Unity Workshop Walkthrough

Setup

Pre-set up values

- Show the resolution settings for standalone
- Show orientation settings for iPhone

Scene

- Make a new folder with a new scene

Camera

- Set to isometric
- Position Camera Z = -2
- How to set viewport to isometric
- Set Game View window to standalone

Controllers

- Make a Controllers GameObject
- RESET TRANSFORM

Input Controller

Script

```
public class InputController : MonoBehaviour
{
    public BirdController birdController;
    public PipeSpawner pipeSpawner;

    // Set the flap key as the space key
    private KeyCode flapKey = KeyCode.Space;

    private bool hasStartedFlapping = false;

    // This function will change behaviour depending on platform
    private bool IsFlapKeyPressed() {
#if UNITY_IOS || UNITY_ANDROID
        return GetTouchInput();
#else
        if (UnityEditor.EditorApplication.isRemoteConnected) {
            // For remote debugging in mobile
            return GetTouchInput();
        }
        else {
            return GetDesktopInput();
        }
#endif
    }

    // For touch inputs
    // Mobile and debug Unity Remote
    private bool GetTouchInput() {
        return Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Began;
    }

    // For regular keycap input
    private bool GetDesktopInput() {
        return Input.GetKeyDown(flapKey);
    }

    private void Update() {
        if (IsFlapKeyPressed()) {

            // If this is the first flap, set the flag to true
            if (!hasStartedFlapping) {
                hasStartedFlapping = true;
                // TODO: Start the pipes
            }

            Debug.Log("Flap!");
            // TODO: Flap the bird
        }
    }
```

Add InputController to Controllers GameObject

Bird

Setup

- Make a new Bird GameObject
- RESET TRANSFORM
- Add
  - Circle collider 2D
  - Rigidbody 2D
    - Freeze x position
    - Mass 1.25 units

Animation

- Make a new animation in the project view
- Drag and drop the animation onto a Bird GameObject
- DOCK THE ANIMATION WINDOW
- Drag and drop the three keyframes into window
- Go to settings show sample rate
- Set sample rate to 12
- Set to LOOP!!

Script

```
public class BirdController : MonoBehaviour
{
    public float flapForce = 5.0f;
    public float rotationTipScale = 2.0f;

    private Rigidbody2D birdRigidbody;
    private Vector2 flapDirection;

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

        // This starts the physics as locked
        birdRigidbody.isKinematic = true;

        flapDirection = Vector2.up * flapForce;
    }
    public void Flap() {
        // Unlock the flapping if this is the first press
        if (birdRigidbody.isKinematic) {
            birdRigidbody.isKinematic = false;
        }

        // Stop moving down first
        birdRigidbody.velocity = Vector2.zero;

        // and FLAP!
        birdRigidbody.AddForce(flapDirection);
    }

    private void OnTriggerEnter2D(Collider2D collision) {
        // TODO: Detect collisions with score zone
    }

    private void OnCollisionEnter2D(Collision2D collision) {
        // TODO: Detect collisions with pipe
    }

    private void FixedUpdate() {
        // The subtle tip up/down effect for the bird
        birdRigidbody.rotation = birdRigidbody.velocity.y * rotationTipScale;
    }
}
```

Put this on the Bird GameObject

Flap force 200

Rotation Tip scale 5

TODOs

- Add the BirdController flapping to InputController
- Add `public BirdController birdController;`
- `birdController.Flap();`
- DEMO CHECKPOINT!

Pipe

Prefab

- Talk about logic and art split up
- Show collider and tag set up for prefabs
    - Trigger vs non trigger

Boundary

- Create empty Boundary GameObject
- RESET TRANSFORM
- Add Box Collider 2D
- Set IsTrigger
- Tag with Boundary

Script

```
public class PipeMover : MonoBehaviour
{
    public Transform pipeTransform;
    public float moveSpeed = 5.0f;

    private void Update() {
        // Move the pipe every frame that it is alive
        pipeTransform.position += Vector3.right * -moveSpeed * Time.deltaTime;
    }

    private void OnTriggerExit2D(Collider2D collision) {
        // When we exit the play area, destroy the game object to save performance
        if (collision.gameObject.CompareTag("Boundary")) {
            Destroy(pipeTransform.gameObject);
        }
    }
}
```

