Place the Portal
Technical Design Document

By **Ajmal Khan**

Common Practices in the Project

- Awake is used to get the components of the object or to initialize variables. In case of singletons, it is used to return the current instance if it is already there or to destroy any new instances.
- OnEnable is used to register the object to a list of objects that inherit from some interface.
- OnDisable is used to unregister the object from the list of objects that inherit from some interface.
- Update is used to handle input.
- FixedUpdate is used to handle Physics calculations.

Design Patterns followed

- Those objects that will always have only one instance in a scene are made singleton classes. These include:
    - UIManager
    - PlayButtonController
- Based on the observer pattern, the above classes are subjects for subscriber objects and they use Register and Unregister methods to make the subscriber objects subscribe to the subject.
    - UIManager uses IClickableUI interface to get following subscribers:
        - PortalIconController
        - PlayButtonController
    - PlayButtonController uses IResetable interface to get following subscribers:
        - BallController
        - CameraController
        - Star
        - Box
        - Laser
        - LaserButton

Flow of the Game

Order of execution of scripts:

1. UIManager
2. PlayButtonController
3. All the rest.

Interfaces

IResetable

- GetOriginalState
    - To get the original state of the object. Depending on the object, the original state can be position, velocity, rotation or a combination of these.
- SetOriginalState
    - To set the original state of the object to whatever we got in the get method.

ICollectible

- OnCollect
    - To set the collectible object invisible as it has been collected. This too can differ with each collectible as to what that object does.

IClickableUI

- OnClick
    - To handle the click on the UI buttons of the canvas.

Constant Classes

NameConstants

- Ball

TagConstants

- Portable
- Placeable
- PortalIcon
- PortalPlaceholder

- Goal
- Collectible
- PlayButton
- Button
- Laser

Scripts

Methods in scripts have following color coding:

- Inbuilt methods whose uses are defined at the beginning of this document - Blue
- Interface methods whose function is also clear from the Interface section - Orange
- Methods that require some explanation - Green

BallController

Serialized Variables:

- Float terminalVelocity

Methods:

- Awake
- OnEnable
- OnDisable
- GetOriginalState
- SetOriginalState
- FixedUpdate
    - To set the terminal velocity of the ball. Done to fix a bug related to speed of the ball causing issue with collision detection.
- OnTriggerEnter2D
    - On collision with goal, end the current level with success.
    - On collision with a star, calculate the no. of stars collected by the ball.

CameraController

Serialized Variables:

- Float minX
- Float maxX
- Float Speed
- Button playButton
- GameObject ball

Methods:

- OnEnable
- OnDisable
- GetOriginalState

- SetOriginalState
- Update
    - Calls MoveCamera.
- MoveCamera
    - If PlayButtonController.Instance.isPlay is true, it calls MoveCamOnDrag. Else, it calls MoveCamWithBall.
- MoveCamOnDrag
    - If the mouse button is clicked, check if the click is not on any collider.
        - If it is, set IsDragging as true and get the origin mouse click position.
        - Else, check if the PortalPlaceholer is dragging using playerController.isDragging. If true, set the shouldMoveWithPortal as true and get the origin mouse click position.
    - If the mouse button is released, set isDragging, shouldMoveWithPortal and camDirection as false.
    - If isDragging is true
        - Set camera X position as:
            - Current camera X position + original mouse X position - current mouse X position.
            - Clamp the X value between the max and min bounds of the level.
        - Use Vector3.Lerp to interpolate between the target cam position and current cam position by a predefined speed.
    - Else if shouldMoveWithPortal is true
        - Calculate camera bounds positions.
        - Set camDirection to left or right based on if the current mouse position is near the left boundary or right boundary. Otherwise, the camDirection is zero.
        - If camDirection is not zero, set camera position as
            - Current cam position + camDirection * some predefined magnitude.
        - Clamp the X value of this between the min and max bounds of the level.
        - Use Vector3.Lerp to interpolate between the target cam position and current cam position by a predefined speed.
- MoveCamWithBall
    - New cam X position will be ball X position and clamp between the bounds of the level.
- CalculateCameraBounds
    - Get the left and right edges X position of the camera.

