**TEMASEK POLYTECHNIC**

**SCHOOL OF INFORMATICS & IT**

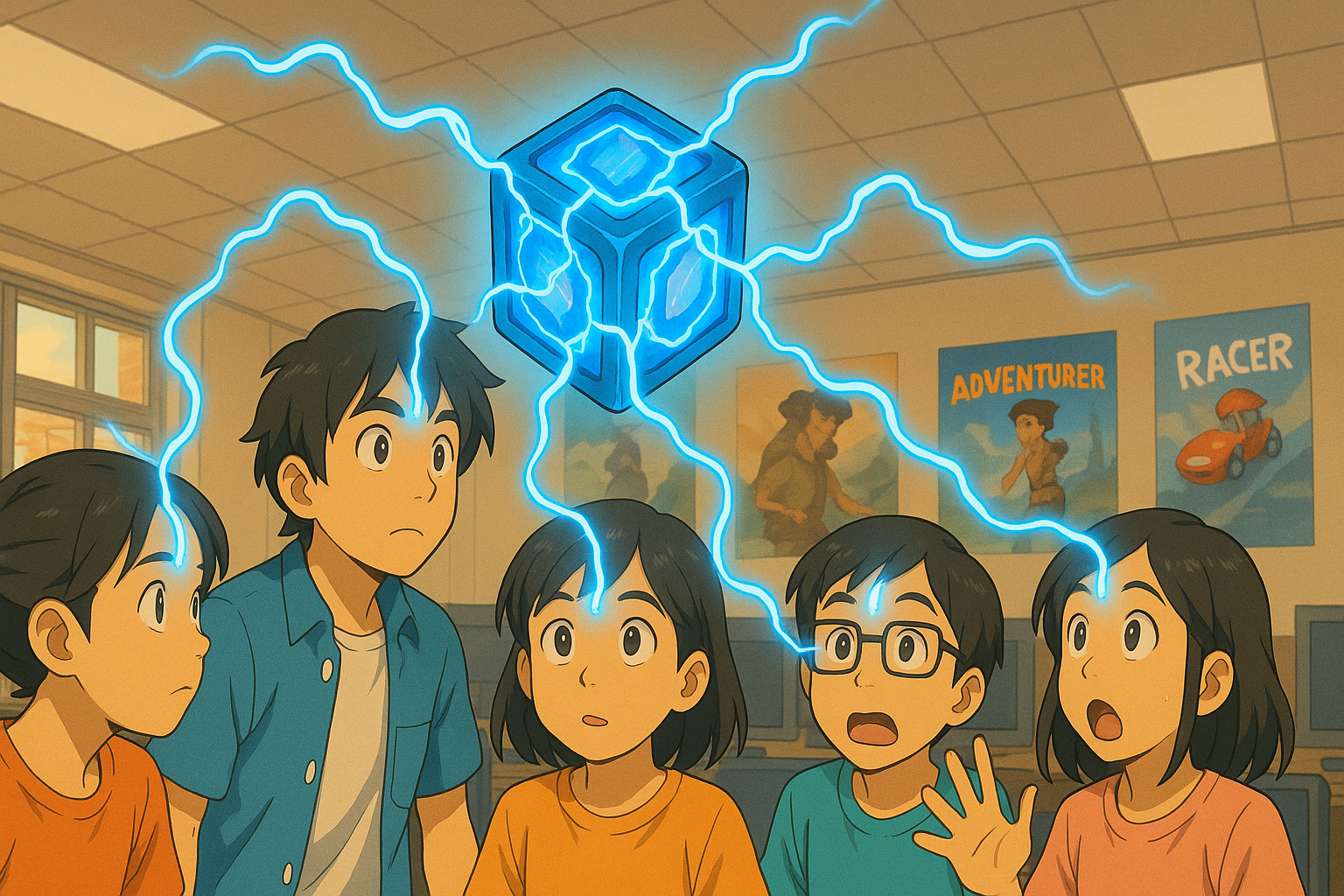
**DIPLOMA IN IMMERSIVE MEDIA & GAME DEVELOPMENT**

**AY2025/2026 APRIL SEMESTER**

**GADV (CGE2C25)**

**Unity Physics Worksheet**

To see the additional comments and resources, make sure you select **All Markup** in the **Review/Tracking** pane



**QUESTION 1 (basics of Unity physics)**

1. What is the core responsibility of a RigidBody component?

It determines the weight and behavior with other objects of an object

1. Why is the Pong game paddle mentioned as an example of an object whose rigidbody’s isKinematic property would be set to true? Hint: what controls the paddle movement?

