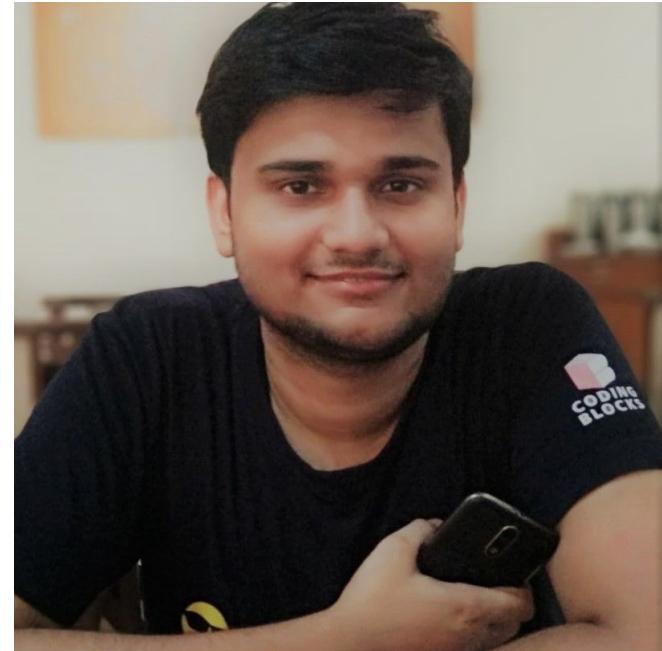
- **Space Shooter**

An Interactive Ultra Casual
Game.

About Me

Sanket Singh

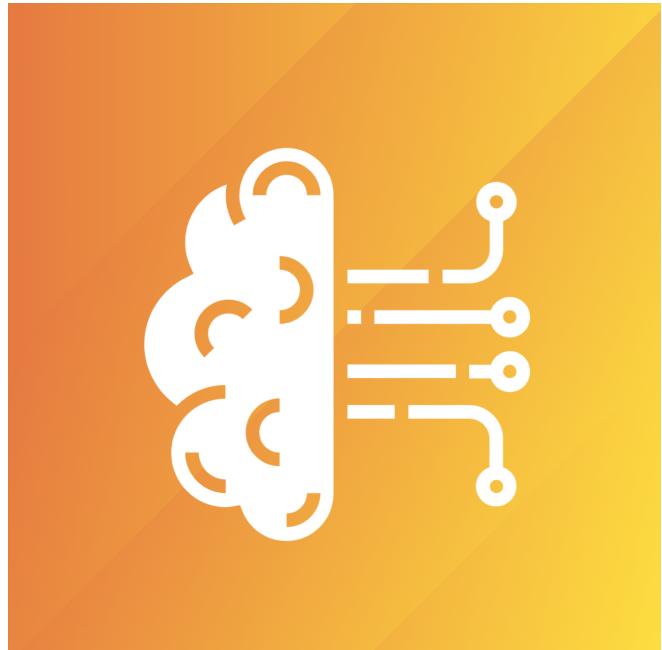
- B.Tech in Computer Science from MAIT
- Former Frontend Engineering Intern At FableHQ
- Former Full Stack Developer At Kakcho Fashions
- Problem Setter, Reviewer At Talview
- Won Infosys Digital Make-a-thon And Codechef Rank 1 In Long Challenge
- Instructor And Developer At Coding Blocks



Game Development

One of the most bright and trending thing in the market

Making games is a creative and technical art form



JavaScript

One of the most popular programming language now-a-days

Backbone of WebDevelopment from years

Loaded with tons of open source frameworks like PhaserJS,
TensorflowJS, ReactJS, AngularJS, VueJS etc.

