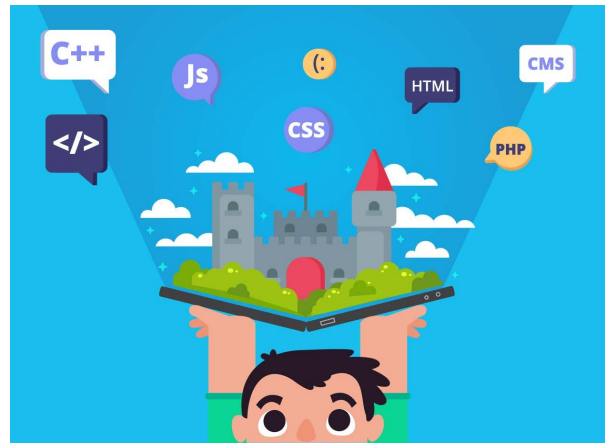# CHECKPOINT REVISION CLASS: PHYSICS ENGINE

What is our GOAL for this MODULE?

In the past few classes, you learned to create games using Physics Engine. Today was a capstone class, to revise and rewrite all the concepts learned in the last few classes.

What did we ACHIEVE in the class TODAY?

- Practiced and implemented programming constructs to strengthen concepts by creating a game using Physics Engine.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Variables - to keep score, count, objects.
- Conditional Programming- to add different scenarios to the game.
- Loops- to repeat the sequence of commands.
- Classes - to create a blueprint of an object.
- Objects - to create new objects for the game.

How did we DO the activities?

1. Break down the task into smaller parts.
2. Add the physics engine library **matter.js.min** to the index.html file.
3. Import physics engine libraries in the sketch.js file.

```
const Engine = Matter.Engine;
const World = Matter.World;
const Bodies = Matter.Bodies;
const Constraint = Matter.Constraint;
```

4. Create a canvas of the desired size and create an engine and world for the game.

```
function setup() {
   createCanvas(3000, 800);
   engine = Engine.create();
   world = engine.world;
```

5. Declare required Global variables which can be used across the program.
6. Load images, animation or sound files using function **preload().**
7. Create a **class** of desired objects and add dimensions to the class in the **constructor().**
8. In the **display()** function display the object. An example of **Box** class is given below for quick reference.

```
class Box {
  constructor(x, y, width, height) {
    this.body = Bodies.rectangle(x, y, width, height, options);
    this.width = width;
    this.height = height;

    World.add(world, this.body);
  }

  display(){
    var pos =this.body.position;
    var angle = this.body.angle;
    push();
    translate(pos.x, pos.y);
    rotate(angle);
    rectMode(CENTER);
    strokeWeight(4);
    stroke("green");
    fill(255);
    rect(0, 0, this.width, this.height);
    pop();
  }
}
```

9. Using the **new** keyword, create the object using the class inside **setup()** of **sketch.js**.
10. Display the object in function **draw().**
11. Create a logic of the game in function **draw().**
12. Use conditional programming to add different scenarios to the game.
13. Add background images for different scenes.
14. Use **mouseDragged(), mouseRelease()** function to add mouse control to the game.
15. Use the **keyPressed()** function to add the key controls to the game.
16. Once the game is complete, run the code to see the desired output.
17. Debug and fix errors, if any, in our code.
18. Upload the code into GitHub.

What's NEXT?

In the next class, you will be introduced to Firebase Database to create multiplayer games.

Extend Your Knowledge:-

1. Explore different usage of the physics engine from the following documentation.
   https://brm.io/matter-js/
   licensed by MIT and Copyright (c) Liam Brummitt and contributors