In Pong, the paddle is swung to hit the ball instead of the ball bouncing towards a stationary paddle. As such isKinematic is set to true as the paddle needs to move to hit the ball

1. Explain what the rigidbody **constraints** are for. Give ONE example of when this might be useful.
2. Watch [this video](https://www.youtube.com/watch?v=ixM2W2tPn6c).
3. What is one major problem with using the Translate function to move an object?

The object does not have collision even if other objects have a rigidbody component

1. Why is the physics-based code put in the FixedUpdate function and not in Update?

FixedUpdate runs every physics state while Update runs every frame

1. What are the THREE different methods covered in the video to move an object using physics?

Explain how each method works.

-Addforce adds a force to an object moving it until friction and gravity slow it down. Objects using AddForce move slowly before accelerating and slide.

-Velocity runs like AddForce where .Velocity = direction\*speed which overrides all physics elements. The object is always either moving or not moving, there is no acceleration

-Move object is similar to Translate but it includes collision. The difference between .Translate and .MovePosition is that in MovePosition, Vector3 is added to transform.position

**QUESTION 2 (Colliders)**

1. What is the core responsibility of a collider component?

It is to define the physical body of an object.

1. Look at the question in [this forum post](https://forums.oculusvr.com/developer/discussion/59641/problems-with-fast-moving-object-collision-in-unity-table-tennis-racket-vs-ball). What is the solution to the problem? Do some research to explain why this is the solution. Draw a diagram to illustrate your answer.

-The solution was to change the ball’s collision detection mode.

It seems that Unity has 4 different modes of detection being, Discrete, Continuous Speculative, Continuous and Continuous Dynamic.

1. What is the difference between collision *detection* and collision *resolution*?

-Collision detection checks if two or more objects are overlapping/intersecting while collision resolution handles how the object is separated or modified to avoid/resolve the collision

**QUESTION 3 (Joints)**

1. What type of joint is used in the video? According to Unity’s documentation, what does this joint do? Give an example of how the joint could be used in a game.

-The joint used was a hinge joint.

-A way the joint can be used in a game to emulate a hinge like in a door.

1. What other types of joints are there? Briefly explain what FOUR of these joints does (do NOT include the type of joint you answered for part a above), and give an example of how each one might be used in a game.

- Character Joint which emulates a ball and socket joint

- Configurable Joint which emulates any skeletal joint which can be used to restrict the movement of the rigid body

- Fixed Joint restricts the movement of a rigid body to follow the movement of another that it is attached to and could be used in the case of detaching a part from a plane on input.

-Spring Joint keeps rigid bodies apart from each other but allows the distance between them to stretch slightly.

**QUESTION 4 (Physic Material)**

1. What is the difference between dynamic friction and static friction?

-Dynamic friction applies when the collider is moving while Static friction applies while it is stationary

1. Find a game that you have seen online or played yourself where physics is a core game mechanic.

-Getting Over It with Bennett Foddy is a game where you control a man in a pot and swing a hammer to move to move up the mountain.

-Physics are used to propel the player up surfaces by pull on surfaces with a Yosemite hammer, launch up by push against the ground and falling & slipping down the mountain.

You must:

* give a brief description of the game
* briefly explain how physics is used
* explain why physics is important
* include a relevant screenshot of the game

**QUESTION 5 (CharacterController)**

1. According to the video, the built-in CharacterController component does not interact with physics. So, a CharacterController does not react to gravity, and when a CharacterController collides with another object with a rigidbody it does not have any force added to it (like when being hit by a cannonball throws an object across the scene).

What gameplay-related reason(s) can you think of to explain why the CharacterController component was designed this way?

1. To make a character controlled by a CharacterController experience gravity while moving, two options are given in the video. What are these?
2. Create a Unity scene with a capsule game object with a CharacterController. The script below will make the character jump when the space bar is pressed.