Ranked X at the tiobe index and it's popularity is increasing day by day



PhaserJS

One of the most famous javascript based Game Development framework

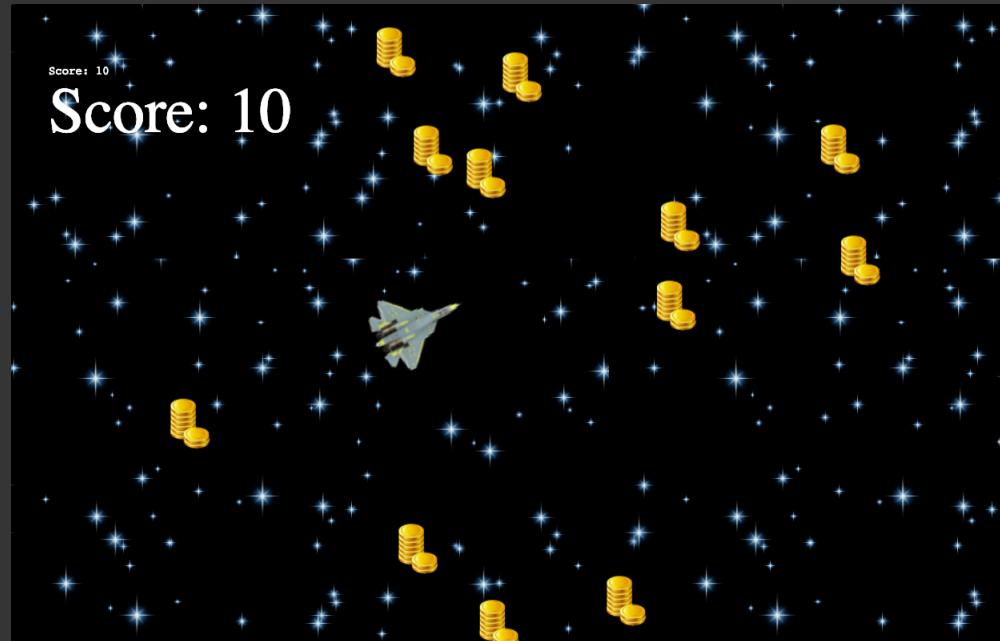
Includes physics engine support

Can be easily used to make ultra casual and HTML based games

We will create a fun project using this



What we are going to build?



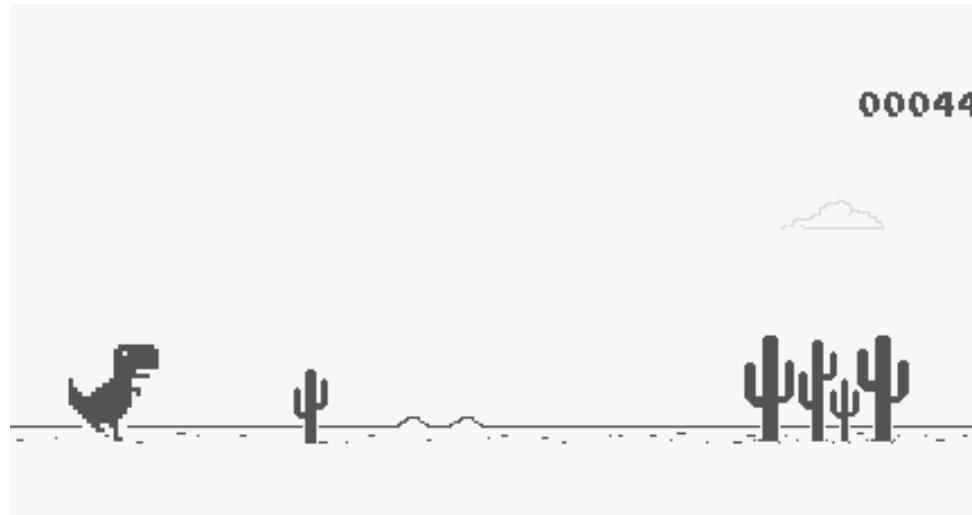


Requirements

You should have pre-requisites fulfilled.

- Any Browser
- Any Code Editor (VS Code Preferable)
- Basic Knowledge of programming in any language (JS Preferable)

Ultra Casual Games???



What elements are you able to visualise in the game?