Star

Methods:

- Awake
- OnEnable

- OnDisable
- GetOriginalState
- SetOriginalState
- OnCollect

Box

Methods:

- Awake
- OnEnable
- OnDisable
- GetOriginalState
- SetOriginalState

Laser

Serialized Variables:

- Float laserDistance
- Transform laserStartPoint
- Bool isLaserOn

Methods:

- Awake
- OnEnable
- OnDisable
- GetOriginalState
- SetOriginalState
- Update
    - Call ShootLaser if isLaserOn is true.
- OnTriggerEnter2D
    - Stop level if the ball collides with the laser.
- ShootLaser
    - Cast a ray towards the right of the laser gun.
    - If it hits a collider, call Draw2DRay with start point and hit point.
    - If not, call Draw2DRay with the start point and predefined laserDistance towards the right of the laser gun..
- Draw2DRay
    - Draw a line between start and end positions using lineRenderer.
    - Set the collider of the laser based on the start and end positions.
- SwitchLaser
    - Set the laser line, laser collider and isLaserOn to the input bool value.

LaserButton

Serialized Variables:

- Float buttonDelay
- Float moveSpeed
- GameObject laserGun

Methods:

- Awake
- OnEnable
- OnDisable
- GetOriginalState
- SetOriginalState
- OnCollisionEnter2D
    - If isbuttonPressed is false and the colliding object has the tag Portable, call PressButton.
- OnCollisionExit2D
    - If isbuttonPressed is true and the colliding object has the tag Portable, check if time since last collision is more than buttonDelay.
        - StartCoroutine(ButtonReleaseCoroutine()).
- PressButton
    - Set current collision time.
    - Get time since last collision as the difference between current and last collision times.
    - Set last collision time as current collision time.
    - StartCoroutine(ButtonPressCoroutine());
    - Call SwitchLaser of Laser class with opposite of isLaserOn.
- ButtonPressCoroutine
    - If isButtonPressed true and while current button position is not equal to buttonDownPosition, set current position to move towards buttonDownPosition with moveSpeed.
- ButtonReleaseCoroutine
    - Wait for buttonDelay seconds.
    - While the current button position is not equal to buttonUpPosition, set the current position to move towards buttonUpPosition with moveSpeed.
    - Set isButtonPressed to false.

PlayerController

Serialized Variables:

- LayerMask targetLayer
- Float distanceToSnap
- GameObject portal

Methods:

- OnMouseDown
    - Set the local scale as 1.2f times current scale to get the feeling of clicking the object.

    - Set isDragging as true.
- OnMouseDrag
    - Get the new position of the object at each frame it is being dragged using Camera.main.ScreenToWorldPoint(Input.mousePosition).
    - If isDragging is true and current position is not equal to new position, set current position as new position.
- OnMouseUp
    - Call GetPortalPlacePostition.
    - Set the local scale back to normal to get the feeling of releasing the object.
    - Set isDragging as false.
    - Call PlacePortal.
- GetPortalPlacePostition
    - Loop at angle in increments of 45 degrees i.e. in 8 directions.
        - Convert the degrees angle to direction vector.
        - Cast a ray from the current position for each direction vector up to Mathf.Infinity and set the colliding layer mask as the Placeable walls.
        - If a collider is hit, calculate distance between current position and hit position.
        - Get the nearestObject, nearestDistance, nearestDirection and nearestHitPoint from the 8 directions.
    - If nearestObject is not null and nearestDistance is greater than the predefined distanceToSnap, pass the object, direction and hitpoint to global variables placePortalAt, portalDirection and portalPosition.
    - Else, set placePortalAt as null.
- PlacePortal
    - If placePortalAt is null, set isPortalPlaced as false and return.
    - Call UnplacePortal to get a bool value unplacePortal. If it is true, set isPortalPlaced as false and return.
    - Calculate the rotation angle in degrees based on portalDirection. It will be tan(y/x) - 90 degrees. Set a Quaternion value with this angle as the Z rotation using Quaternion.Euler.
    - Call GetPortalEnds with angle.
    - Call CheckEnd for both ends of the portal.
    - If both ends are on the edge of a placeable object, set the rotation to the Quaternion calculated above, position to portalPosition and set isPortalPlaced as true.
    - If exactly one end is on the edge of a placeable object, determine which end is on the boundary and which is not. Get the direction and magnitude to move the portal in the direction of the end that is on boundary. Set the rotation to the Quaternion calculated above, position to the new position calculated just above and set isPortalPlaced as true.
    - If no ends are on edge, set rotation as Quaternion.identity and isPortalPlaced as false.
- CheckEnd

