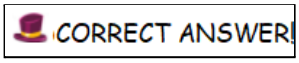
Difference between C29 1:1 & C30 1:4	1. Added different game from constraint concept of 1:1 2. Provided the template code with partial code written 3. Introduced mouseDragged and Released function
Topic	**FRUIT HANGING WITH ROPE**
Class Description	**The students will learn to create a rope body using the matter.js library. The students will also attach a fruit to a rope using constraints.**
Class	**PRO-C30**
Class time	**60 mins**
Goal	● Create a rope body ● Attach the fruit with the rope ● Review the concepts covered in the past few classes
Resources Required	● Teacher Resources ○ Laptop with internet connectivity ○ Visual Studio Code ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ Laptop with internet connectivity ○ Visual Studio Code ○ Earphones with mic ○ Notebook and pen

Student Motivation and Engagement	You will see the following recommendations for student motivation. • **Hats-off**: Specific instructions for giving hats-off will be provided in the lesson. • **Concept Magnifier**: Used to highlight new concepts and connect them with real-life examples. • **Knock-Knock!**: To nudge the students to make sure they are attentive. • **Thinking Caps**: Used to engage the students for an activity or Q&A. • **All types of Quizzes**: Includes revision quizzes, riddles, and pop-up quizzes. • **Candy Boosters**: Used to motivate the students to do better in the activities. • **Important Points to Remember**: To highlight important concepts.
Clearing Doubts	• Try to clear students' doubts then and there. • If the question is irrelevant at the time, let the student know that politely. • At the end of the class, ask the student/s who joined the class a little late, if he/she has understood everything and whether he/she has any questions. Guide the student to the activity video link to go through the missed class.

WARM-UP (5 Minutes)	
Do	**Say**
Welcome students to the class.	Hello! It is so great to see you all. Q How are all of you doing?

Refer Teacher resource 1- Page/Slide 1 for questions.	A Varied. Q Are you all excited about today's class? A Yes.
Start the revision of the previous class. Refer Teacher resource 1- Page/Slide 2 for questions. *Mute all the students.* *Show the following images/words to the students and call out one name at a time.*	In C29 class, we have completed the pirate game Let's start with a quiz game to revise some important concepts from the last class. I am going to show you some questions, and you have to tell me their answers. I'll call out a name, and the student has to answer the question.
Refer Teacher resource 1- Page/Slide 3-5for questions.	
Ques 1: Why do we use the **isSink** state? A. **isSink** is used to store the value as true when the cannonball is sinking. B. **isSink** is used to store the value as false when the cannonball is sinking. C. **isSink** is used to store the value as true when the cannonball is not sinking. D. **isSink** is used to store the value as false when the cannonball is not sinking.	Ans: A: **isSink** is used to store the value as true when the cannonball is sinking.
Ques 2: Name the custom alert boxes made using JS, which we used in the previous class?	Ans: SweetAlert - swal()

Applaud all the students who answered correctly and give a Hats-off. *Refer Teacher resource 1- Page/Slide 6 for images*  	Well done, all of you! Each one of you gets a Hats-off for giving the correct answer.

	*Tick-Tock!* **Time completed: 5 Minutes** **Time remaining: 55 Minutes**

TEACHER ACTIVITY-1 **(25 Minutes)**	
Do	**Say**
Teacher can open Teacher Reference 1 and create excitement for the student. *Refer Teacher resource 1- Page/Slide 7-8 for images* *Teacher can download the code template for teacher activity 1.* *Note: Teacher needs to explain the code lines which is already added to the template*	In class C29, we got started with the physics library **matter.js** and built a Pirate Invasion with it. In this class, we are going to create **Fruit Hanging with Rope**. Q Can anyone identify the objects in the game? A Yes. In this game, we have a watermelon fruit hanging from the rope. We have the basket at the bottom to collect the fruit.