Put this on Pipe Prefab Logic component

Set pipe Transform to be top parent

Set move speed to 3

TODOs

- Populate the collision logic in BirdController

```
    private void OnTriggerEnter2D(Collider2D collision) {
        // If we collide with the score area, increment the score
        if (collision.gameObject.CompareTag("Score")) {
            Debug.Log("Score incrementing");

            // TODO: Increase Score
        }
    }

    private void OnCollisionEnter2D(Collision2D collision) {
        // If we collide with a pipe, stop the game
         if (collision.collider.gameObject.CompareTag("Pipe")) {
            Debug.Log("Hit pipe");

            // TODO: Stop the game
        }
    }
```

Pipe Spawner

Setup

- Make an empty Pipe Spawner GameObject
- RESET TRANSFORM
- Push it off somewhere to the right
- Place it about where you want pipes to come from

Script

```
public class PipeSpawner : MonoBehaviour
{
    public GameObject pipePrefab;
    public float timeBetweenPipes = 2.0f;
    public float spawnRangeUnits = 2.0f;

    private void SpawnPipe() {
        // Pick a random spot
        float randomYOffset = Random.Range(-spawnRangeUnits, spawnRangeUnits);

        // Make a new pipe
        Instantiate(pipePrefab, transform.position + new Vector3(0, randomYOffset, 0),
Quaternion.identity);
    }

    private IEnumerator SpawnPipes() {
        // Runs as long as the time in game is running
        while(true) {
            SpawnPipe();

            // Note that WaitForSeconds is affected by Time.Timescale
            yield return new WaitForSeconds(timeBetweenPipes);
        }
    }

    public void StartSpawningPipes() {
        // External interface to start spawning pipes
        StartCoroutine("SpawnPipes");
    }

}
```

Add this to PipeSpawner GameObject

TODOs

- Update InputController for first time flap to spawn pipes
- Add `public PipeSpawner pipeSpawner;`
- `pipeSpawner.StartSpawningPipes();`
- DEMO CHECKPOINT!

GameController

UI Setup

- Two Canvases
    - Background Canvas
    - UI Canvas
- Adjust Render Mode to Screen Space Camera
    - Background Canvas
        - Plane distance = 5
    - Restart Canvas
        - Plane distance = 0.5

UI Elements

- Add a Background
    - Child of BG Canvas
    - Preserve Aspect
    - Scale up to fill the entire screen
- Add a Score text
    - Child of UI Canvas
    - Change anchor to top middle
- Add a Restart UI GameObject
    - Child of Restart Canvas
    - Add a button as Child
        - Adjust size in Rect Transform
        - Adjust the text to Restart
        - Change font if desired
    - Add Final Score Text as Child
        - Adjust content and font

Script

```
public class GameController : MonoBehaviour
{
    // Singleton access
    public static GameController instance;
    public Text scoreText;
    public GameObject restartUI;
    public Text finalScoreText;

    private int score;

    public void IncreaseScore() {
        score++;
        scoreText.text = score.ToString();
    }

    public void StartPlaying() {
        // Start the game if it was paused
        Time.timeScale = 1.0f;

        Debug.Log("Game started");
    }

    public void StopPlaying() {
        // Stop the game
        Time.timeScale = 0.0f;

        // Turn on the restart UI
        restartUI.SetActive(true);

        finalScoreText.text = "Final Score: " + score.ToString();

        Debug.Log("Game stopped");
    }

    public void RestartGame() {
        // Get this current scene
        Scene thisScene = SceneManager.GetActiveScene();

        // Restart it
        // Do this with build index incase scene names are the same
        SceneManager.LoadScene(thisScene.buildIndex);
    }

    private void Awake() {
        // Singleton enforcement
        if (instance == null) {
            instance = this;
        }
        else {
            Debug.LogError("More than once instance of the Game Controller Singleton.
Deleting the old instance.");
            DestroyImmediate(instance);

            instance = this;
        }
```

```
        // Turn off the restart UI by default
        restartUI.SetActive(false);

        // Start the game
        StartPlaying();
    }
}
```

TODOs

- Add GameController to Controllers GameObject
  - Set up
    - Score Text
    - Restart UI
    - Final Score Text
- Hook up button to RestartGame function
- Adjust BirdController with final restart functions

```
    private void OnTriggerEnter2D(Collider2D collision) {
        // If we collide with the score area, increment the score
        if (collision.gameObject.CompareTag("Score")) {
            Debug.Log("Score incrementing");

            GameController.instance.IncreaseScore();
        }
    }

    private void OnCollisionEnter2D(Collision2D collision) {
        // If we collide with a pipe, stop the game
         if (collision.collider.gameObject.CompareTag("Pipe")) {
            Debug.Log("Hit pipe");

            GameController.instance.StopPlaying();
        }
    }
```