Multiplayer Stacker

1.0

Generated by Doxygen 1.8.10

Tue Oct 6 2015 15:16:41

Contents

1 Namespace Index **1**

1.1 Packages 1

2 Hierarchical Index **3**

2.1 Class Hierarchy 3

3 Class Index **5**

3.1 Class List 5

4 Namespace Documentation **9**

4.1 MultiStack Namespace Reference 9

4.1.1 Enumeration Type Documentation 11

4.1.1.1 GameOverType 11

4.1.1.2 Shape 11

4.1.2 Function Documentation 12

4.1.2.1 CallBack() 12

5 Class Documentation **13**

5.1 MultiStack.Background Struct Reference 13

5.1.1 Detailed Description 13

5.1.2 Member Data Documentation 13

5.1.2.1 backgroundPrefab 13

5.1.2.2 backgroundTilePrefab 13

5.2 MultiStack.BackgroundManager Class Reference 13

5.2.1 Detailed Description 14

5.2.2 Member Data Documentation 14

5.2.2.1 backgrounds 14

5.3 MultiStack.CameraManager Class Reference 14

5.3.1 Detailed Description 15

5.3.2 Member Function Documentation 15

5.3.2.1 CallBackOnTargetReached(CallBack callBack) 15

5.3.2.2 RequestNewTarget(float target) 15

5.3.2.3 ResetPanSpeedOnTargetReached() 15

5.3.2.4 ResetTarget(float panSpeedToTarget=1f) 15

5.3.2.5 SetTarget(float target, float panspeed=1f) 16

5.3.2.6 StartPanning(float panSpeed=1f) 16

5.3.3 Member Data Documentation 16

5.3.3.1 targetY 16

5.3.4 Property Documentation 16

5.3.4.1 hasReachedTarget 16

5.3.4.2 shouldPan 16

5.4 MultiStack.CameraShake Class Reference 16

5.4.1 Detailed Description 17

5.4.2 Member Function Documentation 17

5.4.2.1 Shake(float shakeIntensity=0.05f, float shakeDecay=0.004f) 17

5.5 MultiStack.CanOnlyPickupOnceModifier Class Reference 17

5.5.1 Detailed Description 18

5.5.2 Member Function Documentation 18

5.5.2.1 Activate() 18

5.5.2.2 Deactivate() 18

5.6 MultiStack.CatchTheShapeModifier Class Reference 18

5.6.1 Detailed Description 19

5.6.2 Member Function Documentation 19

5.6.2.1 Activate() 19

5.6.2.2 Deactivate() 19

5.7 MultiStack.ClickableObject Class Reference 19

5.7.1 Detailed Description 19

5.7.2 Member Function Documentation 20

5.7.2.1 Clicked() 20

5.7.3 Property Documentation 20

5.7.3.1 clickable 20

5.8 MultiStack.ClickHandler Class Reference 20

5.8.1 Detailed Description 21

5.8.2 Member Data Documentation 21

5.8.2.1 boxMask 21

5.8.2.2 dragLine 21

5.8.2.3 isMainMenuScene 21

5.8.2.4 playerColours 21

5.8.3 Property Documentation 21

5.8.3.1 canOnlyClickOnce 21

5.8.3.2 dragReversed 21

5.8.3.3 increaseShapeSize 21

5.8.3.4 shakeyShapes 22

5.8.3.5 useSpring 22
5.8.3.6 velocityRatio 22
5.9 MultiStack.Cloud Class Reference 22
5.9.1 Detailed Description 22
5.9.2 Member Data Documentation 22
5.9.2.1 maxSpeed 22
5.9.2.2 minSpeed 23
5.10 MultiStack.CloudController Class Reference 23
5.10.1 Detailed Description 23
5.10.2 Member Function Documentation 23
5.10.2.1 PoolCloud(GameObject cloud) 23
5.10.3 Member Data Documentation 24
5.10.3.1 cloudPrefabs 24
5.10.3.2 maxTimeBetweenClouds 24
5.10.3.3 minTimeBetweenClouds 24
5.10.3.4 yRandomOffset 24
5.11 MultiStack.CloudReturner Class Reference 24
5.11.1 Detailed Description 24
5.11.2 Member Data Documentation 25
5.11.2.1 cloudController 25
5.12 MultiStack.DataPersistence Class Reference 25
5.12.1 Detailed Description 25
5.12.2 Member Function Documentation 25
5.12.2.1 Load() 25
5.12.2.2 Save(int height, int round) 26
5.12.3 Property Documentation 27
5.12.3.1 height 27
5.12.3.2 instance 27
5.12.3.3 round 27
5.13 MultiStack.Destroy Class Reference 27
5.13.1 Detailed Description 27
5.13.2 Member Function Documentation 27
5.13.2.1 ExecuteDestroy() 28
5.14 MultiStack.ExplosiveObject Class Reference 28
5.14.1 Detailed Description 29
5.14.2 Member Function Documentation 29
5.14.2.1 Activate() 29
5.14.2.2 Execute() 29
5.14.3 Member Data Documentation 29
5.14.3.1 explodedObjects 29

5.14.3.2 explosionAudioClip 29
5.14.3.3 explosionPrefab 29
5.14.3.4 explosiveForce 29
5.14.3.5 explosiveRadius 29
5.14.3.6 explosiveSprite 29
5.14.3.7 requiredForceToExplode 29
5.14.4 Property Documentation 29
5.14.4.1 readyToBeActivated 29
5.15 MultiStack.ExplosiveShapeModifier Class Reference 30
5.15.1 Detailed Description 30
5.15.2 Member Function Documentation 30
5.15.2.1 Activate() 30
5.15.2.2 ConditionsMet() 30
5.15.2.3 Deactivate() 31
5.16 MultiStack.FallHandler Class Reference 31
5.16.1 Detailed Description 31
5.16.2 Member Data Documentation 31
5.16.2.1 boxMask 31
5.16.2.2 panSpeed 31
5.16.2.3 requiredRigidBodyYVelocity 32
5.17 MultiStack.GameManager Class Reference 32
5.17.1 Detailed Description 33
5.17.2 Member Function Documentation 33
5.17.2.1 OnGameOver(GameOverType gameOverType=GameOverType.ShapeOffScreen) 33
5.17.2.2 OnRoundOver() 33
5.17.3 Member Data Documentation 33
5.17.3.1 gameModifierManager 33
5.17.3.2 gameOverUI 33
5.17.3.3 infoText 33
5.17.3.4 stageSelector 33
5.17.3.5 uiFlash 33
5.17.4 Property Documentation 33
5.17.4.1 instance 33
5.17.4.2 isGameOver 34
5.18 MultiStack.GameModifier Class Reference 34
5.18.1 Detailed Description 35
5.18.2 Member Function Documentation 35
5.18.2.1 Activate() 35
5.18.2.2 ConditionsMet() 35
5.18.2.3 Deactivate() 35

5.18.3 Member Data Documentation . . . 35
5.18.3.1 IsEnabled . . . 35
5.18.3.2 modifierName . . . 36
5.19 MultiStack.GameModifierManager Class Reference . . . 36
5.19.1 Detailed Description . . . 36
5.19.2 Member Function Documentation . . . 36
5.19.2.1 ApplyNewModifier() . . . 36
5.20 MultiStack.GameOverHandler Class Reference . . . 36
5.20.1 Detailed Description . . . 37
5.21 MultiStack.GameOverUI Class Reference . . . 37
5.21.1 Detailed Description . . . 38
5.21.2 Member Function Documentation . . . 38
5.21.2.1 Activate() . . . 38
5.21.2.2 RestartScene() . . . 38
5.21.2.3 ReturnToMainMenu() . . . 38
5.21.3 Member Data Documentation . . . 38
5.21.3.1 heightReachedText . . . 38
5.21.3.2 heightReachedTitle . . . 38
5.21.3.3 mainMenuButton . . . 38
5.21.3.4 newRecord . . . 38
5.21.3.5 nonGameoverUIText . . . 38
5.21.3.6 playerLostText . . . 38
5.21.3.7 playerLostTitle . . . 38
5.21.3.8 restartButton . . . 39
5.21.3.9 roundsReachedText . . . 39
5.21.3.10 roundsReachedTitle . . . 39
5.21.3.11 title . . . 39
5.22 MultiStack.GameText Class Reference . . . 39
5.22.1 Detailed Description . . . 40
5.22.2 Member Function Documentation . . . 40
5.22.2.1 ChangeFontColour(Color colour) . . . 40
5.22.2.2 HideText() . . . 40
5.22.2.3 HideTextAfterTime(float seconds) . . . 40
5.22.2.4 UpdateText(string msg, Color colour) . . . 40
5.22.2.5 UpdateText(string msg) . . . 40
5.22.3 Member Data Documentation . . . 40
5.22.3.1 clearOnStart . . . 40
5.22.4 Property Documentation . . . 41
5.22.4.1 text . . . 41
5.23 MultiStack.GlassObject Class Reference . . . 41

5.23.1 Detailed Description 42
5.23.2 Member Function Documentation 42
5.23.2.1 Activate() 42
5.23.3 Member Data Documentation 42
5.23.3.1 boxMask 42
5.23.3.2 brokenStages 42
5.23.3.3 crackedAudioClip 42
5.23.3.4 initialGlassSprite 42
5.23.3.5 requiredForceToCrack 42
5.23.3.6 requiredMassToCrack 42
5.23.3.7 smashedAudioClip 42
5.23.3.8 smashedObjectPrefabs 42
5.23.4 Property Documentation 43
5.23.4.1 readyToBeActivated 43
5.24 MultiStack.GlassShapeModifier Class Reference 43
5.24.1 Detailed Description 43
5.24.2 Member Function Documentation 43
5.24.2.1 Activate() 43
5.24.2.2 ConditionsMet() 44
5.24.2.3 Deactivate() 44
5.25 MultiStack.GravityModifier Class Reference 44
5.25.1 Detailed Description 44
5.25.2 Member Function Documentation 45
5.25.2.1 Activate() 45
5.25.2.2 Deactivate() 45
5.25.3 Member Data Documentation 45
5.25.3.1 gravityModifier 45
5.26 MultiStack.HeightText Class Reference 45
5.26.1 Detailed Description 45
5.27 MultiStack.Highscore Class Reference 45
5.27.1 Detailed Description 46
5.28 MultiStack.IncreaseShapeSizeWhileHeldModifier Class Reference 46
5.28.1 Detailed Description 46
5.28.2 Member Function Documentation 46
5.28.2.1 Activate() 46
5.28.2.2 Deactivate() 47
5.29 MultiStack.MainMenu Class Reference 47
5.29.1 Detailed Description 47
5.29.2 Member Function Documentation 48
5.29.2.1 PlayButtonPressed() 48

5.29.2.2 PlayerBoxPlaced() 48
5.29.2.3 PlayerBoxRemoved() 48
5.29.3 Member Data Documentation 48
5.29.3.1 numOfPlayersText 48
5.29.3.2 otherText 48
5.29.3.3 playButton 48
5.29.3.4 playerBoxPrefab 48
5.29.3.5 playerBoxSpawnLocations 48
5.29.3.6 titleText 48
5.30 MultiStack.MoveTime Class Reference 48
5.30.1 Detailed Description 49
5.30.2 Member Data Documentation 49
5.30.2.1 lowOnTimeThreshold 49
5.30.2.2 lowTimeTextColour 49
5.30.2.3 normalTimeTextColour 49
5.31 MultiStack.OneShapeModifier Class Reference 49
5.31.1 Detailed Description 50
5.31.2 Member Function Documentation 50
5.31.2.1 Activate() 50
5.31.2.2 Deactivate() 50
5.31.3 Member Data Documentation 50
5.31.3.1 shape 50
5.32 MultiStack.PlayerBoxPlaced Class Reference 50
5.32.1 Detailed Description 51
5.33 MultiStack.PlayerColours Class Reference 51
5.33.1 Detailed Description 51
5.33.2 Member Data Documentation 51
5.33.2.1 colours 51
5.34 MultiStack.PlayerNumberCount Class Reference 51
5.34.1 Property Documentation 52
5.34.1.1 numberOfPlayers 52
5.35 MultiStack.PlayerPhysicsObject Struct Reference 52
5.35.1 Detailed Description 52
5.35.2 Member Data Documentation 52
5.35.2.1 realPrefab 52
5.35.2.2 shape 52
5.35.2.3 simplePrefab 53
5.35.2.4 weight 53
5.35.3 Property Documentation 53
5.35.3.1 instantiatedPrefab 53

5.36 MultiStack.RandomPlayerOrderModifier Class Reference 53
5.36.1 Detailed Description 53
5.36.2 Member Function Documentation 53
5.36.2.1 Activate() 54
5.36.2.2 Deactivate() 54
5.37 MultiStack.ReverseDragModifier Class Reference 54
5.37.1 Detailed Description 54
5.37.2 Member Function Documentation 54
5.37.2.1 Activate() 54
5.37.2.2 Deactivate() 55
5.38 MultiStack.ScoreData Class Reference 55
5.38.1 Detailed Description 55
5.39 MultiStack.ShakeyShapesModifier Class Reference 55
5.39.1 Detailed Description 56
5.39.2 Member Function Documentation 56
5.39.2.1 Activate() 56
5.39.2.2 Deactivate() 56
5.40 MultiStack.ShapeMovementSpeedModifier Class Reference 56
5.40.1 Detailed Description 57
5.40.2 Member Function Documentation 57
5.40.2.1 Activate() 57
5.40.2.2 Deactivate() 57
5.41 MultiStack.ShapeSizeModifier Class Reference 57
5.41.1 Detailed Description 57
5.41.2 Member Function Documentation 58
5.41.2.1 Activate() 58
5.41.2.2 Deactivate() 58
5.42 MultiStack.SpringModifier Class Reference 58
5.42.1 Detailed Description 58
5.42.2 Member Function Documentation 58
5.42.2.1 Activate() 58
5.42.2.2 Deactivate() 59
5.43 MultiStack.StageSelector Class Reference 59
5.43.1 Detailed Description 59
5.43.2 Member Function Documentation 60
5.43.2.1 CallBackOnStageComplete(CallBack callBack) 60
5.43.2.2 SpawnStage() 61
5.43.3 Member Data Documentation 61
5.43.3.1 infoText 61
5.43.3.2 stagePrefabs 61

5.43.3.3 stageText 61
5.43.4 Property Documentation 61
5.43.4.1 stageSpawned 61
5.44 MultiStack.TimeForMoveModifier Class Reference 61
5.44.1 Detailed Description 62
5.44.2 Member Function Documentation 62
5.44.2.1 Activate() 62
5.44.2.2 Deactivate() 62
5.44.3 Member Data Documentation 62
5.44.3.1 changedTime 62
5.45 MultiStack.TurnManager Class Reference 62
5.45.1 Detailed Description 64
5.45.2 Member Function Documentation 64
5.45.2.1 ApplyCatchTheShapeModifier() 64
5.45.2.2 ApplyGravityModifier(float modifier) 64
5.45.2.3 ApplyShapeModifier(Shape shape) 64
5.45.2.4 ApplySizeModifier(float modifier) 65
5.45.2.5 ChangeShapeToExplosive() 65
5.45.2.6 ChangeShapeToGlass() 65
5.45.2.7 DisableCatchTheShapeModifier() 65
5.45.2.8 DisableGravityModifier() 65
5.45.2.9 DisableShapeModifier() 65
5.45.2.10 DisableSizeModifier() 65
5.45.2.11 Initialise(int numOfPlayers) 65
5.45.2.12 ObjectDropped(Rigidbody2D obj) 65
5.45.2.13 ObjectPickedUp(Rigidbody2D obj) 66
5.45.2.14 StartNewRound() 66
5.45.3 Member Data Documentation 66
5.45.3.1 baseTimeForMove 66
5.45.3.2 physicsObjects 66
5.45.3.3 playerColours 66
5.45.3.4 playerLocations 66
5.45.3.5 playerText 66
5.45.3.6 timeRemainingText 66
5.45.4 Property Documentation 66
5.45.4.1 currentEnhancedHeight 66
5.45.4.2 currentPlayer 67
5.45.4.3 currentRound 67
5.45.4.4 explosiveSpriteAvailableForSpawnedObject 67
5.45.4.5 glassSpriteAvailableForSpawnedObject 67