	When the user clicks on the rope, it will cut the rope and the watermelon falls down.
	We have a template of code ready with us. Let's first create the ground body and for that, we will need to create the ground class. ● Create a new file and name it **ground.js** ● Now, add this file in the **index.html,** so that we can use it in our code

1. Add the file inside the **sketch.js**

```
body>
 <script src="matter.min.js"></script>
 <script src="p5.play.js"></script>
 <script src="rope.js"></script>
 <script src="ground.js"></script>
 <script src="sketch.js"></script>

/body></html>
```

Refer Teacher resource 1- Page/Slide 9 for images	Open the **ground.js** file, here we are going to create the **Ground** class. The ground will be a stationary body, so we only need to create the **constructor()** function and the function to **display** the ground. In the **constructor**, we will specify the **x**, **y** positions, and the **width** and **height**,

	which will be entered by the user while creating the **ground** object for the **ground()** class.

2. Teacher recall the **ground.js** class which is already added from previous code.

```
js > JS Ground.js > Ground
class Ground{

    constructor(){
        var ground_options={
            isStatic : true
        }

        this.ground = Bodies.rectangle(450,390,900,20,ground_options)
        World.add(world,this.ground);
    }
    display(){
        noStroke();
        fill(188,67,67);
        rectMode(CENTER);
        rect(this.ground.position.x,this.ground.position.y,900,20);
    }
}
```

	The first step is complete. Here, we have created a **ground** class and **ground** object. Now, we can see the ground body on the canvas. In the next step, we need to create the rope and the fruit body.
	To make the fruit hang with the rope we need a **constraint**. We have already learned to create a constraint using **Matter.Constraint.create() in class C27**.

	Let's quickly '**namespace**' our **Matter.Constraint** to be called Constraint. Q What is a namespace? A A **namespace** is nothing but the name that is used to identify and refer to objects We are doing this so that we do not have to call **Matter.Constraint** each time in our code. We can simply call it Constraint. This is not compulsory but makes our code easier to read and write.

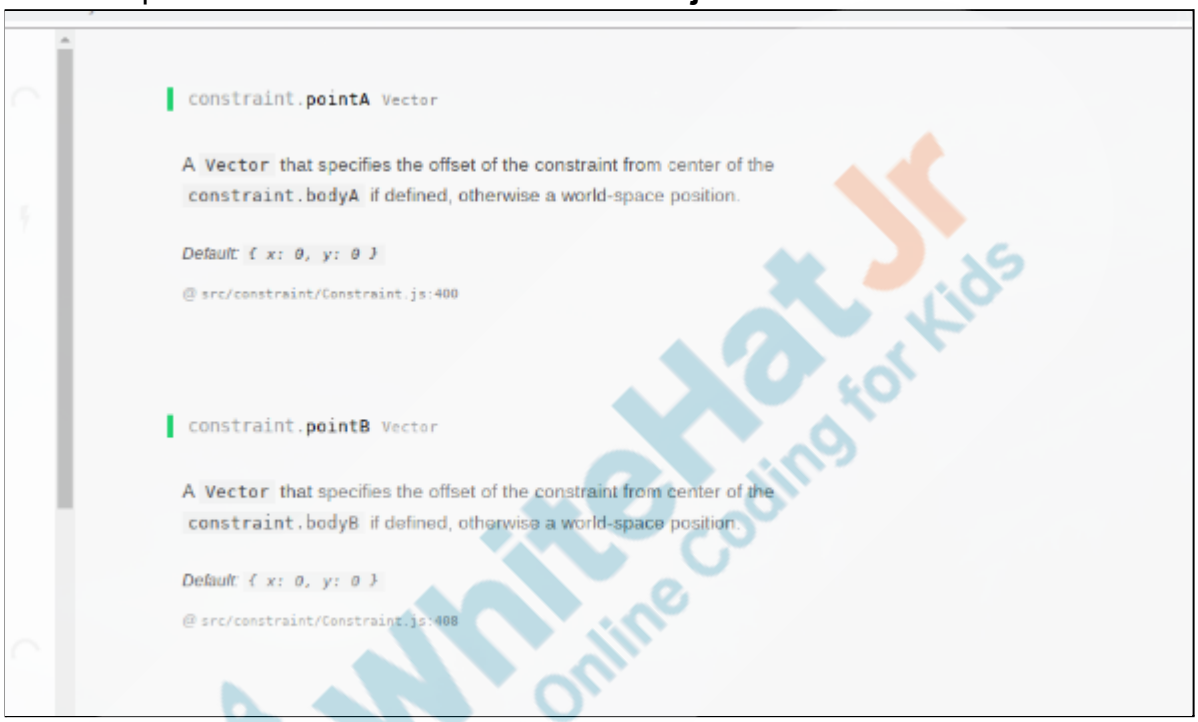
3. Add the namespace for **Matter.constraint**

```
JS sketch.js > setup
1   const Engine = Matter.Engine;
2   const World = Matter.World;
3   const Bodies = Matter.Bodies;
4   const Constraint = Matter.Constraint;
5
```

