# AudioManagerScript Class Reference

### Public Member Functions

void **PlaySound** (AudioClip clip)

### Static Public Attributes

static **AudioManagerScript** **current**

### Private Member Functions

void **Awake** ()

### Private Attributes

AudioSource **audioSource**

### Description

**AudioManagerScript** class controls audio like the sound generated on diamond collection

### Member Function Documentation

#### void AudioManagerScript.PlaySound (AudioClip *clip*)

Plays the audio clip

## CameraScript Class Reference

### Private Member Functions

void **Start** ()

void **Update** ()

### Private Attributes

Transform **player**

float **maxAngle** = 7f

Vector3 **offsetPosition**

### Description

**CameraScript** class controls the camera in the game which keeps following the player and makes sure that the player occupies constant screen space.

## CardinalDirection Class Reference

### Static Public Attributes

static readonly Vector2 **Up** = new Vector2 (0, 1)

static readonly Vector2 **Down** = new Vector2 (0, -1)

static readonly Vector2 **Right** = new Vector2 (1, 0)

static readonly Vector2 **Left** = new Vector2 (-1, 0)

static readonly Vector2 **UpRight** = new Vector2 (1, 1)

static readonly Vector2 **UpLeft** = new Vector2 (-1, 1)

static readonly Vector2 **DownRight** = new Vector2 (1, -1)

static readonly Vector2 **DownLeft** = new Vector2 (-1, -1)

### Description

The **CardinalDirection** class controls the gestures and the corresponding change in direction.

## Corner1PoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 5

bool **willGrow** = true

### Static Public Attributes

static **Corner1PoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools corner type 1

## Corner2PoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 5

bool **willGrow** = true

### Static Public Attributes

static **Corner2PoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools corner type 2

## Corner3PoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 5

bool **willGrow** = true

### Static Public Attributes

static **Corner3PoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools corner type 3

## Corner4PoolerScript Class Reference

## 

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 5

bool **willGrow** = true

### Static Public Attributes

static **Corner4PoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools corner type 4

## GroundEmptyVerticalPoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 1

bool **willGrow** = true

### Static Public Attributes

static **GroundEmptyVerticalPoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools the empty vertical ground platforms

## GroundVerticalObstacle1PoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 1

bool **willGrow** = true

### Static Public Attributes

static **GroundVerticalObstacle1PoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools the vertical obstacle type 1

## GroundVerticalObstacle2PoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 1

bool **willGrow** = true

### Static Public Attributes

static **GroundVerticalObstacle2PoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools the vertical obstacle type 2

## GroundVerticalPoolerScript Class Reference

### Public Member Functions

GameObject **GetPooledObject** ()

### Public Attributes

GameObject **pooledObject**

int **pooledAmount** = 10

bool **willGrow** = true

### Static Public Attributes

static **GroundVerticalPoolerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

### Private Attributes

List< GameObject > **pooledObjects**

### Description

Pools the vertical ground platforms

## PlatformDestroyScript Class Reference

### Private Member Functions

void **Destroy** ()

void **OnTriggerExit** (Collider other)

### Description

**PlatformDestroyScript** class destroys the platform which have already been used or crossed

## PlatformSpawnerScript Class Reference

### Public Member Functions

void **BeginToSpawn** ()

### Public Attributes

GameObject **platform**

bool **gameOver**

GameObject **diamond**

### Static Public Attributes

static **PlatformSpawnerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

void **Update** ()

void **CreateDiamonds** (Vector3 pos)

void **SpawnInitialVertical** ()

void **SpawnVertical** ()

Bounds **GetMaxBounds** (GameObject g)

void **SpawnCornersHorizontal** ()

void **SpawnHorizontalRight** ()

void **SpawnCornerLeft** ()

void **SpawnCornerRight** ()

void **SpawnHorizontalLeft** ()

void **SpawnEmptyHorizontalRight** ()

void **SpawnEmptyHorizontalLeft** ()

void **SpawnCornerUp** ()

void **SpawnObstacleVertical** (GameObject newObj)

void **CreateCombinations** ()

### Private Attributes

GameObject **corner1**

GameObject **corner2**

GameObject **corner3**

GameObject **corner4**

Vector3 **lastPos**

float **size**

float **sizeCorner**

int **direction**

int **counterUp**

int **counterHor**

float **timeForCreation** = 0.8f

### Description

The **PlatformSpawnerScript** handles the random spawning of platforms

### Member Function Documentation

#### void PlatformSpawnerScript.BeginToSpawn ()

Starts the spawning of platforms

#### void PlatformSpawnerScript.CreateCombinations ()[private]

Spawns obstacles randomly

#### void PlatformSpawnerScript.CreateDiamonds (Vector3 *pos*)[private]

Spawns diamonds on the path randomly

#### void PlatformSpawnerScript.SpawnCornerLeft ()[private]

Spawns left corners

#### void PlatformSpawnerScript.SpawnCornerRight ()[private]

Spawns right corners

#### void PlatformSpawnerScript.SpawnCornersHorizontal ()[private]

Spawns the horizontal corners

#### void PlatformSpawnerScript.SpawnEmptyHorizontalLeft ()[private]

Spawns empty horizontal left platforms randomly

#### void PlatformSpawnerScript.SpawnEmptyHorizontalRight ()[private]

Spawns empty horizontal platforms to the right randomly

#### void PlatformSpawnerScript.SpawnHorizontalLeft ()[private]

Spawns horizontal platforms to the left

#### void PlatformSpawnerScript.SpawnHorizontalRight ()[private]

Spawns horizontal platforms to the right

#### void PlatformSpawnerScript.SpawnInitialVertical ()[private]

Spawns initial vertical platform

#### void PlatformSpawnerScript.SpawnObstacleVertical (GameObject *newObj*)[private]

Spawns the vertical obstacles

#### void PlatformSpawnerScript.SpawnVertical ()[private]

Spawns vertical platforms

#### void PlatformSpawnerScript.Start ()[private]

Start is called before the first frame update

## PlayerScript Class Reference

### Public Member Functions

void **Replay** ()