5.45.4.6 instance 67
5.45.4.7 numOfObjectsSpawned 67
5.45.4.8 randomPlayerOrder 67
5.45.4.9 shapeOffsetHeight 67
5.45.4.10 shouldUpdate 67
5.46 MultiStack.UIFlash Class Reference 68
5.46.1 Detailed Description 68
5.46.2 Member Function Documentation 68
5.46.2.1 CallbackOnUIFlashComplete(CallBack callback) 68
5.46.2.2 GameOverUIFlashInSeconds(float seconds) 68
5.46.3 Member Data Documentation 68
5.46.3.1 isMenu 68
5.47 MultiStack.Utilities Class Reference 69
5.47.1 Detailed Description 69
5.47.2 Member Function Documentation 69
5.47.2.1 GetRandomRotation2D() 69
5.47.2.2 InvokeMethodEverySeconds(CallBack method, float seconds) 69
5.47.3 Property Documentation 70
5.47.3.1 instance 70

Index **71**

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

MultiStack 9

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- MultiStack.Background . . . 13
- MonoBehaviour
  - MultiStack.BackgroundManager . . . 13
  - MultiStack.CameraManager . . . 14
  - MultiStack.CameraShake . . . 16
  - MultiStack.ClickableObject . . . 19
  - MultiStack.ClickHandler . . . 20
  - MultiStack.Cloud . . . 22
  - MultiStack.CloudController . . . 23
  - MultiStack.CloudReturner . . . 24
  - MultiStack.DataPersistence . . . 25
  - MultiStack.Destroy . . . 27
  - MultiStack.ExplosiveObject . . . 28
  - MultiStack.FallHandler . . . 31
  - MultiStack.GameManager . . . 32
  - MultiStack.GameModifier . . . 34
    - MultiStack.CanOnlyPickupOnceModifier . . . 17
    - MultiStack.CatchTheShapeModifier . . . 18
    - MultiStack.ExplosiveShapeModifier . . . 30
    - MultiStack.GlassShapeModifier . . . 43
    - MultiStack.GravityModifier . . . 44
    - MultiStack.IncreaseShapeSizeWhileHeldModifier . . . 46
    - MultiStack.OneShapeModifier . . . 49
    - MultiStack.RandomPlayerOrderModifier . . . 53
    - MultiStack.ReverseDragModifier . . . 54
    - MultiStack.ShakeyShapesModifier . . . 55
    - MultiStack.ShapeMovementSpeedModifier . . . 56
    - MultiStack.ShapeSizeModifier . . . 57
    - MultiStack.SpringModifier . . . 58
    - MultiStack.TimeForMoveModifier . . . 61
  - MultiStack.GameModifierManager . . . 36
  - MultiStack.GameOverHandler . . . 36
  - MultiStack.GameOverUI . . . 37
  - MultiStack.GameText . . . 39
  - MultiStack.GlassObject . . . 41
  - MultiStack.HeightText . . . 45
  - MultiStack.Highscore . . . 45
  - MultiStack.MainMenu . . . 47

- MultiStack.MoveTime 48
- MultiStack.PlayerBoxPlaced 50
- MultiStack.PlayerColours 51
- MultiStack.PlayerNumberCount 51
- MultiStack.StageSelector 59
- MultiStack.TurnManager 62
- MultiStack.UIFlash 68
- MultiStack.Utilities 69

MultiStack.PlayerPhysicsObject 52

MultiStack.ScoreData 55

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

MultiStack.Background
Holds background specific data. 13

MultiStack.BackgroundManager
Creates the background object at runtime. Selects background from a pool of possible backgrounds. 13

MultiStack.CameraManager
Attached to the main camera in the Game Scene. Handles camera movement on the y axis. . 14

MultiStack.CameraShake
Handles camera shake on explosion. 16

MultiStack.CanOnlyPickupOnceModifier
When enabled the player can only pickup a shape once. Once a shape is dropped it is no longer draggable. Invokes MultiStack.ClickHandler.canOnlyClickOnce 17

MultiStack.CatchTheShapeModifier
When enabled any newly spawned physics objects are dropped straight away rather than waiting for the player to click on/touch them. 18

MultiStack.ClickableObject
Attached to all physics objects. Defines if a object is currently being held. Used by MultiStack.TurnManager to help decide if a players turn is over. Also used to turn off interaction with object by setting MultiStack.ClickableObject.clickable 19

MultiStack.ClickHandler
The main click/drag manager. Handles picking up, dragging, and dropping of physics objects via mouse or touch screen controls. 20

MultiStack.Cloud
Translates a cloud across the screen based on MultiStack.Cloud.minSpeed and MultiStack.Cloud.maxSpeed. 22

MultiStack.CloudController
Handles creation of clouds. 23

MultiStack.CloudReturner
Invokes MultiStack.CloudController.PoolCloud when object enters trigger. This is placed at opposite end of the screen to the MultiStack.CloudController 24

MultiStack.DataPersistence
Data persistence. Handles saving and loading of height and round data. The highest reached height and round are saved and retrieved from file. 25

MultiStack.Destroy
Used during the explosion animation to destroy the gameobject on animation end. 27

MultiStack.ExplosiveObject
Attached to any shape that can be turned into an explosive shape. Handles changing a shapes sprite and exploding. 28

MultiStack.ExplosiveShapeModifier
When enabled and conditions met a random shape is turned into an explosive shape. Invokes MultiStack.TurnManager.ChangeShapeToExplosive. 30

MultiStack.FallHandler
Moves camera position to this position if a physics object enters its trigger. Used to move camera to a falling shape. 31

MultiStack.GameManager
Controls the game flow. Handles beginning the game, spawning the stage, starting new rounds, and ending the game. 32

MultiStack.GameModifier
The abstract base class for every game modifier. Game modifiers are applied each round to change how the game is played. 34

MultiStack.GameModifierManager
Handles game modifiers. These are modififers applied at the beginning of each round (e.g. low gravity). 36

MultiStack.GameOverHandler
Invokes MultiStack.GameManager.OnGameOver when a physics object enters trigger. 36

MultiStack.GameOverUI
Handles the displaying of the UI in the event of a game over. 37

MultiStack.GameText
A simple wrapper for the text object. 39

MultiStack.GlassObject
Attached to any shape that can be turned into a glass shape. Handles changing a shapes sprite and breaking. 41

MultiStack.GlassShapeModifier
When enabled and conditions met a random shape is turned into an glass shape. Invokes MultiStack.TurnManager.ChangeShapeToGlass. 43

MultiStack.GravityModifier
When enabled a gravity modifier is appled to all spawned shapes. 44

MultiStack.HeightText
The UI for the height shown during the game scene. 45

MultiStack.Highscore
The UI for the highest height reached on the Main menu scene. 45

MultiStack.IncreaseShapeSizeWhileHeldModifier
When enabled a shapes size increases while being dragged. 46

MultiStack.MainMenu
Handles the UI for main menu and number of players select scene. 47

MultiStack.MoveTime
Responsible for updating the current players move time. 48

MultiStack.OneShapeModifier
When enabled only one shape specified by MultiStack.OneShapeModifier.shape will be spawned for that round. 49

MultiStack.PlayerBoxPlaced
Used during the number of player selection scene. Handles alerting MultiStack.MainMenu when a box is placed by invoking MultiStack.MainMenu.PlayerBoxPlaced and MultiStack.MainMenu.↩ PlayerBoxRemoved. 50

MultiStack.PlayerColours
Holds the coours for each player. 51

MultiStack.PlayerNumberCount 51

MultiStack.PlayerPhysicsObject
A structure defining a physics object. 52

MultiStack.RandomPlayerOrderModifier
When enabled the order of players is randomised for this round. 53

MultiStack.ReverseDragModifier
When enabled drag is reversed e.g. when the player drags the mouse/finger left the shape moves right. 54

MultiStack.ScoreData
Class to store serializable height/round data. 55

MultiStack.ShakeyShapesModifier
When enabled a random offset is added to a dragged shape each timestep. 55

MultiStack.ShapeMovementSpeedModifier
When enabled the drag speed is changed to MultiStack.ShapeMovementSpeedModifier.shape↩ MovementSpeed. 56

MultiStack.ShapeSizeModifier
When enabled all newly spawned shapes size is multiplied by MultiStack.ShapeSizeModifier.↩ sizeModifier. 57

MultiStack.SpringModifier
When enabled the shapes are attached to springs when being dragged. This results in a harder to control shape. 58

MultiStack.StageSelector
Used to select a stage for the game. A random stage is selected at the beginning of each game. 59

MultiStack.TimeForMoveModifier
When enabled the time a player has to complete their turn is changed to MultiStack.TimeFor↩ MoveModifier.changedTime. 61

MultiStack.TurnManager
Handles player turns an spawning of player physics objects objects. Manages and maintains a list of spawned physics objects. Also handles applying certain MultiStack.GameModifier e.↩ g. when MultiStack.ExplosiveShapeModifier is applied, this class will loop through the spawned shape list and convert one shape into an explosive shape. 62

MultiStack.UIFlash
User interface flash. Acts as an overlay for the main menu and gameplay scene. 68

MultiStack.Utilities
Useful methods used by a number of classes. 69

Chapter 4

Namespace Documentation

4.1 MultiStack Namespace Reference

Classes

- struct Background

  Holds background specific data.
- class BackgroundManager

  Creates the background object at runtime. Selects background from a pool of possible backgrounds.
- class CameraManager

  Attached to the main camera in the Game Scene. Handles camera movement on the y axis.
- class CameraShake

  Handles camera shake on explosion.
- class CanOnlyPickupOnceModifier

  When enabled the player can only pickup a shape once. Once a shape is dropped it is no longer draggable. Invokes MultiStack.ClickHandler.canOnlyClickOnce
- class CatchTheShapeModifier

  When enabled any newly spawned physics objects are dropped straight away rather than waiting for the player to click on/touch them.
- class ClickableObject

  Attached to all physics objects. Defines if a object is currently being held. Used by MultiStack.TurnManager to help decide if a players turn is over. Also used to turn off interaction with object by setting MultiStack.ClickableObject.↩ clickable
- class ClickHandler

  The main click/drag manager. Handles picking up, dragging, and dropping of physics objects via mouse or touch screen controls.
- class Cloud

  Translates a cloud across the screen based on MultiStack.Cloud.minSpeed and MultiStack.Cloud.maxSpeed.
- class CloudController

  Handles creation of clouds.
- class CloudReturner

  Invokes MultiStack.CloudController.PoolCloud when object enters trigger. This is placed at opposite end of the screen to the MultiStack.CloudController .
- class DataPersistence

  Data persistence. Handles saving and loading of height and round data. The highest reached height and round are saved and retrieved from file.
- class Destroy

  Used during the explosion animation to destroy the gameobject on animation end.
- class ExplosiveObject

Attached to any shape that can be turned into an explosive shape. Handles changing a shapes sprite and exploding.

- class ExplosiveShapeModifier

  When enabled and conditions met a random shape is turned into an explosive shape. Invokes MultiStack.TurnManager.ChangeShapeToExplosive.

- class **Extensions**

  useful Extension methods.

- class FallHandler

  Moves camera position to this position if a physics object enters its trigger. Used to move camera to a falling shape.

- class GameManager

  Controls the game flow. Handles beginning the game, spawning the stage, starting new rounds, and ending the game.

- class GameModifier

  The abstract base class for every game modifier. Game modifiers are applied each round to change how the game is played.

- class GameModifierManager

  Handles game modifiers. These are modififers applied at the beginning of each round (e.g. low gravity).

- class GameOverHandler

  Invokes MultiStack.GameManager.OnGameOver when a physics object enters trigger.

- class GameOverUI

  Handles the displaying of the UI in the event of a game over.

- class GameText

  A simple wrapper for the text object.

- class GlassObject

  Attached to any shape that can be turned into a glass shape. Handles changing a shapes sprite and breaking.

- class GlassShapeModifier

  When enabled and conditions met a random shape is turned into an glass shape. Invokes MultiStack.TurnManager.ChangeShapeToGlass.

- class GravityModifier

  When enabled a gravity modifier is appled to all spawned shapes.

- class HeightText

  The UI for the height shown during the game scene.

- class Highscore

  The UI for the highest height reached on the Main menu scene.

- class IncreaseShapeSizeWhileHeldModifier

  When enabled a shapes size increases while being dragged.

- class MainMenu

  Handles the UI for main menu and number of players select scene.

- class MoveTime

  Responsible for updating the current players move time.

- class OneShapeModifier

  When enabled only one shape specified by MultiStack.OneShapeModifier.shape will be spawned for that round.

- class PlayerBoxPlaced

  Used during the number of player selection scene. Handles alerting MultiStack.MainMenu when a box is placed by invoking MultiStack.MainMenu.PlayerBoxPlaced and MultiStack.MainMenu.PlayerBoxRemoved.

- class PlayerColours

  Holds the coours for each player.

- class PlayerNumberCount
- struct PlayerPhysicsObject

  A structure defining a physics object.

- class RandomPlayerOrderModifier

  When enabled the order of players is randomised for this round.

- class ReverseDragModifier

When enabled drag is reversed e.g. when the player drags the mouse/finger left the shape moves right.

- class ScoreData

  Class to store serializable height/round data.
- class ShakeyShapesModifier

  When enabled a random offset is added to a dragged shape each timestep.
- class ShapeMovementSpeedModifier

  When enabled the drag speed is changed to MultiStack.ShapeMovementSpeedModifier.shapeMovementSpeed.
- class ShapeSizeModifier

  When enabled all newly spawned shapes size is multiplied by MultiStack.ShapeSizeModifier.sizeModifier.
- class SpringModifier

  When enabled the shapes are attached to springs when being dragged. This results in a harder to control shape.
- class StageSelector

  Used to select a stage for the game. A random stage is selected at the beginning of each game.
- class TimeForMoveModifier

  When enabled the time a player has to complete their turn is changed to MultiStack.TimeForMoveModifier.changed↩Time.
- class TurnManager

  Handles player turns an spawning of player physics objects objects. Manages and maintains a list of spawned physics objects. Also handles applying certain MultiStack.GameModifier e.g. when MultiStack.ExplosiveShapeModifier is applied, this class will loop through the spawned shape list and convert one shape into an explosive shape.
- class UIFlash

  User interface flash. Acts as an overlay for the main menu and gameplay scene.
- class Utilities

  Useful methods used by a number of classes.

Enumerations

- enum GameOverType { **ShapeOffScreen**, **TimeUp** }

  Game over type.
- enum Shape {
  Square, **Circle**, **Triangle**, **Rectangle**,
  Irregular }

  Physic object shapes.

Functions

- delegate void CallBack ()

  A delegate method used when specific actions occur. For example if this delegate is passed to MultiStack.Camera↩Manager.CallBackOnTargetReached then it is invoked when the camera reaches its target y position.

4.1.1 Enumeration Type Documentation

4.1.1.1 enum MultiStack.GameOverType [strong]

Game over type.

4.1.1.2 enum MultiStack.Shape [strong]

Physic object shapes.

4.1.2 Function Documentation

4.1.2.1 delegate void MultiStack.CallBack ( )

A delegate method used when specific actions occur. For example if this delegate is passed to MultiStack.CameraManager.CallBackOnTargetReached then it is invoked when the camera reaches its target y position.

Chapter 5

Class Documentation

5.1 MultiStack.Background Struct Reference

Holds background specific data.

Public Attributes

- GameObject backgroundPrefab

  The main background prefab.
- GameObject backgroundTilePrefab

  The prefab used to extend current background in the positive y axis.

5.1.1 Detailed Description

Holds background specific data.

5.1.2 Member Data Documentation

5.1.2.1 GameObject MultiStack.Background.backgroundPrefab

The main background prefab.