Teacher can open the [Teacher Reference 2]	Now let's create an option to create a **constraint**. We need a fruit to be attached to and projected from a point. Let's see if we have that option in the **Matter.Constraint**. Q Do you remember the option in the **Matter.Constraint**, which helps us to attach the fruit to the point?

	constraint.pointA

4. Find the option to create the constraint in **Matter.js**



	Exactly! Let's create a separate class called **Slingshot** and pass the coordinates of a point as one of the parameters to the **Slingshot() constructor**. We can also include the **stiffness and length**.

5. Add the **constructor** to create the constraint.

```
JS SlingShot.js > Slingshot > constructor
class Slingshot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 100
        }
```

Teacher can open [Teacher Reference 2](#) and ask the student to find the option to create a constraint	Ok, now let us use this option to create the constraint. Q Can anyone tell me the syntax to add options to the constraint? A Varied

6. Find the options to add the **options to the constraint**

```
Matter.Constraint.create(options) → Constraint
```

Creates a new constraint. All properties have default values, and many are pre-calculated automatically based on other properties. To simulate a revolute constraint (or pin joint) set `length: 0` and a high `stiffness` value (e.g. `0.7` or above). If the constraint is unstable, try lowering the `stiffness` value and / or increasing `engine.constraintIterations`. For compound bodies, constraints must be applied to the parent body (not one of its parts). See the properties section below for detailed information on what you can pass via the `options` object.

7. Add the **options to the constraint**

```
SlingShot.js > Slingshot > fly
class Slingshot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 100
        }
        this.sling = Constraint.create(options);
        this.pointB=pointB;
```

We need to add this constraint to our **world** just like we did to other bodies eg: ground in our game.

8. Add the constraint to the **world**

```
SlingShot.js > Slingshot > fly
class Slingshot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 100
        }
        this.sling = Constraint.create(options);
        this.pointB=pointB;

        World.add(world, this.sling);

    }
```

We also need to display the constraint. We can use a separate function called **display()**.

Q We need to draw a line to connect between the point and the fruit. Can you

	tell me which instruction we can use here? **A line() instruction**

9. Add the **line() instruction** to make the connection between two objects

```
display(){
    if(this.sling.bodyA){
    var pointA = this.sling.bodyA.position;
    var pointB = this.pointB;

    strokeWeight(4);
    stroke("turquoise");
    line(pointA.x, pointA.y, pointB.x, pointB.y);
    }
```

	Great! We have a template of the game ready with us. Now, you are going to fill in the missing blocks. I will help you through this.

	*Tick-Tock!* **Time completed: 30 Minutes** **Time remaining: 30 Minutes**

STUDENT ACTIVITY - 1 **(25 Minutes)**	
Do	**Say**
Student opens ***Student Activity 1.*** *Refer Teacher resource 1- Page/Slide 10-11 for images*	We have a template ready with us. We need to write the missing code. Let's first create a fruit object.

Challenge 1: *Guide the <student name1> to find the missing code to create a fruit object. Ask the student to share the code in the chatbox.* *Teacher can explain if the student is struggling to code.* *Ask slow learner*	**Challenge 1:** We are provided with the condition to create a fruit body. Now, you need to find the missing value. **Solution:**

10. Create the ball bodies using a **circle**

```
JS sketch.js > draw
12  function preload(){
13    fruit=loadImage("melon.png");
14    g=loadImage("basket.png")
15  }
16  function setup() {
17    createCanvas(900,400);
18    engine = Engine.create();
19    world = engine.world;
20    Engine.run(engine);
21    ground = new Ground();
22
23
24
25    //ball holder with slings
26    ball = Bodies.circle(50,200,20);
27    World.add(world,ball);
28
```

Challenge 2: *Guide the <student name2> to find the parameters that need to come inside the **Slingshot()** constructor.* *Ask the student to share the code in the chatbox.* *Teacher can explain if the student is struggling to code.*	**Challenge 2:** Find out the parameters that need to come inside the **Slingshot()** constructor. The coordinates can be anywhere you want the **slingshot** to appear.