1. Background Space
2. Spaceship/player
3. Coins
4. Bullets
5. Score board and Score value
6. Controls by cursor and shooter

Setup Phaser Environment

```
game = new Phaser.Game(width, height, Phaser.CANVAS, '<GameName>', {some essential functions})
```



The 4 Essential Functions

1. Preload
2. Create
3. Update (25 fps)
4. Render (50-60 fps)

How to preload the assets

For all the corresponding elements of the game we need to preload the asset either an image or an audio in order to use them in our game.

```
game.load.<image/audio>(<filename>)
```



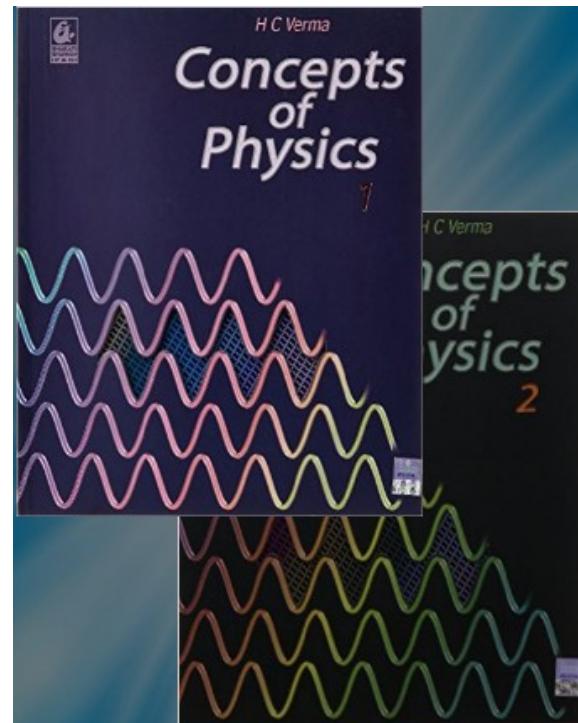
Establish Physics Engine

By saying physics in phaser we mean that

Types Of Physics Engines In PhaserJs

1. Arcade (Phaser.Physics.Arcade)
2. Ninja
3. BOX 2d
4. ChipMunk

game.physics.StartSystem(<Physics Name>)





HTML5
< canvas >

HTML CANVAS

- <canvas> is an HTML element which can be used to draw graphics via scripting (usually JavaScript). This can, for instance, be used to draw graphs, combine photos, or create simple (and not so simple) animations.

Sprite

At its most basic a Sprite consists of a set of coordinates and a texture that is rendered to the canvas.

```
Player = game.add.sprite(starting_x, starting_y, 'asset_name')
```



TileSprite

TileSprite is a Sprite that has a repeating texture. The texture can be scrolled and scaled independently

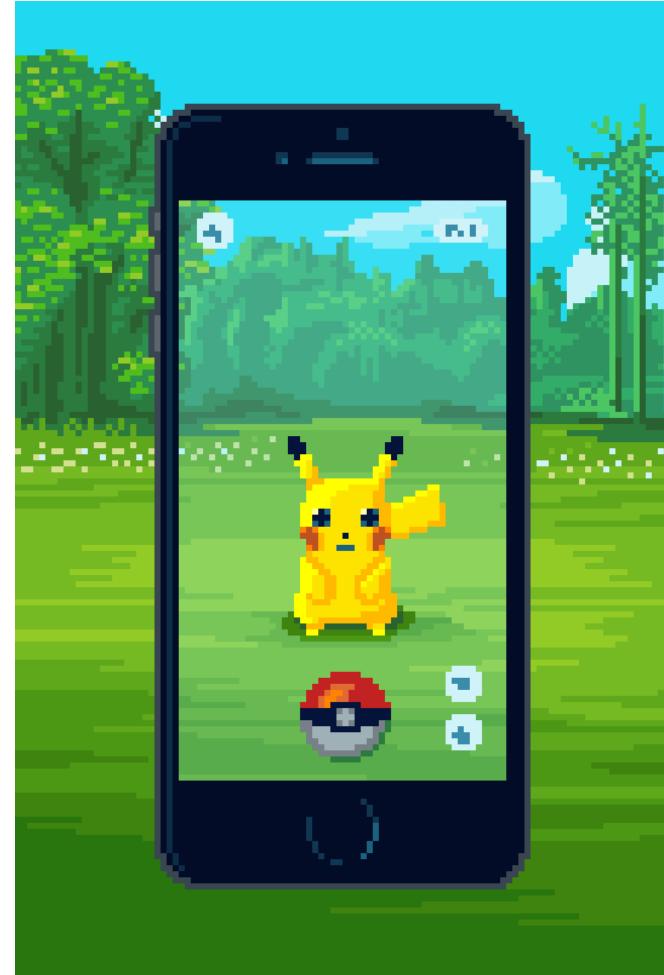
```
space = game.add.tileSprite(0, 0, window.innerWidth, window.innerHeight, <asset-name>)
```

```
// We can make it in motion by updating the tilePosition.<x/y> property
```

Enable Arcade Physics

This will create an Arcade Physics body on the given game object or array of game objects.