5.1.2.2 GameObject MultiStack.Background.backgroundTilePrefab

The prefab used to extend current background in the positive y axis.

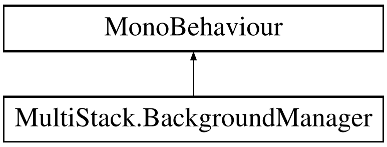
The documentation for this struct was generated from the following file:

- MultiplayerStacker/Scripts/Environment/Background/Background.cs

5.2 MultiStack.BackgroundManager Class Reference

Creates the background object at runtime. Selects background from a pool of possible backgrounds.

Inheritance diagram for MultiStack.BackgroundManager:



Public Attributes

- Background[ ] backgrounds

  The possible background objects.

5.2.1 Detailed Description

Creates the background object at runtime. Selects background from a pool of possible backgrounds.

5.2.2 Member Data Documentation

5.2.2.1 Background [ ] MultiStack.BackgroundManager.backgrounds

The possible background objects.

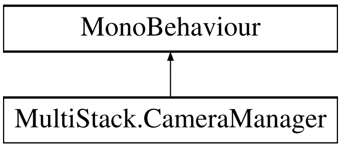
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Environment/Background/BackgroundManager.cs

5.3 MultiStack.CameraManager Class Reference

Attached to the main camera in the Game Scene. Handles camera movement on the y axis.

Inheritance diagram for MultiStack.CameraManager:



Public Member Functions

- void RequestNewTarget (float target)

  Requests a new target y position. Confines camera movement to specific areas dependent on current stack height. Limits target position to within +/- 1.5f of current target, not higher than the stacked shapes + 3f, and greater than 0.
- void SetTarget (float target, float panspeed=1f)

  Sets the a new target y position. No constraints are placed on target.
- void ResetTarget (float panSpeedToTarget=1f)

  Resets the target y position to 0.
- void StartPanning (float panSpeed=1f)

  Begins camera movement.
- void CallBackOnTargetReached (CallBack callBack)

  Injects a method to be called when target is reached. Method is invoked once when target reached and then set to null. Call this method before setting a new target.

- void ResetPanSpeedOnTargetReached ()

  Camera movement speed is reset when target has been reached. Call this before setting a new target if you want the camera movement speed to return to its previous value.

Public Attributes

- float targetY = 0

  The target y position.

Properties

- bool hasReachedTarget `[get]`

  Gets a value indicating whether this MultiStack.CameraManager has come within 0.2f of target y position.
- bool shouldPan `[set]`

  Sets a value indicating whether this MultiStack.CameraManager should pan to target y position.

5.3.1 Detailed Description

Attached to the main camera in the Game Scene. Handles camera movement on the y axis.

5.3.2 Member Function Documentation

5.3.2.1 void MultiStack.CameraManager.CallBackOnTargetReached (CallBack *callBack*)

Injects a method to be called when target is reached. Method is invoked once when target reached and then set to null. Call this method before setting a new target.

Parameters

callBack	Call back.

5.3.2.2 void MultiStack.CameraManager.RequestNewTarget (float *target*)

Requests a new target y position. Confines camera movement to specific areas dependent on current stack height. Limits target position to within +/- 1.5f of current target, not higher than the stacked shapes + 3f, and greater than 0.

Parameters

target	Target y position.

5.3.2.3 void MultiStack.CameraManager.ResetPanSpeedOnTargetReached ( )

Camera movement speed is reset when target has been reached. Call this before setting a new target if you want the camera movement speed to return to its previous value.

5.3.2.4 void MultiStack.CameraManager.ResetTarget (float *panSpeedToTarget* = `1f`)

Resets the target y position to 0.

Parameters

panSpeedTo↩ Target	Pan speed to target.

5.3.2.5 void MultiStack.CameraManager.SetTarget (float *target,* float *panspeed* = `1f`)

Sets the a new target y position. No constraints are placed on target.

Parameters

target	Target.
panspeed	The speed to move to new target.

5.3.2.6 void MultiStack.CameraManager.StartPanning (float *panSpeed* = `1f`)

Begins camera movement.

Parameters

panSpeed	Pan speed to target.

5.3.3 Member Data Documentation

5.3.3.1 float MultiStack.CameraManager.targetY = 0

The target y position.

5.3.4 Property Documentation

5.3.4.1 bool MultiStack.CameraManager.hasReachedTarget `[get]`

Gets a value indicating whether this MultiStack.CameraManager has come within 0.2f of target y position.

`true` if has reached target; otherwise, `false`.

5.3.4.2 bool MultiStack.CameraManager.shouldPan `[set]`

Sets a value indicating whether this MultiStack.CameraManager should pan to target y position.

`true` if should pan; otherwise, `false`.

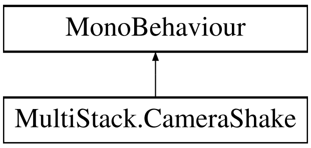
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Camera/CameraManager.cs

5.4 MultiStack.CameraShake Class Reference

Handles camera shake on explosion.

Inheritance diagram for MultiStack.CameraShake:



Public Member Functions

- void Shake (float shakeIntensity=0.05f, float shakeDecay=0.004f)

  Begins the shake effect with the desired intensity and decay.

5.4.1 Detailed Description

Handles camera shake on explosion.

5.4.2 Member Function Documentation

5.4.2.1 void MultiStack.CameraShake.Shake (float *shakeIntensity* = `0.05f`, float *shakeDecay* = `0.004f`)

Begins the shake effect with the desired intensity and decay.

Parameters

shakeIntensity	Shake intensity.
shakeDecay	Shake decay.

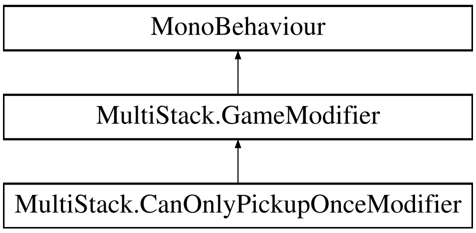
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/CameraShake.cs

5.5 MultiStack.CanOnlyPickupOnceModifier Class Reference

When enabled the player can only pickup a shape once. Once a shape is dropped it is no longer draggable. Invokes MultiStack.ClickHandler.canOnlyClickOnce

Inheritance diagram for MultiStack.CanOnlyPickupOnceModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- ClickHandler **clickHandler**

5.5.1 Detailed Description

When enabled the player can only pickup a shape once. Once a shape is dropped it is no longer draggable. Invokes MultiStack.ClickHandler.canOnlyClickOnce

5.5.2 Member Function Documentation

5.5.2.1 override void MultiStack.CanOnlyPickupOnceModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.5.2.2 override void MultiStack.CanOnlyPickupOnceModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

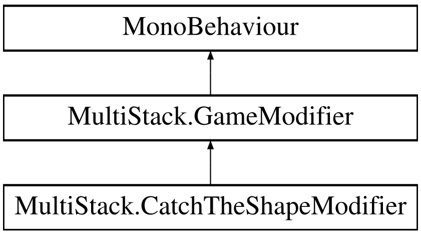
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/CanOnlyPickupOnceModifier.cs

5.6 MultiStack.CatchTheShapeModifier Class Reference

When enabled any newly spawned physics objects are dropped straight away rather than waiting for the player to click on/touch them.

Inheritance diagram for MultiStack.CatchTheShapeModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Additional Inherited Members

5.6.1 Detailed Description

When enabled any newly spawned physics objects are dropped straight away rather than waiting for the player to click on/touch them.

5.6.2 Member Function Documentation

5.6.2.1 override void MultiStack.CatchTheShapeModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.6.2.2 override void MultiStack.CatchTheShapeModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

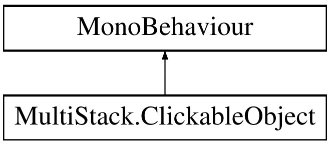
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/CatchTheShapeModifier.cs

5.7 MultiStack.ClickableObject Class Reference

Attached to all physics objects. Defines if a object is currently being held. Used by MultiStack.TurnManager to help decide if a players turn is over. Also used to turn off interaction with object by setting MultiStack.ClickableObject.↩ clickable

Inheritance diagram for MultiStack.ClickableObject:



Public Member Functions

- void Clicked ()

  Physics obkect has been 'clicked' on.

Properties

- bool clickable [get]

  Gets a value indicating whether this MultiStack.ClickableObject is selectable.

5.7.1 Detailed Description

Attached to all physics objects. Defines if a object is currently being held. Used by MultiStack.TurnManager to help decide if a players turn is over. Also used to turn off interaction with object by setting MultiStack.ClickableObject.↩ clickable

5.7.2 Member Function Documentation

5.7.2.1 void MultiStack.ClickableObject.Clicked ( )

Physics obkect has been 'clicked' on.

5.7.3 Property Documentation

5.7.3.1 bool MultiStack.ClickableObject.clickable [get]

Gets a value indicating whether this MultiStack.ClickableObject is selectable.

`true` if clickable; otherwise, `false`.

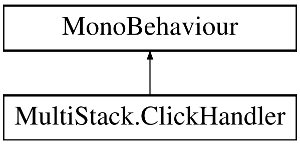
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/ClickableObject.cs

5.8 MultiStack.ClickHandler Class Reference

The main click/drag manager. Handles picking up, dragging, and dropping of physics objects via mouse or touch screen controls.

Inheritance diagram for MultiStack.ClickHandler:



Public Attributes

- bool isMainMenuScene = false

  Sets whether this is main menu scene. Camera does not follow picked up objects if it is the main menu scene.
- LayerMask boxMask

  The layermask of the physics objects.
- LineRenderer dragLine

  The line renderer for the line drawn between cursor/finger and grabbed object.
- PlayerColours playerColours

  Reference to player colours, used to change the colour of MultiStack.ClickHandler.dragLine based on current player.

Properties

- bool useSpring [get, set]

  Gets or sets a value indicating whether this MultiStack.ClickHandler should use a spring when dragging an object.
- float velocityRatio [get, set]

  Gets or sets the velocity ratio used when dragging an object.
- bool dragReversed [set]

  Sets a value indicating whether this MultiStack.ClickHandler has reversed drag i.e. when the player drags right, the shape moves left.
- bool shakeyShapes [get, set]

Gets or sets a value indicating whether this MultiStack.ClickHandler should use shakey shapes. When enabled a random offset is added to the shapes position each time step.

- bool increaseShapeSize `[get, set]`

  Gets or sets a value indicating whether this MultiStack.ClickHandler should increase shape size when a shape is held.

- bool canOnlyClickOnce `[get, set]`

  Gets or sets a value indicating whether this MultiStack.ClickHandler can only grab an object once.

5.8.1 Detailed Description

The main click/drag manager. Handles picking up, dragging, and dropping of physics objects via mouse or touch screen controls.

5.8.2 Member Data Documentation

5.8.2.1 LayerMask MultiStack.ClickHandler.boxMask

The layermask of the physics objects.

5.8.2.2 LineRenderer MultiStack.ClickHandler.dragLine

The line renderer for the line drawn between cursor/finger and grabbed object.

5.8.2.3 bool MultiStack.ClickHandler.isMainMenuScene = false

Sets whether this is main menu scene. Camera does not follow picked up objects if it is the main menu scene.

5.8.2.4 PlayerColours MultiStack.ClickHandler.playerColours

Reference to player colours, used to change the colour of MultiStack.ClickHandler.dragLine based on current player.

5.8.3 Property Documentation

5.8.3.1 bool MultiStack.ClickHandler.canOnlyClickOnce `[get], [set]`

Gets or sets a value indicating whether this MultiStack.ClickHandler can only grab an object once.

`true` if can only click once; otherwise, `false`.

5.8.3.2 bool MultiStack.ClickHandler.dragReversed `[set]`

Sets a value indicating whether this MultiStack.ClickHandler has reversed drag i.e. when the player drags right, the shape moves left.

`true` if drag reversed; otherwise, `false`.

5.8.3.3 bool MultiStack.ClickHandler.increaseShapeSize `[get], [set]`

Gets or sets a value indicating whether this MultiStack.ClickHandler should increase shape size when a shape is held.

`true` if increase shape size; otherwise, `false`.

5.8.3.4 bool MultiStack.ClickHandler.shakeyShapes [get], [set]

Gets or sets a value indicating whether this MultiStack.ClickHandler should use shakey shapes. When enabled a random offset is added to the shapes position each time step.

`true` if shakey shapes; otherwise, `false`.

5.8.3.5 bool MultiStack.ClickHandler.useSpring [get], [set]

Gets or sets a value indicating whether this MultiStack.ClickHandler should use a spring when dragging an object.

`true` if use spring; otherwise, `false`.

5.8.3.6 float MultiStack.ClickHandler.velocityRatio [get], [set]

Gets or sets the velocity ratio used when dragging an object.

The velocity ratio.

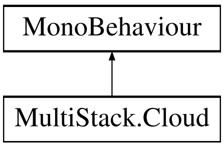
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/ClickHandler.cs

5.9 MultiStack.Cloud Class Reference

Translates a cloud across the screen based on MultiStack.Cloud.minSpeed and MultiStack.Cloud.maxSpeed.

Inheritance diagram for MultiStack.Cloud:



Public Attributes

- float minSpeed

  The minimum speed of the cloud.
- float maxSpeed

  The maximum speed of the cloud.

5.9.1 Detailed Description

Translates a cloud across the screen based on MultiStack.Cloud.minSpeed and MultiStack.Cloud.maxSpeed.

5.9.2 Member Data Documentation

5.9.2.1 float MultiStack.Cloud.maxSpeed

The maximum speed of the cloud.

5.9.2.2 float MultiStack.Cloud.minSpeed

The minimum speed of the cloud.

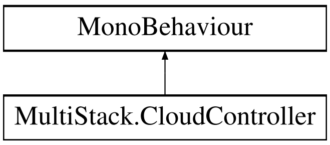
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Environment/Clouds/Cloud.cs

5.10 MultiStack.CloudController Class Reference

Handles creation of clouds.

Inheritance diagram for MultiStack.CloudController:



Public Member Functions

- void PoolCloud (GameObject cloud)

  Adds a cloud to a pool to be reused when a new cloud is required.

Public Attributes

- float maxTimeBetweenClouds

  The maximum time between spawning clouds.
- float minTimeBetweenClouds

  The minimum time between spawning clouds.
- float yRandomOffset = 5f

  A random offset between -yRandomOffset and +yRandomOffset is applied to the y position of all spawned clouds.
- GameObject[ ] cloudPrefabs

  The cloud prefabs to be instantiated.

5.10.1 Detailed Description

Handles creation of clouds.

5.10.2 Member Function Documentation

5.10.2.1 void MultiStack.CloudController.PoolCloud (GameObject *cloud*)

Adds a cloud to a pool to be reused when a new cloud is required.

Parameters

cloud	Cloud.

5.10.3 Member Data Documentation

5.10.3.1 GameObject [ ] MultiStack.CloudController.cloudPrefabs

The cloud prefabs to be instantiated.

5.10.3.2 float MultiStack.CloudController.maxTimeBetweenClouds

The maximum time between spawning clouds.

5.10.3.3 float MultiStack.CloudController.minTimeBetweenClouds

The minimum time between spawning clouds.

5.10.3.4 float MultiStack.CloudController.yRandomOffset = 5f

A random offset between -yRandomOffset and +yRandomOffset is applied to the y position of all spawned clouds.

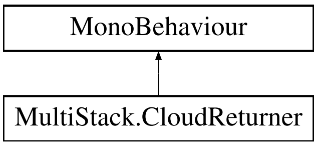
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Environment/Clouds/CloudController.cs

5.11 MultiStack.CloudReturner Class Reference

Invokes MultiStack.CloudController.PoolCloud when object enters trigger. This is placed at opposite end of the screen to the MultiStack.CloudController .

Inheritance diagram for MultiStack.CloudReturner:



Public Attributes

- CloudController cloudController

  Reference to the cloud controller.