11. Find out the parameters that need to come inside the **Slingshot()** constructor.

```
sketch.js > draw
  fruit=loadImage("melon.png");
  g=loadImage("basket.png")
}
function setup() {
  createCanvas(900,400);
  engine = Engine.create();
  world = engine.world;
  Engine.run(engine);
  ground = new Ground();



  //ball holder with slings
  ball = Bodies.circle(50,200,20);
  World.add(world,ball);

  slingShot = new Slingshot(this.ball,{x:100,y:100});

}
```

*Teacher opens the doc from Teacher Activity 3 and shows the students about the **mouseDragged** option*	The fruit seems to be constrained to a point now. We want it to move only when we are dragging the mouse. Q How do we do that? Any ideas? A Varied
Teacher opens the doc from *Teacher Activity 4 can guide the student to set the position*	We will use a function called **mouseDragged** for this. To implement **mouseDragged** inside the **matter.js** we can use **setPosition().**

12. Find the **setPosition** function

```
Matter.Body.setPosition(body, position)
```

Sets the position of the body instantly. Velocity, angle, force etc. are unchanged.

Parameters

body	Body
position	Vector

@ src/body/Body.js:416

Challenge 3:
Guide the <student name3> to find the parameters need to come inside the ***setPosition()***
Ask the student to share the code in the chatbox.
Teacher can explain if the student is struggling to code.

Challenge 3:
Find out the parameters that need to come inside the **setPosition()**

Hint:
The parameter will be a body that needs to be moved and its position is based on mouse movement of **x and y.**

13. Declare the value inside the **setPosition** to make the **fruit move**

```
function draw() {
  background(56,44,44);

  //Engine.update(engine);
  //text(mouseX + ',' + mouseY, 10, 15);

  ground.display();
  g.scale=.025;


  imageMode(CENTER)
  image(fruit ,ball.position.x,ball.position.y,40,40);
  image(g,450,270)

  slingShot.display();
}
function mouseDragged(){
  Matter.Body.setPosition(this.ball,{x:mouseX,y:mouseY});
}
```

Refer Teacher resource 1- Page/Slide 12 for images	When we release the mouse it moves **to and fro** but is still connected to the point. We now just want to make the **fruit detach** from the **constraint** when the mouse is released. Similar to **mouseDragged()** we have **mouseReleased()**. Inside the function **mouseReleased()**, **slingShot.fly()** is called.

14. Create a function for **mouseRelease()**

```
function mouseReleased(){
  slingShot.fly();

}
```

Challenge 4: *Guide the <student name4>to find the*	**Challenge 4:** We have defined **slingShot.fly()** inside the **slingShot class**. Find the missing value to

missing value to detach the fruit. *Ask the student to share the code in the chatbox.* *Teacher can explain if the student is struggling to code.*	detach the fruit. **Hint:** Attaching nothing to **bodyA** will free the fruit from the constraint. '**null**' implies nothing in javascript.

15. Add the code to detach the fruit

SlingShot.js > Slingshot > display

```
class Slingshot{
    constructor(bodyA, pointB){
        var options = {
            bodyA: bodyA,
            pointB: pointB,
            stiffness: 0.04,
            length: 100
        }
        this.sling = Constraint.create(options);
        this.pointB=pointB;
        World.add(world, this.sling);

    }
    fly(){
        this.sling.bodyA =null;
    }
```

Refer Teacher resource 1- Page/Slide 13 for images	Amazing!
Refer Teacher resource 1- Page/Slide 14-15 for images *Teacher can show the image added for the game*	We have completed the game. Before executing, let's check the image added for the game.