### Public Attributes

Image **startImage**

AudioClip **diamondFx**

### Private Member Functions

void **Start** ()

void **Update** ()

void **Jump** ()

void **OnTriggerEnter** (Collider other)

void **Slide** ()

void **TurnLeft** ()

void **ExitSlide** ()

void **OnCollisionEnter** (Collision collision)

void **TurnRight** ()

### Private Attributes

bool **started**

*A boolean variable. Checks if the game is already started or not.*

bool **jumping**

Rigidbody **rb**

Animator **animator**

float **speed**

float **jump**

int **turn**

*accounts for player turns*

float **colHeight**

float **colRadius**

float **colCenterY**

float **colCenterZ**

### Description

The **PlayerScript** class. Includes all player control and gesture functionalities like jump, turn, sway, etc.

### Member Function Documentation

#### void PlayerScript.ExitSlide ()[private]

Exist slide animation

#### void PlayerScript.Jump ()[private]

Makes the player jump

#### void PlayerScript.OnTriggerEnter (Collider *other*)[private]

Trigger function for colliding with obstacle, fence or collecting diamonds

#### void PlayerScript.Replay ()

Restarts the game

#### void PlayerScript.Slide ()[private]

Makes the player slide

#### void PlayerScript.Start ()[private]

Start is called before the first frame update

#### void PlayerScript.TurnLeft ()[private]

Makes the player turn left

#### void PlayerScript.TurnRight ()[private]

Makes the player turn right

#### void PlayerScript.Update ()[private]

Update is called once per frame

## ScoreManagerScript Class Reference

### Public Member Functions

void **StartScore** ()

void **DiamondScore** ()

void **StopScore** ()

### Public Attributes

Text **scoreText**

Text **highScoreText**

GameObject **scoreTxtObj**

GameObject **panelObj**

### Static Public Attributes

static **ScoreManagerScript** **current**

### Private Member Functions

void **Awake** ()

void **Start** ()

void **Update** ()

void **IncrementScore** ()

### Private Attributes

int **score**

### Description

The **ScoreManagerScript** class manages the score updates in the game. The score is incremented by 1 in every 0.5 seconds and score is incremented by 10 if a diamond is collected.

## SwipeManager Class Reference

### Public Member Functions

delegate void **OnSwipeDetectedHandler** (Swipe swipeDirection, Vector2 swipeVelocity)

### Static Public Member Functions

static bool **IsSwiping** ()

static bool **IsSwipingRight** ()

static bool **IsSwipingLeft** ()

static bool **IsSwipingUp** ()

static bool **IsSwipingDown** ()

static bool **IsSwipingDownLeft** ()

static bool **IsSwipingDownRight** ()

static bool **IsSwipingUpLeft** ()

static bool **IsSwipingUpRight** ()

### Static Public Attributes

static Vector2 **swipeVelocity**

### Properties

static OnSwipeDetectedHandler **OnSwipeDetected**

### Private Member Functions

void **Awake** ()

void **Update** ()

### Static Private Member Functions

static void **DetectSwipe** ()

*Attempts to detect the current swipe direction. Should be called over multiple frames in an Update-like loop.*

static bool **GetTouchInput** ()

static bool **GetMouseInput** ()

static bool **IsDirection** (Vector2 direction, Vector2 cardinalDirection)

static Swipe **GetSwipeDirByTouch** (Vector2 currentSwipe)

static bool **IsSwipingDirection** (Swipe swipeDir)

### Private Attributes

float **minSwipeLength** = 0.5f

bool **triggerSwipeAtMinLength** = false

bool **useEightDirections** = false

### Static Private Attributes

const float **eightDirAngle** = 0.906f

const float **fourDirAngle** = 0.5f

const float **defaultDPI** = 72f

const float **dpcmFactor** = 2.54f

static Dictionary< Swipe, Vector2 > **cardinalDirections**

static OnSwipeDetectedHandler **\_OnSwipeDetected**

static float **dpcm**

static float **swipeStartTime**

static float **swipeEndTime**

static bool **autoDetectSwipes**

static bool **swipeEnded**

static Swipe **swipeDirection**

static Vector2 **firstPressPos**

static Vector2 **secondPressPos**

static **SwipeManager** **instance**

### Description

The **SwipeManager** class controls the gestures and their corresponding actions/results on the player.

### Member Function Documentation

#### static void SwipeManager.DetectSwipe ()[static], [private]

Attempts to detect the current swipe direction. Should be called over multiple frames in an Update-like loop.

### Member Data Documentation

#### Dictionary<Swipe, Vector2> SwipeManager.cardinalDirections[static], [private]

**Initial value:**= new Dictionary<Swipe, Vector2> ()

{

{ Swipe.Up, CardinalDirection.Up },

{ Swipe.Down, CardinalDirection.Down },

{ Swipe.Right, CardinalDirection.Right },

{ Swipe.Left, CardinalDirection.Left },

{ Swipe.UpRight, CardinalDirection.UpRight },

{ Swipe.UpLeft, CardinalDirection.UpLeft },

{ Swipe.DownRight, CardinalDirection.DownRight },

{ Swipe.DownLeft, CardinalDirection.DownLeft }

}