5.11.1 Detailed Description

Invokes MultiStack.CloudController.PoolCloud when object enters trigger. This is placed at opposite end of the screen to the MultiStack.CloudController .

5.11.2 Member Data Documentation

5.11.2.1 CloudController MultiStack.CloudReturner.cloudController

Reference to the cloud controller.

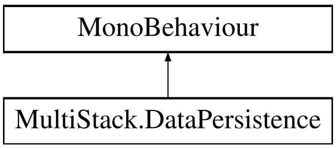
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Environment/Clouds/CloudReturner.cs

5.12 MultiStack.DataPersistence Class Reference

Data persistence. Handles saving and loading of height and round data. The highest reached height and round are saved and retrieved from file.

Inheritance diagram for MultiStack.DataPersistence:



Public Member Functions

- void Load ()

  Load the highest round and height from file.
- void Save (int height, int round)

  If height or round score greater than stored height/round then it is saved to file.

Properties

- int height `[get]`

  Gets the height. Loaded from file.
- int round `[get]`

  Gets the round. Loaded from file.
- static DataPersistence instance `[get]`

  Gets the instance of this class. Can be accessed from any script.

5.12.1 Detailed Description

Data persistence. Handles saving and loading of height and round data. The highest reached height and round are saved and retrieved from file.

5.12.2 Member Function Documentation

5.12.2.1 void MultiStack.DataPersistence.Load ( )

Load the highest round and height from file.

5.12.2.2 void MultiStack.DataPersistence.Save (int *height,* int *round*)

If height or round score greater than stored height/round then it is saved to file.

Parameters

score	Score.

5.12.3 Property Documentation

5.12.3.1 int MultiStack.DataPersistence.height `[get]`

Gets the height. Loaded from file.

The score.

5.12.3.2 DataPersistence MultiStack.DataPersistence.instance `[static], [get]`

Gets the instance of this class. Can be accessed from any script.

The instance.

5.12.3.3 int MultiStack.DataPersistence.round `[get]`

Gets the round. Loaded from file.

The round.

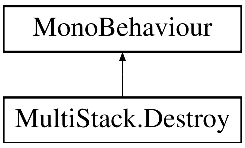
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/High Score/DataPersistence.cs

5.13 MultiStack.Destroy Class Reference

Used during the explosion animation to destroy the gameobject on animation end.

Inheritance diagram for MultiStack.Destroy:



Public Member Functions

- void ExecuteDestroy ()

  Executes the destroy command on gameobject.

5.13.1 Detailed Description

Used during the explosion animation to destroy the gameobject on animation end.

5.13.2 Member Function Documentation

5.13.2.1 void MultiStack.Destroy.ExecuteDestroy ( )

Executes the destroy command on gameobject.

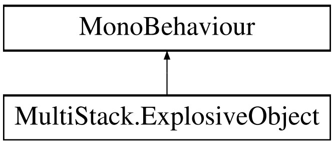
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Animation/Destroy.cs

5.14 MultiStack.ExplosiveObject Class Reference

Attached to any shape that can be turned into an explosive shape. Handles changing a shapes sprite and exploding.

Inheritance diagram for MultiStack.ExplosiveObject:



Public Member Functions

- void Activate ()

  Activate this instance and changes the shapes sprite to that of MultiStack.ExplosiveObject.explosiveSprite.
- void Execute ()

  Explodes this shape. Plays explosion audio, instantiates explosion fragments, plays explosion animation, applies explosive force to all shapes within the radius, and enables camera shake.

Public Attributes

- GameObject explosionPrefab

  The prfab for the explosion animation.
- Sprite explosiveSprite

  The shapes explosive sprite.
- GameObject[ ] explodedObjects

  Object prefabs to be spawned when the shape is exploded.
- float explosiveForce = 500f

  The force applied to shapes within the MultiStack.ExplosiveObject.explosiveRadius
- float explosiveRadius = 2f

  The explosion radius. Force is applied to all other shapes within this radius.
- float requiredForceToExplode = 5.3f

  The force if contact between this shape and another shape to trigger the explosion.
- AudioClip explosionAudioClip

  An audio clip to play when an explosion occurs.

Properties

- bool readyToBeActivated `[get]`

  Gets a value indicating whether this MultiStack.ExplosiveObject is ready to be activated.

5.14.1 Detailed Description

Attached to any shape that can be turned into an explosive shape. Handles changing a shapes sprite and exploding.

5.14.2 Member Function Documentation

5.14.2.1 void MultiStack.ExplosiveObject.Activate ( )

Activate this instance and changes the shapes sprite to that of MultiStack.ExplosiveObject.explosiveSprite.

5.14.2.2 void MultiStack.ExplosiveObject.Execute ( )

Explodes this shape. Plays explosion audio, instantiates explosion fragments, plays explosion animation, applies explosive force to all shapes within the radius, and enables camera shake.

5.14.3 Member Data Documentation

5.14.3.1 GameObject [ ] MultiStack.ExplosiveObject.explodedObjects

Object prefabs to be spawned when the shape is exploded.

5.14.3.2 AudioClip MultiStack.ExplosiveObject.explosionAudioClip

An audio clip to play when an explosion occurs.

5.14.3.3 GameObject MultiStack.ExplosiveObject.explosionPrefab

The prfab for the explosion animation.

5.14.3.4 float MultiStack.ExplosiveObject.explosiveForce = 500f

The force applied to shapes within the MultiStack.ExplosiveObject.explosiveRadius

5.14.3.5 float MultiStack.ExplosiveObject.explosiveRadius = 2f

The explosion radius. Force is applied to all other shapes within this radius.

5.14.3.6 Sprite MultiStack.ExplosiveObject.explosiveSprite

The shapes explosive sprite.

5.14.3.7 float MultiStack.ExplosiveObject.requiredForceToExplode = 5.3f

The force if contact between this shape and another shape to trigger the explosion.

5.14.4 Property Documentation

5.14.4.1 bool MultiStack.ExplosiveObject.readyToBeActivated `[get]`

Gets a value indicating whether this MultiStack.ExplosiveObject is ready to be activated.

`true` if ready to be activated; otherwise, `false`.

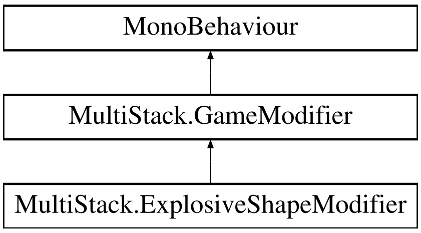
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Objects/ExplosiveObject.cs

5.15 MultiStack.ExplosiveShapeModifier Class Reference

When enabled and conditions met a random shape is turned into an explosive shape. Invokes MultiStack.TurnManager.ChangeShapeToExplosive.

Inheritance diagram for MultiStack.ExplosiveShapeModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.
- override bool ConditionsMet ()

  Returns true if any spawned shape can be turned into an explosive shape.

Additional Inherited Members

5.15.1 Detailed Description

When enabled and conditions met a random shape is turned into an explosive shape. Invokes MultiStack.TurnManager.ChangeShapeToExplosive.

5.15.2 Member Function Documentation

5.15.2.1 override void MultiStack.ExplosiveShapeModifier.Activate ( ) `[virtual]`

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.15.2.2 override bool MultiStack.ExplosiveShapeModifier.ConditionsMet ( ) `[virtual]`

Returns true if any spawned shape can be turned into an explosive shape.

Returns

true if a shape can be turned into explosive shape.

`false`

Reimplemented from MultiStack.GameModifier.

5.15.2.3 override void MultiStack.ExplosiveShapeModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

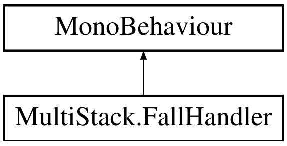
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/ExplosiveShapeModifier.cs

5.16 MultiStack.FallHandler Class Reference

Moves camera position to this position if a physics object enters its trigger. Used to move camera to a falling shape.

Inheritance diagram for MultiStack.FallHandler:



Public Attributes

- LayerMask boxMask

    The layermask of the physics object.
- float requiredRigidBodyYVelocity = 4f

    The required physics shape y velocity for the camera to focus on the object.
- float panSpeed = 2f

    The speed at which the camera pans to the object.

5.16.1 Detailed Description

Moves camera position to this position if a physics object enters its trigger. Used to move camera to a falling shape.

5.16.2 Member Data Documentation

5.16.2.1 LayerMask MultiStack.FallHandler.boxMask

The layermask of the physics object.

5.16.2.2 float MultiStack.FallHandler.panSpeed = 2f

The speed at which the camera pans to the object.

5.16.2.3 float MultiStack.FallHandler.requiredRigidBodyYVelocity = 4f

The required physics shape y velocity for the camera to focus on the object.

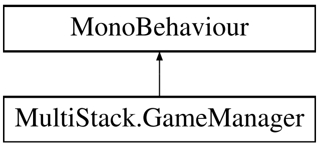
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Managers/FallHandler.cs

5.17 MultiStack.GameManager Class Reference

Controls the game flow. Handles beginning the game, spawning the stage, starting new rounds, and ending the game.

Inheritance diagram for MultiStack.GameManager:



Public Member Functions

- void OnGameOver (GameOverType gameOverType=GameOverType.ShapeOffScreen)

  Called in the event of a game over (either a players timer reaches zero or a shape falls out of bounds). Disables camera panning and MultiStack.TurnManager updates, and saves current height and round via MultiStack.DataPersistence
- void OnRoundOver ()

  Raised at the end of each round. Shows new round text, applies a round modifier via MultiStack.GameModifierManager and invokes MultiStack.TurnManager.StartNewRound.

Public Attributes

- UIFlash uiFlash

  Reference to UIFlash, used to hide the game behind a semi-transparent texture on game over.
- GameOverUI gameOverUI

  The game over UI controller.
- StageSelector stageSelector

  The stage selector. Used to spawn the stage at the beginning of the game.
- GameModifierManager gameModifierManager

  The game modifier manager. Used to apply a game modifier at the beginning of each round.
- GameText infoText

  The info text UI object. Provides information to the player.

Properties

- bool isGameOver `[get]`

  Gets a value indicating whether the game is over.
- static GameManager instance `[get]`

  Gets the instance of this class. Accessible from any class.

5.17.1 Detailed Description

Controls the game flow. Handles beginning the game, spawning the stage, starting new rounds, and ending the game.

5.17.2 Member Function Documentation

5.17.2.1 void MultiStack.GameManager.OnGameOver (GameOverType *gameOverType* = `GameOverType.ShapeOffScreen`)

Called in the event of a game over (either a players timer reaches zero or a shape falls out of bounds). Disables camera panning and MultiStack.TurnManager updates, and saves current height and round via MultiStack.DataPersistence

5.17.2.2 void MultiStack.GameManager.OnRoundOver ( )

Raised at the end of each round. Shows new round text, applies a round modifier via MultiStack.GameModifierManager and invokes MultiStack.TurnManager.StartNewRound.

5.17.3 Member Data Documentation

5.17.3.1 GameModifierManager MultiStack.GameManager.gameModifierManager

The game modifier manager. Used to apply a game modifier at the beginning of each round.

5.17.3.2 GameOverUI MultiStack.GameManager.gameOverUI

The game over UI controller.

5.17.3.3 GameText MultiStack.GameManager.infoText

The info text UI object. Provides information to the player.

5.17.3.4 StageSelector MultiStack.GameManager.stageSelector

The stage selector. Used to spawn the stage at the beginning of the game.

5.17.3.5 UIFlash MultiStack.GameManager.uiFlash

Reference to UIFlash, used to hide the game behind a semi-transparent texture on game over.

5.17.4 Property Documentation

5.17.4.1 GameManager MultiStack.GameManager.instance `[static], [get]`

Gets the instance of this class. Accessible from any class.

The instance.

5.17.4.2 bool MultiStack.GameManager.isGameOver `[get]`

Gets a value indicating whether the game is over.

`true` if is game over; otherwise, `false`.

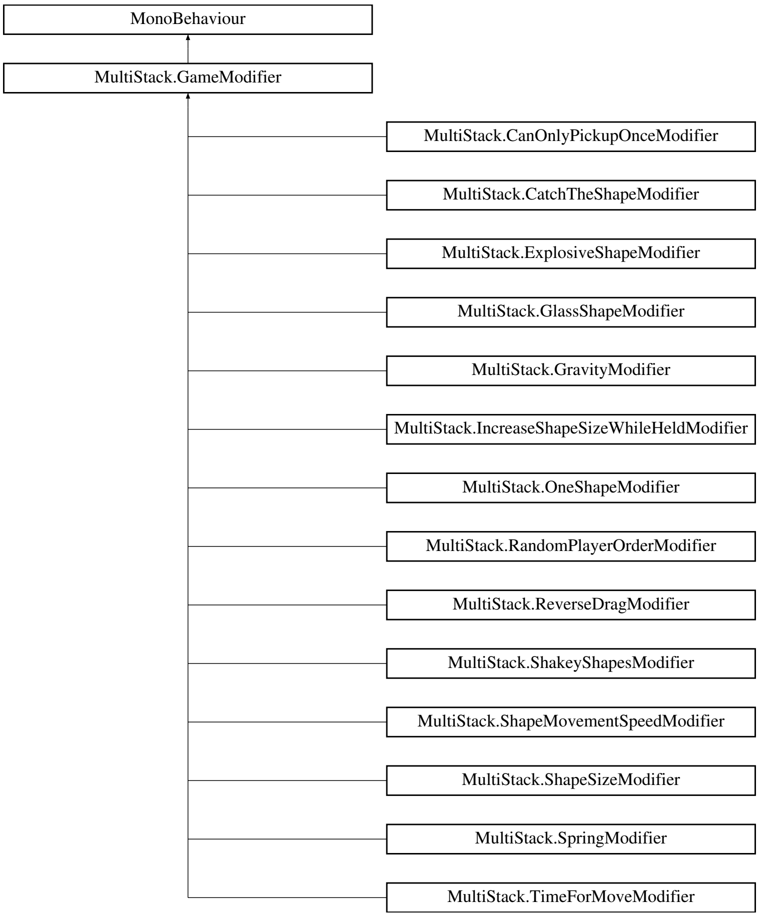
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Managers/GameManager.cs

5.18 MultiStack.GameModifier Class Reference

The abstract base class for every game modifier. Game modifiers are applied each round to change how the game is played.

Inheritance diagram for MultiStack.GameModifier:


MonoBehaviour
MultiStack.GameModifier
MultiStack.CanOnlyPickupOnceModifier
MultiStack.CatchTheShapeModifier
MultiStack.ExplosiveShapeModifier
MultiStack.GlassShapeModifier
MultiStack.GravityModifier
MultiStack.IncreaseShapeSizeWhileHeldModifier
MultiStack.OneShapeModifier
MultiStack.RandomPlayerOrderModifier
MultiStack.ReverseDragModifier
MultiStack.ShakeyShapesModifier
MultiStack.ShapeMovementSpeedModifier
MultiStack.ShapeSizeModifier
MultiStack.SpringModifier
MultiStack.TimeForMoveModifier

Public Member Functions

- abstract void Activate ()

  Activate this modifier. Called at the beginning of the round.
- abstract void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.
- virtual bool ConditionsMet ()

  Place any prerequisites for the modifier here. By defauly a modifier has all conditions met unless this method is overriden.

Public Attributes

- bool IsEnabled = true

  Set whether this game modifier is enabled. Disable this in the inspector to prevent the modifier from being applied.
- string modifierName

  The modifier name. This is shown in the UI when the modifier is applied.

5.18.1 Detailed Description

The abstract base class for every game modifier. Game modifiers are applied each round to change how the game is played.

5.18.2 Member Function Documentation

5.18.2.1 abstract void MultiStack.GameModifier.Activate ( ) [pure virtual]

Activate this modifier. Called at the beginning of the round.