16. Show the image added inside the preload function.

```
function preload(){
    backgroundImg = loadImage("background.png");

 fruit=loadImage("melon.png");
 g=loadImage("basket.png")
}
```

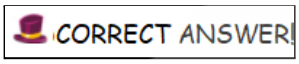
17. Display the image inside the **draw()** function

```
function draw() {
  background(backgroundImg);

  //Engine.update(engine);
  //text(mouseX + ',' + mouseY, 10, 15);

  ground.display();
  g.scale=.025;


  imageMode(CENTER)
  image(fruit ,ball.position.x,ball.position.y,40,40);
  image(g,450,270)
```

Output:

	Tick-Tock! **Time completed: 55 Minutes** **Time remaining: 5 Minutes**

WRAP-UP (5 Minutes)	
Do	**Say**
Refer Teacher resource 1- Page/Slide 16-17 for images *Refer Teacher resource 1- Page/Slide 18-20 for images*	We learned to create a constraint between the fruit and the rope. Q Choose the correct option to sets the position of the body instantly. A. Matter.Body.setPosition(this.ball,{x:mouseX ,y:mouseY})

Ask everyone. *Ask students to type the answer to the questions in the chat window.* *Observe students and see which student is not answering and encourage them to answer the next question.* *Encourage students to answer every question.*	B. Matter.setPosition(this.ball,{x:mousex,y:mousey}) C. Matter.Body.setposition(this.ball,{x:mouseX,y:mouseY}) D. Matter.setPosition(this.ball,{x:mouseX,y:mouseY}) A Matter.Body.setPosition(this.ball,{x:mouseX,y:mouseY}) Q Which method is used to create a new rigid body model with a circle? A Matter.Bodies.circle()
Applaud all the students who answered correctly and give a Hats-off. *Refer Teacher resource 1- Page/Slide 21 for images*  	Well done, all of you! Each one of you gets one Hats-off for giving the correct answer. Great!
Inform students about what they will be doing next week. *Refer Teacher resource 1- Page/Slide 22 for images*	Next class, we'll be working to create a new game "**multiplayer car racing**"
Give information about the project. *Refer Teacher resource 1- Page/Slide 23 for images* *Create excitement in the students so that they are motivated.* *Make sure the students understand what they are expected to do in the project.*	**BLOWER PIPE** In class 30, we learned how to apply force on the body by pressing a mouse button. In this project, we will create a blower pipe game using the same concepts.

Teacher Resource - Project Solution	**Story:** Jenny had been to a carnival fest. There she saw a toy which was quite interesting, as to win she had to continuously keep blowing into the pipe to keep the ball in the air. When she got home, she thought of creating that toy virtually. Can you help her make that toy? I am very excited to see your project solution and I know you will do really well. Bye Bye!

ADDITIONAL STUDENT ACTIVITIES - FOR THE FAST LEARNERS	
Do	**Say**

Encourage the students to write reflection notes in their reflection journals using markdown.	**Additional Activity I**
	
	Why don't you all write some reflection notes in your reflection journal using markdown?
	
	Use these as guiding questions while writing:
	• What happened today?
	  ○ Describe what happened.
	  ○ The code I wrote.
	• How did I feel after the class?
	• What have I learned about programming and developing games?
	• What aspects of the class helped me? What did I find difficult?

LINKS

Teacher Resource 1 (With Speaker Notes)	Teacher Resource 1 (With Speaker Notes)	https://s3-whjr-curriculum-uploads.whjr.online/082405fe-931f-449a-bf4f-9cc1e8c1a0de.html
Teacher Resource 2 (Without Speaker Notes)	Teacher Resource 2 (Without Speaker Notes)	https://s3-whjr-curriculum-uploads.whjr.online/accf4345-8e4f-4356-b61a-47c14f0b5c5a.html
Teacher Activity 1	Boilerplate code	https://github.com/rashmi-sathya/Fruit-and-rope
Student Activity 1	Boilerplate code	https://github.com/rashmi-sathya/Fruit-and-rope
Teacher Reference 1	Student activity solution	https://rashmi-sathya.github.io/fruit-and-rope-final/
Teacher Reference 2	Matter.js documentation	https://brm.io/matter-js/docs/classes/Constraint.html
Teacher Reference 3	Mouse Dragged	https://p5js.org/reference/#/p5/mouseDragged
Teacher Reference 4	Set position	https://brm.io/matter-js/docs/classes/Body.html
Teacher Reference 5	Output	https://github.com/rashmi-sathya/fruit-and-rope-final
Teacher Resource-Project Solution	Project Solution	https://github.com/rashmi-sathya/project-solution-blower-pipe