Fill in the missing parts of the code, then copy the whole Character class code here. Make sure it is formatted properly!

Text

Description automatically generated

**QUESTION 6 (Constant Force component)**

1. For the Constant Force component, what is the difference between the Force and Relative Force properties, and between the Torque and Relative Torque properties?

Force moves the object along the global axis while relative force moves along the object’s local axis.

It is the same as the difference between Force and Relative Force for Torque.

1. When the cube is in the air with a Force X of 1, it moves, but when it falls to the plane, it doesn’t move.

Why not?

The force of friction is to strong for a force of 1 to over come.

1. An object has a constant force applied along its local positive Z axis. What happens to the object’s speed over time?

Explain your answer.

When it lands, the object slows down before accelerating and falling off the edge of the plane.

**QUESTION 7 (Forces)**

1. Paste your **Player** class code here. Make sure your code is readable, properly formatted, and commented.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class Character : MonoBehaviour

{

CharacterController characterController;

public float speed = 6.0f;

public float jumpSpeed = 8.0f;

public float gravity = 20.0f;

private Vector3 moveDirection = Vector3.zero;

public float Power = 10000.0f;

public float Radius = 100.0f;

public Vector3 vec = new Vector3(2,2,0);

Rigidbody rb;

// Start is called before the first frame update

void Start()

{

characterController = GetComponent<CharacterController>();

rb = GetComponent<Rigidbody>();

}

void CheckExplode()

{

if (Input.GetKeyDown(KeyCode.E))

{

Collider[] colliders = Physics.OverlapSphere(transform.position, Radius);

foreach (Collider hit in colliders)

{

Rigidbody rb = hit.GetComponent<Rigidbody>();

if (rb != null)

rb.AddExplosionForce(Power, transform.position, Radius);

}

}

}

// Update is called once per frame

void Update()

{

if (characterController.isGrounded)

{

moveDirection = new Vector3(Input.GetAxis("Horizontal"), 0f, Input.GetAxis("Vertical"));

moveDirection \*= speed;

if (Input.GetKeyDown(KeyCode.Space))

{

moveDirection.y = jumpSpeed;

}

}

//Gravity

moveDirection.y -= gravity\*Time.deltaTime;

characterController.Move(moveDirection\*Time.deltaTime);

CheckExplode();

}

}

1. Paste your **Kick** function code here. Make sure your code is readable, properly formatted, and commented.

void Kick()

{

if (Input.GetKeyDown(KeyCode.K))

{

Debug.Log("Kick");

Collider[] colliders = Physics.OverlapSphere(transform.position, Radius);

foreach (Collider hit in colliders)

{

Rigidbody rb = hit.GetComponent<Rigidbody>();

if (rb != null)

rb.AddForce(new Vector3(0,2,2)\*Power);

}

}}

1. What does the upwardsModifier argument of AddExplosiveForce do?



It moves where the centre of the explosion is located.

**QUESTION 8 (Forces cont.)**

1. What are the FOUR different values for ForceMode? Explain what each value does.

Impulse adds an instant force to the rigidbody using its mass

Acceleration adds a continuous acceleration to the rigidbody without its mass

Force adds a continuous force on the rigidbody

VelocityChange adds an instant velocity change ignoring its mass.

1. Paste your code for firing the sphere here.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class ImpluseQn8 : MonoBehaviour

{

public float Force = 10f;

private Vector3 Pulse;

Rigidbody rb;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody>();

}

// Update is called once per frame

void Update()

{

if (Input.GetKeyDown(KeyCode.UpArrow))

{

Pulse = new Vector3(0, 0, Force);

rb.AddForce(Pulse, ForceMode.Impulse);

}

if (Input.GetKeyDown(KeyCode.DownArrow))

{

Pulse = new Vector3(0, 0, -Force);

rb.AddForce(Pulse, ForceMode.Impulse);

}

if (Input.GetKeyDown(KeyCode.LeftArrow))

{

Pulse = new Vector3(-Force, 0, 0);

rb.AddForce(Pulse, ForceMode.Impulse);

}

if (Input.GetKeyDown(KeyCode.RightArrow))

{

Pulse = new Vector3(Force, 0, 0);

rb.AddForce(Pulse, ForceMode.Impulse);

}

if (Input.GetKeyDown(KeyCode.Space))

{

Pulse = new Vector3(0, Force, 0);

rb.AddForce(Pulse, ForceMode.Impulse);

}

}

}

Paste your code for rotating the beam here.