[https://phaser.io/docs/2.6.2/
Phaser.Physics.html](https://phaser.io/docs/2.6.2/Phaser.Physics.html)
`game.physics.arcade.enable(<object_name>)`



Add Text and Sound

```
Variable = game.add.text(x, y, "Initital String", {fill: "Color hex",  
font: "80px Arial"})
```

```
Variable = game.add.audio('asset_name')
```



Adding Groups In Phaser

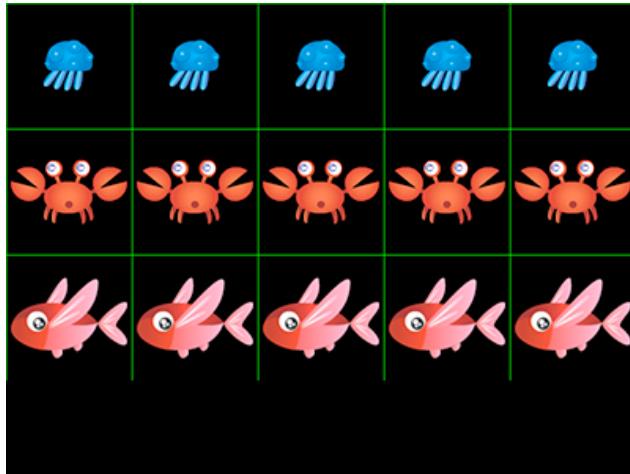
A Group is a container for [display objects](#) including [Sprites](#) and [Images](#).

Groups form the logical tree structure of the display/scene graph where local transformations are applied to children.

For instance, all children are also moved/rotated/scaled when the group is moved/rotated/scaled.

In addition, Groups provides support for fast pooling and object recycling.

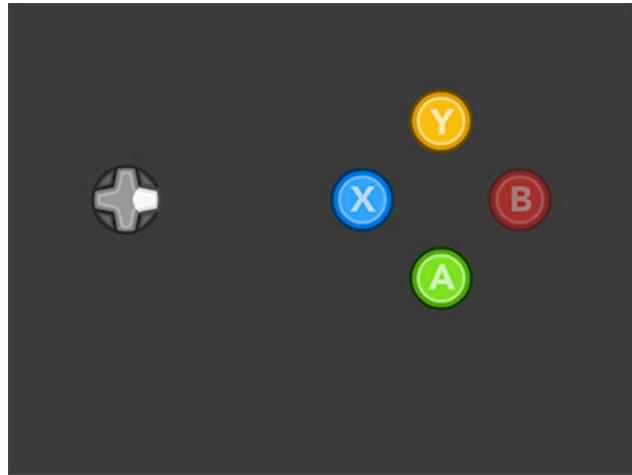
Groups are also display objects and can be nested as children within other Groups.



```
variable=game.add.group()  
variable.scale.set(<value>)  
// Create Items of group  
Iterate and create new item using  
Item = variable.create(x, y, 'asset_name')
```

Adding Controllers

```
variable=game.input.keyboard.createCursorKeys()  
variable=game.input.keyboard.addKey(Phaser.Keyboard.SPACEBAR)
```



Adding Speed To Sprite

```
player.body.velocity.<x, y> = <multiplier>*(game.input.<x, y> - player.<x, y>)
```



Adding bullet and weapons

<https://phaser.io/examples/v2/weapon/bullet-frame-cycle>



Collision Logic

<https://phaser.io/docs/2.4.4/Phaser.Physics.Arcade.html#collide>



Thank You!

Code is available - cb.lk/firstgame

Maintain The Player At Mouse Tip

We will use [Math.atan2\(\)](#) func.

```
Math.atan2((game.input.y - player.y),(game.input.x - player.x)) * 180 / Math.PI + 90;
```