Implemented in MultiStack.OneShapeModifier, MultiStack.TimeForMoveModifier, MultiStack.GravityModifier, MultiStack.ShapeMovementSpeedModifier, MultiStack.CanOnlyPickupOnceModifier, MultiStack.ExplosiveShapeModifier, MultiStack.IncreaseShapeSizeWhileHeldModifier, MultiStack.ReverseDragModifier, MultiStack.ShakeyShapesModifier, MultiStack.ShapeSizeModifier, MultiStack.SpringModifier, MultiStack.GlassShapeModifier, MultiStack.CatchTheShapeModifier, and MultiStack.RandomPlayerOrderModifier.

5.18.2.2 virtual bool MultiStack.GameModifier.ConditionsMet ( ) [virtual]

Place any prerequisites for the modifier here. By defauly a modifier has all conditions met unless this method is overriden.

Returns

`true`, if met was conditionsed, `false` otherwise.

Reimplemented in MultiStack.ExplosiveShapeModifier, and MultiStack.GlassShapeModifier.

5.18.2.3 abstract void MultiStack.GameModifier.Deactivate ( ) [pure virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implemented in MultiStack.OneShapeModifier, MultiStack.TimeForMoveModifier, MultiStack.GravityModifier, MultiStack.ShapeMovementSpeedModifier, MultiStack.CanOnlyPickupOnceModifier, MultiStack.ExplosiveShapeModifier, MultiStack.IncreaseShapeSizeWhileHeldModifier, MultiStack.ReverseDragModifier, MultiStack.ShakeyShapesModifier, MultiStack.ShapeSizeModifier, MultiStack.SpringModifier, MultiStack.GlassShapeModifier, MultiStack.CatchTheShapeModifier, and MultiStack.RandomPlayerOrderModifier.

5.18.3 Member Data Documentation

5.18.3.1 bool MultiStack.GameModifier.IsEnabled = true

Set whether this game modifier is enabled. Disable this in the inspector to prevent the modifier from being applied.

5.18.3.2 string MultiStack.GameModifier.modifierName

The modifier name. This is shown in the UI when the modifier is applied.

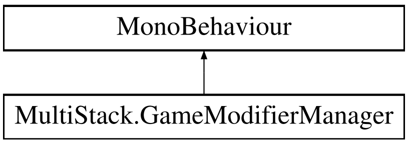
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/GameModifier.cs

5.19 MultiStack.GameModifierManager Class Reference

Handles game modifiers. These are modififers applied at the beginning of each round (e.g. low gravity).

Inheritance diagram for MultiStack.GameModifierManager:



Public Member Functions

- string ApplyNewModifier ()
  Applies a new modifier if there are any valid and enabled modifiers.

5.19.1 Detailed Description

Handles game modifiers. These are modififers applied at the beginning of each round (e.g. low gravity).

5.19.2 Member Function Documentation

5.19.2.1 string MultiStack.GameModifierManager.ApplyNewModifier ( )

Applies a new modifier if there are any valid and enabled modifiers.

Returns

The new modifiers name, used to update the UI.

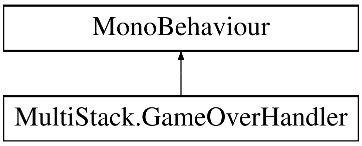
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Managers/GameModifierManager.cs

5.20 MultiStack.GameOverHandler Class Reference

Invokes MultiStack.GameManager.OnGameOver when a physics object enters trigger.

Inheritance diagram for MultiStack.GameOverHandler:



5.20.1 Detailed Description

Invokes MultiStack.GameManager.OnGameOver when a physics object enters trigger.

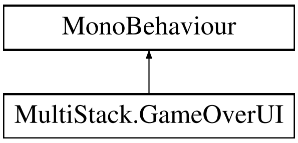
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Managers/GameOverHandler.cs

5.21 MultiStack.GameOverUI Class Reference

Handles the displaying of the UI in the event of a game over.

Inheritance diagram for MultiStack.GameOverUI:



Public Member Functions

- void Activate ()

  Activate this instance. Shows the game over UI.
- void RestartScene ()

  Reloads the current scene.
- void ReturnToMainMenu ()

  Returns to main menu scene.

Public Attributes

- GameObject title

  The gameobject for the title UI.
- GameObject newRecord

  The gameobject that holds the new record UI elements.
- GameText[ ] nonGameoverUIText

  Add any UI not related to the game over UI that you would like to hide.
- GameObject playerLostTitle

  The title UI for player lost.
- GameText playerLostText

  The player lost text. Holds the number of the player that has lost the game.
- GameObject heightReachedTitle

  The UI title for the hieght field.
- GameText heightReachedText

  The hieght reached text. Holds the height reached the current game.
- GameObject roundsReachedTitle

  The rounds reached title.
- GameText roundsReachedText

  The rounds reached text. Holds the round reached in the current game.
- GameObject restartButton

  The restart button.
- GameObject mainMenuButton

  The main menu button.

5.21.1 Detailed Description

Handles the displaying of the UI in the event of a game over.

5.21.2 Member Function Documentation

5.21.2.1 void MultiStack.GameOverUI.Activate ( )

Activate this instance. Shows the game over UI.

5.21.2.2 void MultiStack.GameOverUI.RestartScene ( )

Reloads the current scene.

5.21.2.3 void MultiStack.GameOverUI.ReturnToMainMenu ( )

Returns to main menu scene.

5.21.3 Member Data Documentation

5.21.3.1 GameText MultiStack.GameOverUI.heightReachedText

The hieght reached text. Holds the height reached the current game.

5.21.3.2 GameObject MultiStack.GameOverUI.heightReachedTitle

The UI title for the hieght field.

5.21.3.3 GameObject MultiStack.GameOverUI.mainMenuButton

The main menu button.

5.21.3.4 GameObject MultiStack.GameOverUI.newRecord

The gameobject that holds the new record UI elements.

5.21.3.5 GameText [ ] MultiStack.GameOverUI.nonGameoverUIText

Add any UI not related to the game over UI that you would like to hide.

5.21.3.6 GameText MultiStack.GameOverUI.playerLostText

The player lost text. Holds the number of the player that has lost the game.

5.21.3.7 GameObject MultiStack.GameOverUI.playerLostTitle

The title UI for player lost.

5.21.3.8 GameObject MultiStack.GameOverUI.restartButton

The restart button.

5.21.3.9 GameText MultiStack.GameOverUI.roundsReachedText

The rounds reached text. Holds the round reached in the current game.

5.21.3.10 GameObject MultiStack.GameOverUI.roundsReachedTitle

The rounds reached title.

5.21.3.11 GameObject MultiStack.GameOverUI.title

The gameobject for the title UI.

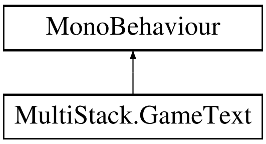
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/GameOverUI.cs

5.22 MultiStack.GameText Class Reference

A simple wrapper for the text object.

Inheritance diagram for MultiStack.GameText:



Public Member Functions

- void UpdateText (string msg, Color colour)

  Updates the UI text with the desired font colour.
- void UpdateText (string msg)

  Updates the UI text.
- void HideTextAfterTime (float seconds)

  Hides the text after time in seconds.
- void HideText ()

  Clears the UI text field.
- void ChangeFontColour (Color colour)

  Changes the font colour.

Public Attributes

- bool clearOnStart = true

  CLears all text on scene start.

Properties

- string text [get]

  Gets the text value.

5.22.1 Detailed Description

A simple wrapper for the text object.

5.22.2 Member Function Documentation

5.22.2.1 void MultiStack.GameText.ChangeFontColour (Color *colour*)

Changes the font colour.

Parameters

colour	Colour.

5.22.2.2 void MultiStack.GameText.HideText ( )

Clears the UI text field.

5.22.2.3 void MultiStack.GameText.HideTextAfterTime (float *seconds*)

Hides the text after time in seconds.

Parameters

seconds	Seconds.

5.22.2.4 void MultiStack.GameText.UpdateText (string *msg,* Color *colour*)

Updates the UI text with the desired font colour.

Parameters

msg	Message.
colour	Colour.

5.22.2.5 void MultiStack.GameText.UpdateText (string *msg*)

Updates the UI text.

Parameters

msg	Message.

5.22.3 Member Data Documentation

5.22.3.1 bool MultiStack.GameText.clearOnStart = true

CLears all text on scene start.

5.22.4 Property Documentation

5.22.4.1 string MultiStack.GameText.text [get]

Gets the text value.

The text.

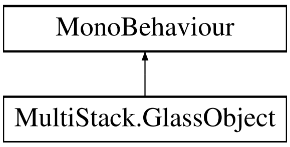
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/GameText.cs

5.23 MultiStack.GlassObject Class Reference

Attached to any shape that can be turned into a glass shape. Handles changing a shapes sprite and breaking.

Inheritance diagram for MultiStack.GlassObject:



Public Member Functions

- void Activate ()

  Activate this instance and changes the shapes sprite to that of MultiStack.GlassObject.initialGlassSprite.

Public Attributes

- Sprite initialGlassSprite

  The shapes sprite is changed to this sprite when it is activated.
- Sprite[ ] brokenStages

  The broken stages. A glass shape can go through a number of stages before breaking. This holds the sprite for each stage.
- GameObject[ ] smashedObjectPrefabs

  The smashed object prefabs. Spawned when the glass object breaks.
- AudioClip crackedAudioClip

  The audioclip to play when the glass shape cracks.
- AudioClip smashedAudioClip

  The audio clip to play when the glass shape breaks.
- float requiredForceToCrack = 2f

  The required force to crack. When an object collides with this shape, if the collision force is above this force then the shape cracks/breaks.
- float requiredMassToCrack = 2f

  The required combined mass of shapes placed on top of this shape to cause the shape to break/crack.
- LayerMask boxMask

  The layermask for physics objects.

Properties

- bool readyToBeActivated [get]

  Gets a value indicating whether this MultiStack.GlassObject is ready to be activated.

5.23.1 Detailed Description

Attached to any shape that can be turned into a glass shape. Handles changing a shapes sprite and breaking.

5.23.2 Member Function Documentation

5.23.2.1 void MultiStack.GlassObject.Activate ( )

Activate this instance and changes the shapes sprite to that of MultiStack.GlassObject.initialGlassSprite.

5.23.3 Member Data Documentation

5.23.3.1 LayerMask MultiStack.GlassObject.boxMask

The layermask for physics objects.

5.23.3.2 Sprite [ ] MultiStack.GlassObject.brokenStages

The broken stages. A glass shape can go through a number of stages before breaking. This holds the sprite for each stage.

5.23.3.3 AudioClip MultiStack.GlassObject.crackedAudioClip

The audioclip to play when the glass shape cracks.

5.23.3.4 Sprite MultiStack.GlassObject.initialGlassSprite

The shapes sprite is changed to this sprite when it is activated.

5.23.3.5 float MultiStack.GlassObject.requiredForceToCrack = 2f

The required force to crack. When an object collides with this shape, if the collision force is above this force then the shape cracks/breaks.

5.23.3.6 float MultiStack.GlassObject.requiredMassToCrack = 2f

The required combined mass of shapes placed on top of this shape to cause the shape to break/crack.

5.23.3.7 AudioClip MultiStack.GlassObject.smashedAudioClip

The audio clip to play when the glass shape breaks.

5.23.3.8 GameObject [ ] MultiStack.GlassObject.smashedObjectPrefabs

The smashed object prefabs. Spawned when the glass object breaks.

5.23.4 Property Documentation

5.23.4.1 bool MultiStack.GlassObject.readyToBeActivated [get]

Gets a value indicating whether this MultiStack.GlassObject is ready to be activated.

`true` if ready to be activated; otherwise, `false`.

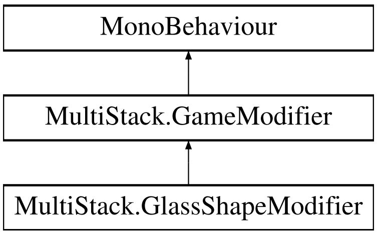
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Objects/GlassObject.cs

5.24 MultiStack.GlassShapeModifier Class Reference

When enabled and conditions met a random shape is turned into an glass shape. Invokes MultiStack.TurnManager.ChangeShapeToGlass.

Inheritance diagram for MultiStack.GlassShapeModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.
- override bool ConditionsMet ()

  Returns true if any spawned shape can be turned into an glass shape.

Additional Inherited Members

5.24.1 Detailed Description

When enabled and conditions met a random shape is turned into an glass shape. Invokes MultiStack.TurnManager.ChangeShapeToGlass.

5.24.2 Member Function Documentation

5.24.2.1 override void MultiStack.GlassShapeModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.24.2.2 override bool MultiStack.GlassShapeModifier.ConditionsMet ( ) [virtual]

Returns true if any spawned shape can be turned into an glass shape.

Returns

true if a shape can be turned into glass shape.

`false`

Reimplemented from MultiStack.GameModifier.

5.24.2.3 override void MultiStack.GlassShapeModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

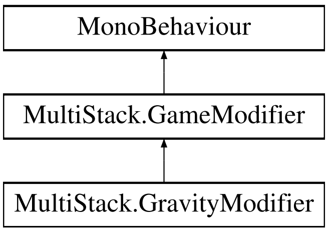
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/GlassShapeModifier.cs

5.25 MultiStack.GravityModifier Class Reference

When enabled a gravity modifier is appled to all spawned shapes.

Inheritance diagram for MultiStack.GravityModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- float gravityModifier

  Every spawned objects gravity will be multiplied by this modifier.
- TurnManager **turnManager**

5.25.1 Detailed Description

When enabled a gravity modifier is appled to all spawned shapes.

5.25.2 Member Function Documentation

5.25.2.1 override void MultiStack.GravityModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.25.2.2 override void MultiStack.GravityModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

5.25.3 Member Data Documentation

5.25.3.1 float MultiStack.GravityModifier.gravityModifier

Every spawned objects gravity will be multiplied by this modifier.

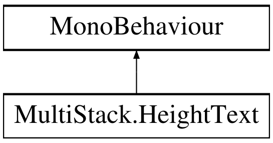
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/GravityModifier.cs

5.26 MultiStack.HeightText Class Reference

The UI for the height shown during the game scene.

Inheritance diagram for MultiStack.HeightText:



5.26.1 Detailed Description

The UI for the height shown during the game scene.

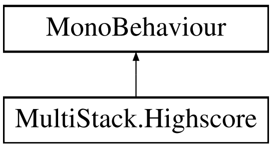
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/HeightText.cs

5.27 MultiStack.Highscore Class Reference

The UI for the highest height reached on the Main menu scene.

Inheritance diagram for MultiStack.Highscore:



5.27.1 Detailed Description

The UI for the highest height reached on the Main menu scene.

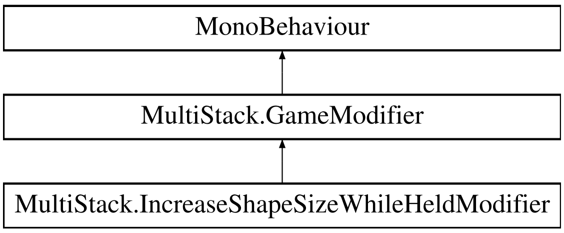
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/Highscore.cs

5.28 MultiStack.IncreaseShapeSizeWhileHeldModifier Class Reference

When enabled a shapes size increases while being dragged.

Inheritance diagram for MultiStack.IncreaseShapeSizeWhileHeldModifier:



Public Member Functions

- override void Activate ()

    Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

    Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- ClickHandler **clickHandler**

5.28.1 Detailed Description

When enabled a shapes size increases while being dragged.

5.28.2 Member Function Documentation

5.28.2.1 override void MultiStack.IncreaseShapeSizeWhileHeldModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.28.2.2 override void MultiStack.IncreaseShapeSizeWhileHeldModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

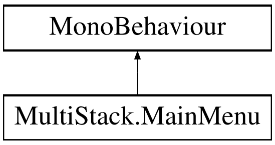
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/IncreaseShapeSizeWhileHeldModifier.cs

5.29 MultiStack.MainMenu Class Reference

Handles the UI for main menu and number of players select scene.