        - Cast a ray in the direction opposite to the portal direction from the ends of the portal.
        - If the ray hits a point that is very close to the portalEnd, then it is on the edge of the placeable object.
        - Return true if the end is on edge.
        - Else return false.
- GetDirectionToMovePortal
    - Calculate the direction from center to the end that is on the boundary.
- GetMagnitudeToMovePortal
    - Return magnitude calculated as half of the X scale of the portal. This is an approximate value.
- GetPortalEnds
    - Return endpoints of the portal in portalEndA and portalEndB.
    - Endpoints are (x +- rcosQ, y +- rsinQ) and then move a little in the direction of the portal so that the ray hits the collider.
- UnplacePortal
    - If portalPosition is the current position, return true.
    - Else return false.

PortalController

Serialized Variables:

- Transform destination

Methods:

- Awake
- OnTriggerEnter2D
    - If the destination is null, return.
    - If colliding object has tag portable and distance between portal position and colliding object position is greater than 0.3f:
        - Get Portal Entry Direction.
        - Get Portal Exit Direction.
        - Get Ball Entry Direction.
        - Set Ball Exit Direction as sum of above 3 vectors.
        - Change Ball Velocity direction to Ball Exit Direction.
        - Change Ball position to Destination position + an offset.

PlayButtonController

Serialized Variables:

- GameObject portal

Methods:

- Awake
- OnEnable
- OnDisable

- RegisterResetableObject
- UnregisterResetableObject
- OnClick
    - If isPlay is true, call StartGame.
    - Else call ResetGame.
- Start
    - Set Time.timeScale to zero to freeze time at the start of level.
    - Call GetOriginalPosition.
- GetOriginalPosition
    - For each object in the list of resettable objects, call object.GetOriginalState.
- SetOriginalPosition
    - For each object in the list of resettable objects, call object.SetOriginalState.
- StartGame
    - Set Time.timeScale to 1f to resume time.
    - Set isPlay to false.
    - Call ActivatePortals.
- ResetGame
    - Set isPlay to true.
    - Call SetOriginalPosition.
    - Call DeactivatePortals.
    - Set Time.timeScale to zero to freeze time.
- ActivatePortals
    - Find all objects with the tag PortalPlaceholder in an array..
    - Set noOfPortals to no. of placeholders found i.e. length of array.
    - If noOfPortals > 0, for each portalPlaceholder, get playerController, portalController and the portal.
    - Instantiate the portal in place of the placeholder and set the placeholders as inactive.
    - Loop at all portals:
        - If both portals are placed, then add destinations.
        - If only one portal is placed, then remove the destination.
- DeactivatePortals
    - If portal placeholders are available, set placeholders as active.
    - If portals are available, destroy portals.

PortalIconController

Serialized Variables:

- GameObject portalX
- GameObject portalY
- Vector3 portalXPosition
- Vector3 portalYPosition

Methods:

- OnEnable

- OnDisable
- OnClick
    - Set portal icon as inactive.
    - Instantiate two portal placeholders on the screen.

UIManager

Serialized Variables:

- List<Button> buttons

Methods:

- Awake
- RegisterClickableObject
- UnregisterClickableObject
- Start
    - Call AddListenerToObjects.
- AddListenerToObjects
    - For each button in list of buttons, button.onClick.AddListener(() => OnButtonClick(button));
- OnButtonClick
    - For each object in the list of clickableUI objects, call the OnClick method for those objects.