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class BeamSpin : MonoBehaviour

{

Rigidbody rb;

public float force = 10f;

private Vector3 Pulse;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody>();

}

// Update is called once per frame

void Update()

{

if (Input.GetKeyDown(KeyCode.Z))

{

Pulse = new Vector3(0,force,0);

rb.AddTorque(Pulse, ForceMode.Impulse);

}

if (Input.GetKeyDown(KeyCode.X))

{

Pulse = new Vector3(0, -force, 0);

rb.AddTorque(Pulse, ForceMode.Impulse);

}

}

}

**QUESTION 9 (Collision event-handling)**

1. Paste your code for **CosmicCube.cs** and **Orb.cs** here. Make sure your code is readable, properly formatted, and commented.

Cube code

using System.Collections;

using System.Collections.Generic;

using Unity.VisualScripting;

using UnityEditor.AssetImporters;

using UnityEngine;

public class CheckCollision : MonoBehaviour

{

Rigidbody rb;

MeshRenderer mat;

public Color color;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody>();

mat = GetComponent<MeshRenderer>();

}

public void OnCollisionEnter(Collision Sphere)

{

if (Sphere.gameObject.CompareTag("Sphere"))

{

mat.material.color = color;

Debug.Log("Change");

}

}

public void OnCollisionExit(Collision collision)

{

if (rb != null)

{

mat.material.color = Color.red;

}

}

}

Orb Code

using System.Collections;

using System.Collections.Generic;

using UnityEngine;

public class OrbLaunch : MonoBehaviour

{

public float Force = 10f;

private Vector3 Pulse;

Rigidbody rb;

// Start is called before the first frame update

void Start()

{

rb = GetComponent<Rigidbody>();

}

// Update is called once per frame

void Update()

{

if (Input.GetKeyDown(KeyCode.Space))

{

Pulse = new Vector3(0, 0 ,Force);

rb.AddForce(Pulse, ForceMode.Impulse);

}

}

}

**QUESTION 10 (Raycasting)**

1. Paste your completed code for the CheckLineOfSight() function here.

void CheckLineOfSight()

{

GameObject[] enemies = GameObject.FindGameObjectsWithTag("Enemy");

RaycastHit hit;

foreach (GameObject enemy in enemies)

{

Vector3 vec = enemy.transform.position - transform.position;

Debug.DrawRay(transform.position, transform.TransformDirection(vec));

if (Physics.Raycast(transform.position, vec, out hit, 30f))

{

if (hit.collider.gameObject.CompareTag("Enemy"))

{

enemy.GetComponent<MeshRenderer>().material.color = Color.green;

}

else{

enemy.GetComponent<MeshRenderer>().material.color= Color.red;

}

}

}

}

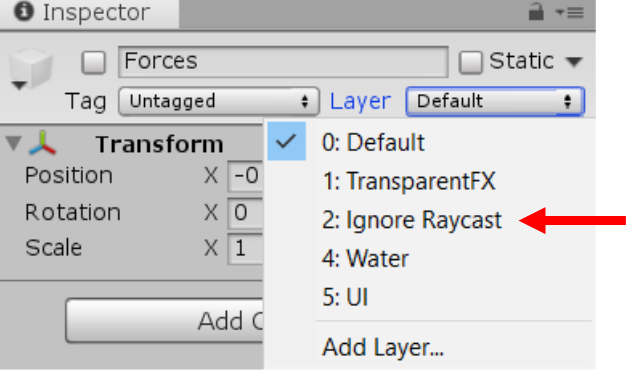
1. When checking for a hit using a ray, the code below is used:

if (Physics.Raycast(transform.position, vec, out hit, vec.magnitude))

Why is **vec.magnitude** used for the length of the ray?

It is to ensure the length of the ray is based on the magnitude of the vector which is the distance between the player and enemy object

1. The layers dropdown in the Inspector has an entry for **Ignore Raycast**.



What does this mean? Why do you think this is important? Give an example.

It means that the object does not get hit by raycast. It allows only certain things to get hit by the raycast.