Inheritance diagram for MultiStack.MainMenu:



Public Member Functions

- void PlayerBoxPlaced ()

  Called when the player places a box on the platform.
- void PlayerBoxRemoved ()

  Called when the player removes a box from the platform.
- void PlayButtonPressed ()

  Called when the play button has been pressed. If it is the first time it has been pressed, then it pans the camera down and then spawns the boxes else it loads the game scene.

Public Attributes

- GameText titleText

  The title text.
- GameText numOfPlayersText

  The number of players text.
- GameText otherText

  The other text. This is used to show messages to the player.
- RectTransform playButton

  The play button rect transform. Used to translate the button when moving to the number of players screen.
- GameObject playerBoxPrefab

  The prefab for the physics boxes used in the number of players select screen.
- Transform[ ] playerBoxSpawnLocations

  The box spawn locations.

5.29.1 Detailed Description

Handles the UI for main menu and number of players select scene.

5.29.2 Member Function Documentation

5.29.2.1 void MultiStack.MainMenu.PlayButtonPressed ( )

Called when the play button has been pressed. If it is the first time it has been pressed, then it pans the camera down and then spawns the boxes else it loads the game scene.

5.29.2.2 void MultiStack.MainMenu.PlayerBoxPlaced ( )

Called when the player places a box on the platform.

5.29.2.3 void MultiStack.MainMenu.PlayerBoxRemoved ( )

Called when the player removes a box from the platform.

5.29.3 Member Data Documentation

5.29.3.1 GameText MultiStack.MainMenu.numOfPlayersText

The number of players text.

5.29.3.2 GameText MultiStack.MainMenu.otherText

The other text. This is used to show messages to the player.

5.29.3.3 RectTransform MultiStack.MainMenu.playButton

The play button rect transform. Used to translate the button when moving to the number of players screen.

5.29.3.4 GameObject MultiStack.MainMenu.playerBoxPrefab

The prefab for the physics boxes used in the number of players select screen.

5.29.3.5 Transform [ ] MultiStack.MainMenu.playerBoxSpawnLocations

The box spawn locations.

5.29.3.6 GameText MultiStack.MainMenu.titleText

The title text.

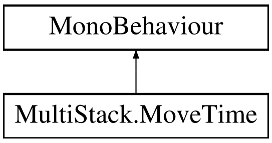
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/MainMenu.cs

5.30 MultiStack.MoveTime Class Reference

Responsible for updating the current players move time.

Inheritance diagram for MultiStack.MoveTime:



Public Attributes

- int lowOnTimeThreshold

  The low on time threshold. When the players current time is below this threshold the time will pulse and move to the centre of the screen,
- Color normalTimeTextColour

  The colour of the countdown text when the time is above the MultiStack.MoveTime.lowOnTimeThreshold.
- Color lowTimeTextColour

  The colour of the countdown text when the time is below the MultiStack.MoveTime.lowOnTimeThreshold.

5.30.1 Detailed Description

Responsible for updating the current players move time.

5.30.2 Member Data Documentation

5.30.2.1 int MultiStack.MoveTime.lowOnTimeThreshold

The low on time threshold. When the players current time is below this threshold the time will pulse and move to the centre of the screen,

5.30.2.2 Color MultiStack.MoveTime.lowTimeTextColour

The colour of the countdown text when the time is below the MultiStack.MoveTime.lowOnTimeThreshold.

5.30.2.3 Color MultiStack.MoveTime.normalTimeTextColour

The colour of the countdown text when the time is above the MultiStack.MoveTime.lowOnTimeThreshold.

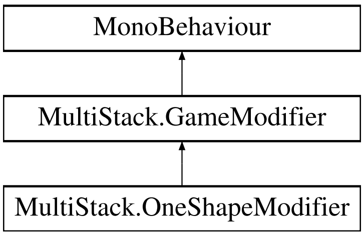
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/MoveTime.cs

5.31 MultiStack.OneShapeModifier Class Reference

When enabled only one shape specified by MultiStack.OneShapeModifier.shape will be spawned for that round.

Inheritance diagram for MultiStack.OneShapeModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- Shape shape

  The sole shape to be spawned this round.

5.31.1 Detailed Description

When enabled only one shape specified by MultiStack.OneShapeModifier.shape will be spawned for that round.

5.31.2 Member Function Documentation

5.31.2.1 override void MultiStack.OneShapeModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.31.2.2 override void MultiStack.OneShapeModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

5.31.3 Member Data Documentation

5.31.3.1 Shape MultiStack.OneShapeModifier.shape

The sole shape to be spawned this round.

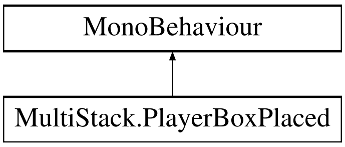
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/OneShapeModifier.cs

5.32 MultiStack.PlayerBoxPlaced Class Reference

Used during the number of player selection scene. Handles alerting MultiStack.MainMenu when a box is placed by invoking MultiStack.MainMenu.PlayerBoxPlaced and MultiStack.MainMenu.PlayerBoxRemoved.

Inheritance diagram for MultiStack.PlayerBoxPlaced:



5.32.1 Detailed Description

Used during the number of player selection scene. Handles alerting MultiStack.MainMenu when a box is placed by invoking MultiStack.MainMenu.PlayerBoxPlaced and MultiStack.MainMenu.PlayerBoxRemoved.

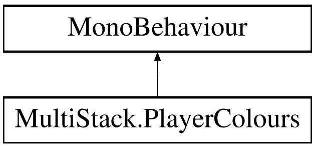
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/PlayerBoxPlaced.cs

5.33 MultiStack.PlayerColours Class Reference

Holds the coours for each player.

Inheritance diagram for MultiStack.PlayerColours:



Public Attributes

- Color[ ] colours

  The colours for each player.

5.33.1 Detailed Description

Holds the coours for each player.

5.33.2 Member Data Documentation

5.33.2.1 Color [ ] MultiStack.PlayerColours.colours

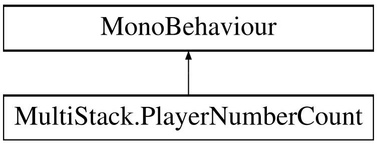
The colours for each player.

The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/PlayerColours.cs

5.34 MultiStack.PlayerNumberCount Class Reference

Inheritance diagram for MultiStack.PlayerNumberCount:



Properties

- int numberOfPlayers `[get, set]`

  Gets or sets the number of players. Used to store the number of players between scenes.

5.34.1 Property Documentation

5.34.1.1 int MultiStack.PlayerNumberCount.numberOfPlayers `[get], [set]`

Gets or sets the number of players. Used to store the number of players between scenes.

The number of players.

The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/PlayerNumberCount.cs

5.35 MultiStack.PlayerPhysicsObject Struct Reference

A structure defining a physics object.

Public Attributes

- Shape shape

  The type of shape this object represents.
- GameObject simplePrefab

  A simple prefab without any of the associated scripts. When the shapes are being looped through and displayed at the beginning of a players turn the simple prefabs are used.
- GameObject realPrefab

  The real prefab. This is the prefab that is spawned once a shape has been selected.
- float weight

  The chance that this object will spawn. The weight is proportional to other shapes weights.

Properties

- GameObject instantiatedPrefab `[get, set]`

  Gets or sets the instantiated simple prefab.

5.35.1 Detailed Description

A structure defining a physics object.

5.35.2 Member Data Documentation

5.35.2.1 GameObject MultiStack.PlayerPhysicsObject.realPrefab

The real prefab. This is the prefab that is spawned once a shape has been selected.

5.35.2.2 Shape MultiStack.PlayerPhysicsObject.shape

The type of shape this object represents.

5.35.2.3 GameObject MultiStack.PlayerPhysicsObject.simplePrefab

A simple prefab without any of the associated scripts. When the shapes are being looped through and displayed at the beginning of a players turn the simple prefabs are used.

5.35.2.4 float MultiStack.PlayerPhysicsObject.weight

The chance that this object will spawn. The weight is proportional to other shapes weights.

5.35.3 Property Documentation

5.35.3.1 GameObject MultiStack.PlayerPhysicsObject.instantiatedPrefab [get], [set]

Gets or sets the instantiated simple prefab.

The instantiated prefab.

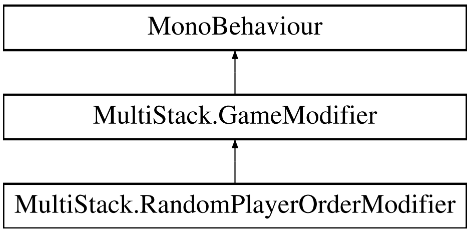
The documentation for this struct was generated from the following file:

- MultiplayerStacker/Scripts/PlayerPhysicsObject.cs

5.36 MultiStack.RandomPlayerOrderModifier Class Reference

When enabled the order of players is randomised for this round.

Inheritance diagram for MultiStack.RandomPlayerOrderModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Additional Inherited Members

5.36.1 Detailed Description

When enabled the order of players is randomised for this round.

5.36.2 Member Function Documentation

5.36.2.1 override void MultiStack.RandomPlayerOrderModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.36.2.2 override void MultiStack.RandomPlayerOrderModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

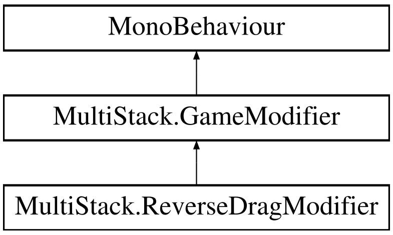
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/RandomPlayerOrderModifier.cs

5.37 MultiStack.ReverseDragModifier Class Reference

When enabled drag is reversed e.g. when the player drags the mouse/finger left the shape moves right.

Inheritance diagram for MultiStack.ReverseDragModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- ClickHandler **clickHandler**

5.37.1 Detailed Description

When enabled drag is reversed e.g. when the player drags the mouse/finger left the shape moves right.

5.37.2 Member Function Documentation

5.37.2.1 override void MultiStack.ReverseDragModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.37.2.2 override void MultiStack.ReverseDragModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/ReverseDragModifier.cs

5.38 MultiStack.ScoreData Class Reference

Class to store serializable height/round data.

Public Member Functions

- **ScoreData** (int height, int round)

Properties

- int **height** [get]
- int **round** [get]

5.38.1 Detailed Description

Class to store serializable height/round data.

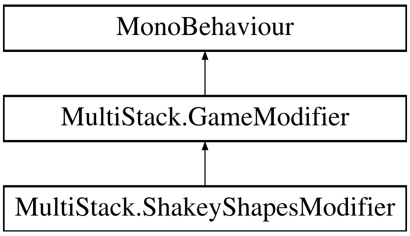
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/High Score/DataPersistence.cs

5.39 MultiStack.ShakeyShapesModifier Class Reference

When enabled a random offset is added to a dragged shape each timestep.

Inheritance diagram for MultiStack.ShakeyShapesModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- ClickHandler **clickHandler**

5.39.1 Detailed Description

When enabled a random offset is added to a dragged shape each timestep.

5.39.2 Member Function Documentation

5.39.2.1 override void MultiStack.ShakeyShapesModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.39.2.2 override void MultiStack.ShakeyShapesModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

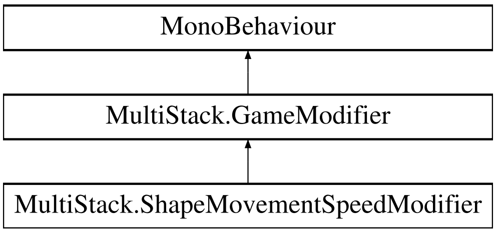
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/ShakeyShapesModifier.cs

5.40 MultiStack.ShapeMovementSpeedModifier Class Reference

When enabled the drag speed is changed to MultiStack.ShapeMovementSpeedModifier.shapeMovementSpeed.

Inheritance diagram for MultiStack.ShapeMovementSpeedModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- ClickHandler **clickHandler**
- float **shapeMovementSpeed**

5.40.1 Detailed Description

When enabled the drag speed is changed to MultiStack.ShapeMovementSpeedModifier.shapeMovementSpeed.

5.40.2 Member Function Documentation

5.40.2.1 override void MultiStack.ShapeMovementSpeedModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.40.2.2 override void MultiStack.ShapeMovementSpeedModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

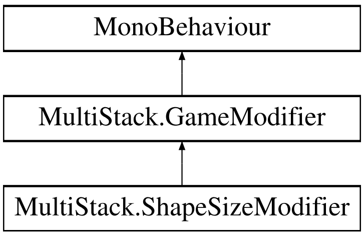
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/ShapeMovementSpeedModifier.cs

5.41 MultiStack.ShapeSizeModifier Class Reference

When enabled all newly spawned shapes size is multiplied by MultiStack.ShapeSizeModifier.sizeModifier.

Inheritance diagram for MultiStack.ShapeSizeModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- float **sizeModifier**

5.41.1 Detailed Description

When enabled all newly spawned shapes size is multiplied by MultiStack.ShapeSizeModifier.sizeModifier.

5.41.2 Member Function Documentation

5.41.2.1 override void MultiStack.ShapeSizeModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.41.2.2 override void MultiStack.ShapeSizeModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

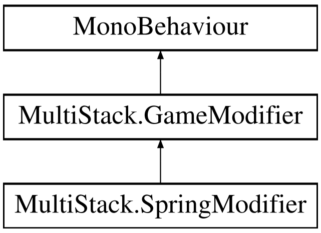
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/ShapeSizeModifier.cs

5.42 MultiStack.SpringModifier Class Reference

When enabled the shapes are attached to springs when being dragged. This results in a harder to control shape.

Inheritance diagram for MultiStack.SpringModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- ClickHandler **clickHandler**

5.42.1 Detailed Description

When enabled the shapes are attached to springs when being dragged. This results in a harder to control shape.

5.42.2 Member Function Documentation

5.42.2.1 override void MultiStack.SpringModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.42.2.2 override void MultiStack.SpringModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

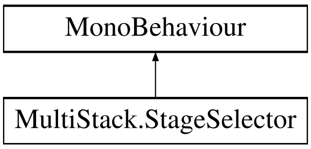
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/SpringModifier.cs

5.43 MultiStack.StageSelector Class Reference

Used to select a stage for the game. A random stage is selected at the beginning of each game.

Inheritance diagram for MultiStack.StageSelector:



Public Member Functions

- void CallBackOnStageComplete (CallBack callBack)

  Injects a method to be invoked when the stage spawn has completed.
- void SpawnStage ()

  Begins the stage spawning process. Updates the UI with MultiStack.StageSelector.stageText, loops through the stages and spawns the final stage. Invokes callback method set by MultiStack.StageSelector.CallBackOnStageComplete if present.

Public Attributes

- GameText infoText

  Used to display a message at the beginning of the stage spawn process.
- string stageText = "What stage will it be this time?"

  The message shown when before stage spawning begins.
- GameObject[ ] stagePrefabs

  The stage prefabs.

Properties

- bool stageSpawned [get]

  Gets a value indicating whether this MultiStack.StageSelector has spawned a stage.

5.43.1 Detailed Description

Used to select a stage for the game. A random stage is selected at the beginning of each game.

5.43.2 Member Function Documentation

5.43.2.1 void MultiStack.StageSelector.CallBackOnStageComplete (CallBack *callBack*)

Injects a method to be invoked when the stage spawn has completed.

Parameters

callBack	Call back.

5.43.2.2 void MultiStack.StageSelector.SpawnStage ( )

Begins the stage spawning process. Updates the UI with MultiStack.StageSelector.stageText, loops through the stages and spawns the final stage. Invokes callback method set by MultiStack.StageSelector.CallBackOnStage↩ Complete if present.

5.43.3 Member Data Documentation

5.43.3.1 GameText MultiStack.StageSelector.infoText

Used to display a message at the beginning of the stage spawn process.

5.43.3.2 GameObject [ ] MultiStack.StageSelector.stagePrefabs

The stage prefabs.

