Game Mechanics Exercises

Step 1: Project Setup
1. Open Unity and create a new project.
2. Choose a project name and select the 2D project template.
3. Once the project is created, switch to the Scene view.

Step 2: Scene Design
1. Click on the "GameObject" button in the bottom-left corner and select "Create Empty".
2. Right-click on the newly created empty object in the Hierarchy panel, select "Rename," and name it "Player".
3. Select the "Player" object in the Hierarchy panel and click on the "Add Component" button in the Inspector panel.
4. Search for and add the "Rigidbody2D" component. This will enable physical interactions for the player.
5. Select the "Player" object in the Hierarchy panel and design a game character in the Scene view by adjusting its position and size.
6. Right-click in the Hierarchy panel, select "Create Empty," and name it "Ground."
7. Select the "Ground" object, click on the "Add Component" button in the Inspector panel.
8. Add the "Box Collider 2D" component. This will create a ground that the player can collide with.
9. Design the ground by resizing and positioning the "Ground" object in the Scene view.
10. Select the "Player" object in the Hierarchy panel and click on the "Add Component" button in the Inspector panel and add the “Sprite Renderer”. Choose your Player sprite.
11.Select the "Player" object in the Hierarchy panel and click on the "Tag" button in the Inspector panel and choose the “Player”.
12. Select the "Ground" object in the Hierarchy panel and click on the "Add Component" button in the Inspector panel and add the “Sprite Renderer”. Choose your Ground sprite.
13.Select the "Ground" object in the Hierarchy panel and click on the "Tag" button in the Inspector panel and choose the “Obstacle”.

Step 3: Player Controls

1. Select the "Player" object in the Hierarchy panel.
2. Click on the "Add Component" button in the Inspector panel and select "New Script."
3. Name the script "PlayerController" and open it.
4. Paste the following sample code into the "PlayerController" script:

PlayerController.cs | PlayerJump.cs

selection

```
using UnityEngine;

public class PlayerController : MonoBehaviour
{
    public float speed = 5f;
    private Rigidbody2D rb;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    private void FixedUpdate()
    {
        float moveInput = Input.GetAxis("Horizontal");
        rb.velocity = new Vector2(moveInput * speed, rb.velocity.y);
    }
}
```

5. The code contains a simple control mechanism that allows the player to move horizontally.

Step 4: Jumping Mechanism

1. Select the "Player" object in the Hierarchy panel.
2. Click on the "Add Component" button in the Inspector panel and select "New Script."
3. Name the script "PlayerJump" and open it.
4. Paste the following sample code into the "PlayerJump" script:

PlayerController.cs | PlayerJump.cs

selection

```
using UnityEngine;

public class PlayerJump : MonoBehaviour
{
    public float jumpForce = 5f;
    private bool isGrounded;
    private Rigidbody2D rb;

    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }

    private void Update()
    {
        if (Input.GetButtonDown("Jump") && isGrounded)
        {
            rb.AddForce(Vector2.up * jumpForce, ForceMode2D.Impulse);
            isGrounded = false;
        }
    }

    private void OnCollisionEnter2D(Collision2D collision)
    {
        if (collision.gameObject.CompareTag("Obstacle"))
        {
            isGrounded = true;
        }
    }
}
```

5. The code allows the player to jump and re-jump when colliding with the ground.

By following these steps, you can have a simple 2D game. The provided code examples allow the player to move horizontally and jump. You can further develop and customize your project using these codes. You can expand your project by adding more complex mechanics, enemies, obstacles, or goals. Good luck with your project!