5.43.3.3 string MultiStack.StageSelector.stageText = "What stage will it be this time?"

The message shown when before stage spawning begins.

5.43.4 Property Documentation

5.43.4.1 bool MultiStack.StageSelector.stageSpawned [get]

Gets a value indicating whether this MultiStack.StageSelector has spawned a stage.

`true` if stage spawned; otherwise, `false`.

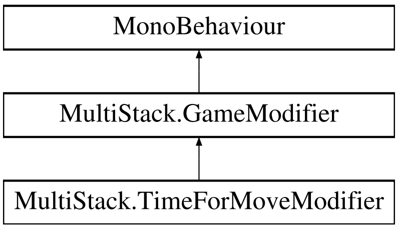
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Managers/StageSelector.cs

5.44 MultiStack.TimeForMoveModifier Class Reference

When enabled the time a player has to complete their turn is changed to MultiStack.TimeForMoveModifier.↩ changedTime.

Inheritance diagram for MultiStack.TimeForMoveModifier:



Public Member Functions

- override void Activate ()

  Activate this modifier. Called at the beginning of the round.
- override void Deactivate ()

  Deactivate this modifier. Called before a new modifier is enabled.

Public Attributes

- int changedTime = 15

  The new move time for this round.

5.44.1 Detailed Description

When enabled the time a player has to complete their turn is changed to MultiStack.TimeForMoveModifier.↩ changedTime.

5.44.2 Member Function Documentation

5.44.2.1 override void MultiStack.TimeForMoveModifier.Activate ( ) [virtual]

Activate this modifier. Called at the beginning of the round.

Implements MultiStack.GameModifier.

5.44.2.2 override void MultiStack.TimeForMoveModifier.Deactivate ( ) [virtual]

Deactivate this modifier. Called before a new modifier is enabled.

Implements MultiStack.GameModifier.

5.44.3 Member Data Documentation

5.44.3.1 int MultiStack.TimeForMoveModifier.changedTime = 15

The new move time for this round.

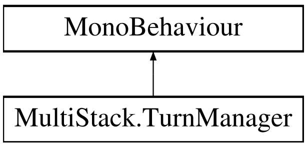
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Modifiers/TimeForMoveModifier.cs

5.45 MultiStack.TurnManager Class Reference

Handles player turns an spawning of player physics objects objects. Manages and maintains a list of spawned physics objects. Also handles applying certain MultiStack.GameModifier e.g. when MultiStack.ExplosiveShape↩ Modifier is applied, this class will loop through the spawned shape list and convert one shape into an explosive shape.

Inheritance diagram for MultiStack.TurnManager:



Public Member Functions

- void Initialise (int numOfPlayers)

  Sets number of players. Loads MultiStack.TurnManager.physicsObjects.
- void ChangeShapeToGlass ()

  Changes a shape to glass. The most recent shape that can be turn into glass is selected.
- void ChangeShapeToExplosive ()

  Changes a shape to explosive. The most recent shape that can be turned into an explosive is selected.
- void ObjectPickedUp (Rigidbody2D obj)

  Specifies that an object has been picked up. This is used to help decide if the current player has finished their turn.
- void ObjectDropped (Rigidbody2D obj)

  Specifies that an object has been picked dropped. This is used to help decide if the current player has finished their turn.
- void StartNewRound ()

  Starts a new round, increments current round counter and begins a new turn if not gameover.
- void ApplyGravityModifier (float modifier)

  Loops through each spawned physics object and multiplies the gravvity scale by the modifier.
- void DisableGravityModifier ()

  Disables the gravity modifier. Resets each spawned physics object gravity scale.
- void ApplySizeModifier (float modifier)

  Applies a size modifier to each spawned physics object for this round.
- void DisableSizeModifier ()

  Disables the size modifier.
- void ApplyShapeModifier (Shape shape)

  When invoked only the specified shape will be spawned this round.
- void DisableShapeModifier ()

  Disables the shape modifier. Any type of shape will be spawned next round.
- void ApplyCatchTheShapeModifier ()

  Applies the catch the shape modifier. When this is activated, when a shape is spawned it is dropped and the player has to catch it before it falls.
- void DisableCatchTheShapeModifier ()

  Disables the catch the shape modifier.

Public Attributes

- Transform[ ] playerLocations

  The player locations where the plays shape will spawn.
- PlayerPhysicsObject[ ] physicsObjects

  The physics objects that can be spawned.
- GameText playerText

  The UI that will inform the player it is their turn.
- GameText timeRemainingText

  The UI that shows the current players time remaining for their turn.
- PlayerColours playerColours

  The player colours. The MultiStack.TurnManager.playerText UI is shown in the current players colour.
- int baseTimeForMove = 20

  The base tim (without modifiers) that each player has to complete their turn.

Properties

- int currentRound `[get]`

  The current round number.
- int currentEnhancedHeight `[get]`

  Gets the height of the current stack of physics objects + 4f. This is used for the UI.
- float shapeOffsetHeight `[get]`

  Gets the height of the shape stack offset. Used by the MultiStack.CameraManager to limit how far the camera can scroll upwards.
- int numOfObjectsSpawned `[get]`

  The number of physics objects in the current scene.
- bool explosiveSpriteAvailableForSpawnedObject `[get]`

  Gets a value indicating whether this MultiStack.TurnManager has spawned a physics object that can be turned into an explosive object.
- bool glassSpriteAvailableForSpawnedObject `[get]`

  Gets a value indicating whether this MultiStack.TurnManager has spawned an object that can be turned into a glass object.
- bool randomPlayerOrder `[set]`

  Sets a value indicating whether this MultiStack.TurnManager should shuffle the players for the next round.
- int currentPlayer `[get]`

  The index of the current player.
- bool shouldUpdate `[set]`

  Sets a value indicating whether this MultiStack.TurnManager should update.
- static TurnManager instance `[get]`

  Signleton pattern. Gets the instance. Accessible from any class.

5.45.1 Detailed Description

Handles player turns an spawning of player physics objects objects. Manages and maintains a list of spawned physics objects. Also handles applying certain MultiStack.GameModifier e.g. when MultiStack.ExplosiveShapeModifier is applied, this class will loop through the spawned shape list and convert one shape into an explosive shape.

5.45.2 Member Function Documentation

5.45.2.1 void MultiStack.TurnManager.ApplyCatchTheShapeModifier ( )

Applies the catch the shape modifier. When this is activated, when a shape is spawned it is dropped and the player has to catch it before it falls.

5.45.2.2 void MultiStack.TurnManager.ApplyGravityModifier (float *modifier*)

Loops through each spawned physics object and multiplies the gravvity scale by the modifier.

Parameters

modifier	Modifier.

5.45.2.3 void MultiStack.TurnManager.ApplyShapeModifier (Shape *shape*)

When invoked only the specified shape will be spawned this round.

Parameters

shape	Shape.

5.45.2.4 void MultiStack.TurnManager.ApplySizeModifier (float *modifier*)

Applies a size modifier to each spawned physics object for this round.

Parameters

modifier	Modifier.

5.45.2.5 void MultiStack.TurnManager.ChangeShapeToExplosive ( )

Changes a shape to explosive. The most recent shape that can be turned into an explosive is selected.

5.45.2.6 void MultiStack.TurnManager.ChangeShapeToGlass ( )

Changes a shape to glass. The most recent shape that can be turn into glass is selected.

5.45.2.7 void MultiStack.TurnManager.DisableCatchTheShapeModifier ( )

Disables the catch the shape modifier.

5.45.2.8 void MultiStack.TurnManager.DisableGravityModifier ( )

Disables the gravity modifier. Resets each spawned physics object gravity scale.

5.45.2.9 void MultiStack.TurnManager.DisableShapeModifier ( )

Disables the shape modifier. Any type of shape will be spawned next round.

5.45.2.10 void MultiStack.TurnManager.DisableSizeModifier ( )

Disables the size modifier.

5.45.2.11 void MultiStack.TurnManager.Initialise (int *numOfPlayers*)

Sets number of players. Loads MultiStack.TurnManager.physicsObjects.

Parameters

numOfPlayers	Number of players.

5.45.2.12 void MultiStack.TurnManager.ObjectDropped (Rigidbody2D *obj*)

Specifies that an object has been picked dropped. This is used to help decide if the current player has finished their turn.

Parameters

obj	Object.

5.45.2.13 void MultiStack.TurnManager.ObjectPickedUp (Rigidbody2D *obj*)

Specifies that an object has been picked up. This is used to help decide if the current player has finished their turn.

Parameters

obj	Object.

5.45.2.14 void MultiStack.TurnManager.StartNewRound ( )

Starts a new round, increments current round counter and begins a new turn if not gameover.

5.45.3 Member Data Documentation

5.45.3.1 int MultiStack.TurnManager.baseTimeForMove = 20

The base tim (without modifiers) that each player has to complete their turn.

5.45.3.2 PlayerPhysicsObject [ ] MultiStack.TurnManager.physicsObjects

The physics objects that can be spawned.

5.45.3.3 PlayerColours MultiStack.TurnManager.playerColours

The player colours. The MultiStack.TurnManager.playerText UI is shown in the current players colour.

5.45.3.4 Transform [ ] MultiStack.TurnManager.playerLocations

The player locations where the plays shape will spawn.

5.45.3.5 GameText MultiStack.TurnManager.playerText

The UI that will inform the player it is their turn.

5.45.3.6 GameText MultiStack.TurnManager.timeRemainingText

The UI that shows the current players time remaining for their turn.

5.45.4 Property Documentation

5.45.4.1 int MultiStack.TurnManager.currentEnhancedHeight `[get]`

Gets the height of the current stack of physics objects + 4f. This is used for the UI.

The height of the current.

5.45.4.2 int MultiStack.TurnManager.currentPlayer [get]

The index of the current player.

The current player.

5.45.4.3 int MultiStack.TurnManager.currentRound [get]

The current round number.

The current round.

5.45.4.4 bool MultiStack.TurnManager.explosiveSpriteAvailableForSpawnedObject [get]

Gets a value indicating whether this MultiStack.TurnManager has spawned a physics object that can be turned into an explosive object.

`true` if explosive sprite available for spawned object; otherwise, `false`.

5.45.4.5 bool MultiStack.TurnManager.glassSpriteAvailableForSpawnedObject [get]

Gets a value indicating whether this MultiStack.TurnManager has spawned an object that can be turned into a glass object.

`true` if glass sprite available for spawned object; otherwise, `false`.

5.45.4.6 TurnManager MultiStack.TurnManager.instance [static], [get]

Signleton pattern. Gets the instance. Accessible from any class.

The instance.

5.45.4.7 int MultiStack.TurnManager.numOfObjectsSpawned [get]

The number of physics objects in the current scene.

The number of objects spawned.

5.45.4.8 bool MultiStack.TurnManager.randomPlayerOrder [set]

Sets a value indicating whether this MultiStack.TurnManager should shuffle the players for the next round.

`true` if random player order; otherwise, `false`.

5.45.4.9 float MultiStack.TurnManager.shapeOffsetHeight [get]

Gets the height of the shape stack offset. Used by the MultiStack.CameraManager to limit how far the camera can scroll upwards.

The height of the shape offset.

5.45.4.10 bool MultiStack.TurnManager.shouldUpdate [set]

Sets a value indicating whether this MultiStack.TurnManager should update.

`true` if should update; otherwise, `false`.

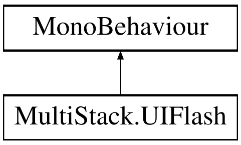
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Managers/TurnManager.cs

5.46 MultiStack.UIFlash Class Reference

User interface flash. Acts as an overlay for the main menu and gameplay scene.

Inheritance diagram for MultiStack.UIFlash:



Public Member Functions

- void CallbackOnUIFlashComplete (CallBack callback)

  Add a callback delegate to be invoked when the UI flash is complete.
- void GameOverUIFlashInSeconds (float seconds)

  Games the over user interface flash.

Public Attributes

- bool isMenu

  Overlay behaviour is different for menu and game scene. For menu, the overlay is less translucent.

5.46.1 Detailed Description

User interface flash. Acts as an overlay for the main menu and gameplay scene.

5.46.2 Member Function Documentation

5.46.2.1 void MultiStack.UIFlash.CallbackOnUIFlashComplete (CallBack *callback*)

Add a callback delegate to be invoked when the UI flash is complete.

Parameters

callback	Callback.

5.46.2.2 void MultiStack.UIFlash.GameOverUIFlashInSeconds (float *seconds*)

Games the over user interface flash.

5.46.3 Member Data Documentation

5.46.3.1 bool MultiStack.UIFlash.isMenu

Overlay behaviour is different for menu and game scene. For menu, the overlay is less translucent.

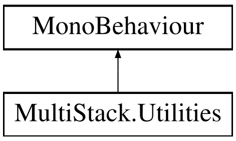
The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/UI/UIFlash.cs

5.47 MultiStack.Utilities Class Reference

Useful methods used by a number of classes.

Inheritance diagram for MultiStack.Utilities:



Public Member Functions

- Quaternion GetRandomRotation2D ()

  Gets a random 2d rotation (z-axis).
- void InvokeMethodEverySeconds (CallBack method, float seconds)

  Invokes the delegate method at a time interval specified by seconds.

Properties

- static Utilities instance `[get]`

  Gets the instance of this class. Only one instance exists. Provides static access from any class.

5.47.1 Detailed Description

Useful methods used by a number of classes.

5.47.2 Member Function Documentation

5.47.2.1 Quaternion MultiStack.Utilities.GetRandomRotation2D ( )

Gets a random 2d rotation (z-axis).

Returns

The random rotation.

5.47.2.2 void MultiStack.Utilities.InvokeMethodEverySeconds (CallBack *method,* float *seconds*)

Invokes the delegate method at a time interval specified by seconds.

Parameters

method	Method.
seconds	Time between each method call.

5.47.3 Property Documentation

5.47.3.1 Utilities MultiStack.Utilities.instance [static], [get]

Gets the instance of this class. Only one instance exists. Provides static access from any class.

The instance.

The documentation for this class was generated from the following file:

- MultiplayerStacker/Scripts/Helpers/Utilities.cs

Index

Activate
- MultiStack::CanOnlyPickupOnceModifier, 18
- MultiStack::CatchTheShapeModifier, 19
- MultiStack::ExplosiveObject, 29
- MultiStack::ExplosiveShapeModifier, 30
- MultiStack::GameModifier, 35
- MultiStack::GameOverUI, 38
- MultiStack::GlassObject, 42
- MultiStack::GlassShapeModifier, 43
- MultiStack::GravityModifier, 45
- MultiStack::IncreaseShapeSizeWhileHeldModifier, 46
- MultiStack::OneShapeModifier, 50
- MultiStack::RandomPlayerOrderModifier, 53
- MultiStack::ReverseDragModifier, 54
- MultiStack::ShakeyShapesModifier, 56
- MultiStack::ShapeMovementSpeedModifier, 57
- MultiStack::ShapeSizeModifier, 58
- MultiStack::SpringModifier, 58
- MultiStack::TimeForMoveModifier, 62

ApplyCatchTheShapeModifier
- MultiStack::TurnManager, 64

ApplyGravityModifier
- MultiStack::TurnManager, 64

ApplyNewModifier
- MultiStack::GameModifierManager, 36

ApplyShapeModifier
- MultiStack::TurnManager, 64

ApplySizeModifier
- MultiStack::TurnManager, 65

backgroundPrefab
- MultiStack::Background, 13

backgroundTilePrefab
- MultiStack::Background, 13

backgrounds
- MultiStack::BackgroundManager, 14

baseTimeForMove
- MultiStack::TurnManager, 66

boxMask
- MultiStack::ClickHandler, 21
- MultiStack::FallHandler, 31
- MultiStack::GlassObject, 42

brokenStages
- MultiStack::GlassObject, 42

CallBack
- MultiStack, 12

CallBackOnStageComplete
- MultiStack::StageSelector, 60

CallBackOnTargetReached
- MultiStack::CameraManager, 15

CallbackOnUIFlashComplete
- MultiStack::UIFlash, 68

canOnlyClickOnce
- MultiStack::ClickHandler, 21

ChangeFontColour
- MultiStack::GameText, 40

ChangeShapeToExplosive
- MultiStack::TurnManager, 65

ChangeShapeToGlass
- MultiStack::TurnManager, 65

changedTime
- MultiStack::TimeForMoveModifier, 62

clearOnStart
- MultiStack::GameText, 40

clickable
- MultiStack::ClickableObject, 20

Clicked
- MultiStack::ClickableObject, 20

cloudController
- MultiStack::CloudReturner, 25

cloudPrefabs
- MultiStack::CloudController, 24

colours
- MultiStack::PlayerColours, 51

ConditionsMet
- MultiStack::ExplosiveShapeModifier, 30
- MultiStack::GameModifier, 35
- MultiStack::GlassShapeModifier, 43

crackedAudioClip
- MultiStack::GlassObject, 42

currentEnhancedHeight
- MultiStack::TurnManager, 66

currentPlayer
- MultiStack::TurnManager, 66

currentRound
- MultiStack::TurnManager, 67

Deactivate
- MultiStack::CanOnlyPickupOnceModifier, 18
- MultiStack::CatchTheShapeModifier, 19
- MultiStack::ExplosiveShapeModifier, 31
- MultiStack::GameModifier, 35
- MultiStack::GlassShapeModifier, 44
- MultiStack::GravityModifier, 45
- MultiStack::IncreaseShapeSizeWhileHeldModifier, 46
- MultiStack::OneShapeModifier, 50
- MultiStack::RandomPlayerOrderModifier, 54

MultiStack::ReverseDragModifier, 54
MultiStack::ShakeyShapesModifier, 56
MultiStack::ShapeMovementSpeedModifier, 57
MultiStack::ShapeSizeModifier, 58
MultiStack::SpringModifier, 58
MultiStack::TimeForMoveModifier, 62
DisableCatchTheShapeModifier
MultiStack::TurnManager, 65
DisableGravityModifier
MultiStack::TurnManager, 65
DisableShapeModifier
MultiStack::TurnManager, 65
DisableSizeModifier
MultiStack::TurnManager, 65
dragLine
MultiStack::ClickHandler, 21
dragReversed
MultiStack::ClickHandler, 21

Execute
MultiStack::ExplosiveObject, 29
ExecuteDestroy
MultiStack::Destroy, 27
explodedObjects
MultiStack::ExplosiveObject, 29
explosionAudioClip
MultiStack::ExplosiveObject, 29
explosionPrefab
MultiStack::ExplosiveObject, 29
explosiveForce
MultiStack::ExplosiveObject, 29
explosiveRadius
MultiStack::ExplosiveObject, 29
explosiveSprite
MultiStack::ExplosiveObject, 29
explosiveSpriteAvailableForSpawnedObject
MultiStack::TurnManager, 67

gameModifierManager
MultiStack::GameManager, 33
GameOverType
MultiStack, 11
gameOverUI
MultiStack::GameManager, 33
GameOverUIFlashInSeconds
MultiStack::UIFlash, 68
GetRandomRotation2D
MultiStack::Utilities, 69
glassSpriteAvailableForSpawnedObject
MultiStack::TurnManager, 67
gravityModifier
MultiStack::GravityModifier, 45

hasReachedTarget
MultiStack::CameraManager, 16
height
MultiStack::DataPersistence, 27
heightReachedText
MultiStack::GameOverUI, 38
heightReachedTitle
MultiStack::GameOverUI, 38
HideText
MultiStack::GameText, 40
HideTextAfterTime
MultiStack::GameText, 40

increaseShapeSize
MultiStack::ClickHandler, 21
infoText
MultiStack::GameManager, 33
MultiStack::StageSelector, 61
initialGlassSprite
MultiStack::GlassObject, 42
Initialise
MultiStack::TurnManager, 65
instance
MultiStack::DataPersistence, 27
MultiStack::GameManager, 33
MultiStack::TurnManager, 67
MultiStack::Utilities, 70
instantiatedPrefab
MultiStack::PlayerPhysicsObject, 53
InvokeMethodEverySeconds
MultiStack::Utilities, 69
IsEnabled
MultiStack::GameModifier, 35
isGameOver
MultiStack::GameManager, 33
isMainMenuScene
MultiStack::ClickHandler, 21
isMenu
MultiStack::UIFlash, 68

Load
MultiStack::DataPersistence, 25
lowOnTimeThreshold
MultiStack::MoveTime, 49
lowTimeTextColour
MultiStack::MoveTime, 49

mainMenuButton
MultiStack::GameOverUI, 38
maxSpeed
MultiStack::Cloud, 22
maxTimeBetweenClouds
MultiStack::CloudController, 24
minSpeed
MultiStack::Cloud, 22
minTimeBetweenClouds
MultiStack::CloudController, 24
modifierName
MultiStack::GameModifier, 35
MultiStack, 9
CallBack, 12
GameOverType, 11
Shape, 11
MultiStack.Background, 13
MultiStack.BackgroundManager, 13

MultiStack.CameraManager, 14
MultiStack.CameraShake, 16
MultiStack.CanOnlyPickupOnceModifier, 17
MultiStack.CatchTheShapeModifier, 18
MultiStack.ClickHandler, 20
MultiStack.ClickableObject, 19
MultiStack.Cloud, 22
MultiStack.CloudController, 23
MultiStack.CloudReturner, 24
MultiStack.DataPersistence, 25
MultiStack.Destroy, 27
MultiStack.ExplosiveObject, 28
MultiStack.ExplosiveShapeModifier, 30
MultiStack.FallHandler, 31
MultiStack.GameManager, 32
MultiStack.GameModifier, 34
MultiStack.GameModifierManager, 36
MultiStack.GameOverHandler, 36
MultiStack.GameOverUI, 37
MultiStack.GameText, 39
MultiStack.GlassObject, 41
MultiStack.GlassShapeModifier, 43
MultiStack.GravityModifier, 44
MultiStack.HeightText, 45
MultiStack.Highscore, 45
MultiStack.IncreaseShapeSizeWhileHeldModifier, 46
MultiStack.MainMenu, 47
MultiStack.MoveTime, 48
MultiStack.OneShapeModifier, 49
MultiStack.PlayerBoxPlaced, 50
MultiStack.PlayerColours, 51
MultiStack.PlayerNumberCount, 51
MultiStack.PlayerPhysicsObject, 52
MultiStack.RandomPlayerOrderModifier, 53
MultiStack.ReverseDragModifier, 54
MultiStack.ScoreData, 55
MultiStack.ShakeyShapesModifier, 55
MultiStack.ShapeMovementSpeedModifier, 56
MultiStack.ShapeSizeModifier, 57
MultiStack.SpringModifier, 58
MultiStack.StageSelector, 59
MultiStack.TimeForMoveModifier, 61
MultiStack.TurnManager, 62
MultiStack.UIFlash, 68
MultiStack.Utilities, 69
MultiStack::Background
- backgroundPrefab, 13
- backgroundTilePrefab, 13

MultiStack::BackgroundManager
- backgrounds, 14

MultiStack::CameraManager
- CallBackOnTargetReached, 15
- hasReachedTarget, 16
- RequestNewTarget, 15
- ResetPanSpeedOnTargetReached, 15
- ResetTarget, 15
- SetTarget, 16
- shouldPan, 16
- StartPanning, 16
- targetY, 16

MultiStack::CameraShake
- Shake, 17

MultiStack::CanOnlyPickupOnceModifier
- Activate, 18
- Deactivate, 18

MultiStack::CatchTheShapeModifier
- Activate, 19
- Deactivate, 19

MultiStack::ClickHandler
- boxMask, 21
- canOnlyClickOnce, 21
- dragLine, 21
- dragReversed, 21
- increaseShapeSize, 21
- isMainMenuScene, 21
- playerColours, 21
- shakeyShapes, 21
- useSpring, 22
- velocityRatio, 22

MultiStack::ClickableObject
- clickable, 20
- Clicked, 20

MultiStack::Cloud
- maxSpeed, 22
- minSpeed, 22

MultiStack::CloudController
- cloudPrefabs, 24
- maxTimeBetweenClouds, 24
- minTimeBetweenClouds, 24
- PoolCloud, 23
- yRandomOffset, 24

MultiStack::CloudReturner
- cloudController, 25

MultiStack::DataPersistence
- height, 27
- instance, 27
- Load, 25
- round, 27
- Save, 25

MultiStack::Destroy
- ExecuteDestroy, 27

MultiStack::ExplosiveObject
- Activate, 29
- Execute, 29
- explodedObjects, 29
- explosionAudioClip, 29
- explosionPrefab, 29
- explosiveForce, 29
- explosiveRadius, 29
- explosiveSprite, 29
- readyToBeActivated, 29
- requiredForceToExplode, 29

MultiStack::ExplosiveShapeModifier
- Activate, 30
- ConditionsMet, 30
- Deactivate, 31

MultiStack::FallHandler
- boxMask, 31
- panSpeed, 31
- requiredRigidBodyYVelocity, 31

MultiStack::GameManager
- gameModifierManager, 33
- gameOverUI, 33
- infoText, 33
- instance, 33
- isGameOver, 33
- OnGameOver, 33
- OnRoundOver, 33
- stageSelector, 33
- uiFlash, 33

MultiStack::GameModifier
- Activate, 35
- ConditionsMet, 35
- Deactivate, 35
- IsEnabled, 35
- modifierName, 35

MultiStack::GameModifierManager
- ApplyNewModifier, 36

MultiStack::GameOverUI
- Activate, 38
- heightReachedText, 38
- heightReachedTitle, 38
- mainMenuButton, 38
- newRecord, 38
- nonGameoverUIText, 38
- playerLostText, 38
- playerLostTitle, 38
- restartButton, 38
- RestartScene, 38
- ReturnToMainMenu, 38
- roundsReachedText, 39
- roundsReachedTitle, 39
- title, 39

MultiStack::GameText
- ChangeFontColour, 40
- clearOnStart, 40
- HideText, 40
- HideTextAfterTime, 40
- text, 41
- UpdateText, 40

MultiStack::GlassObject
- Activate, 42
- boxMask, 42
- brokenStages, 42
- crackedAudioClip, 42
- initialGlassSprite, 42
- readyToBeActivated, 43
- requiredForceToCrack, 42
- requiredMassToCrack, 42
- smashedAudioClip, 42
- smashedObjectPrefabs, 42

MultiStack::GlassShapeModifier
- Activate, 43
- ConditionsMet, 43
- Deactivate, 44

MultiStack::GravityModifier
- Activate, 45
- Deactivate, 45
- gravityModifier, 45

MultiStack::IncreaseShapeSizeWhileHeldModifier
- Activate, 46
- Deactivate, 46

MultiStack::MainMenu
- numOfPlayersText, 48
- otherText, 48
- playButton, 48
- PlayButtonPressed, 48
- PlayerBoxPlaced, 48
- playerBoxPrefab, 48
- PlayerBoxRemoved, 48
- playerBoxSpawnLocations, 48
- titleText, 48

MultiStack::MoveTime
- lowOnTimeThreshold, 49
- lowTimeTextColour, 49
- normalTimeTextColour, 49

MultiStack::OneShapeModifier
- Activate, 50
- Deactivate, 50
- shape, 50

MultiStack::PlayerColours
- colours, 51

MultiStack::PlayerNumberCount
- numberOfPlayers, 52

MultiStack::PlayerPhysicsObject
- instantiatedPrefab, 53
- realPrefab, 52
- shape, 52
- simplePrefab, 52
- weight, 53

MultiStack::RandomPlayerOrderModifier
- Activate, 53
- Deactivate, 54

MultiStack::ReverseDragModifier
- Activate, 54
- Deactivate, 54

MultiStack::ShakeyShapesModifier
- Activate, 56
- Deactivate, 56

MultiStack::ShapeMovementSpeedModifier
- Activate, 57
- Deactivate, 57

MultiStack::ShapeSizeModifier
- Activate, 58
- Deactivate, 58

MultiStack::SpringModifier
- Activate, 58
- Deactivate, 58

MultiStack::StageSelector
- CallBackOnStageComplete, 60
- infoText, 61
- SpawnStage, 61

stagePrefabs, 61
stageSpawned, 61
stageText, 61
MultiStack::TimeForMoveModifier
Activate, 62
changedTime, 62
Deactivate, 62
MultiStack::TurnManager
ApplyCatchTheShapeModifier, 64
ApplyGravityModifier, 64
ApplyShapeModifier, 64
ApplySizeModifier, 65
baseTimeForMove, 66
ChangeShapeToExplosive, 65
ChangeShapeToGlass, 65
currentEnhancedHeight, 66
currentPlayer, 66
currentRound, 67
DisableCatchTheShapeModifier, 65
DisableGravityModifier, 65
DisableShapeModifier, 65
DisableSizeModifier, 65
explosiveSpriteAvailableForSpawnedObject, 67
glassSpriteAvailableForSpawnedObject, 67
Initialise, 65
instance, 67
numOfObjectsSpawned, 67
ObjectDropped, 65
ObjectPickedUp, 66
physicsObjects, 66
playerColours, 66
playerLocations, 66
playerText, 66
randomPlayerOrder, 67
shapeOffsetHeight, 67
shouldUpdate, 67
StartNewRound, 66
timeRemainingText, 66
MultiStack::UIFlash
CallbackOnUIFlashComplete, 68
GameOverUIFlashInSeconds, 68
isMenu, 68
MultiStack::Utilities
GetRandomRotation2D, 69
instance, 70
InvokeMethodEverySeconds, 69

newRecord
MultiStack::GameOverUI, 38
nonGameoverUIText
MultiStack::GameOverUI, 38
normalTimeTextColour
MultiStack::MoveTime, 49
numOfObjectsSpawned
MultiStack::TurnManager, 67
numOfPlayersText
MultiStack::MainMenu, 48
numberOfPlayers
MultiStack::PlayerNumberCount, 52

ObjectDropped
MultiStack::TurnManager, 65
ObjectPickedUp
MultiStack::TurnManager, 66
OnGameOver
MultiStack::GameManager, 33
OnRoundOver
MultiStack::GameManager, 33
otherText
MultiStack::MainMenu, 48

panSpeed
MultiStack::FallHandler, 31
physicsObjects
MultiStack::TurnManager, 66
playButton
MultiStack::MainMenu, 48
PlayButtonPressed
MultiStack::MainMenu, 48
PlayerBoxPlaced
MultiStack::MainMenu, 48
playerBoxPrefab
MultiStack::MainMenu, 48
PlayerBoxRemoved
MultiStack::MainMenu, 48
playerBoxSpawnLocations
MultiStack::MainMenu, 48
playerColours
MultiStack::ClickHandler, 21
MultiStack::TurnManager, 66
playerLocations
MultiStack::TurnManager, 66
playerLostText
MultiStack::GameOverUI, 38
playerLostTitle
MultiStack::GameOverUI, 38
playerText
MultiStack::TurnManager, 66
PoolCloud
MultiStack::CloudController, 23

randomPlayerOrder
MultiStack::TurnManager, 67
readyToBeActivated
MultiStack::ExplosiveObject, 29
MultiStack::GlassObject, 43
realPrefab
MultiStack::PlayerPhysicsObject, 52
RequestNewTarget
MultiStack::CameraManager, 15
requiredForceToCrack
MultiStack::GlassObject, 42
requiredForceToExplode
MultiStack::ExplosiveObject, 29
requiredMassToCrack
MultiStack::GlassObject, 42
requiredRigidBodyYVelocity
MultiStack::FallHandler, 31
ResetPanSpeedOnTargetReached

MultiStack::CameraManager, 15
ResetTarget
MultiStack::CameraManager, 15
restartButton
MultiStack::GameOverUI, 38
RestartScene
MultiStack::GameOverUI, 38
ReturnToMainMenu
MultiStack::GameOverUI, 38
round
MultiStack::DataPersistence, 27
roundsReachedText
MultiStack::GameOverUI, 39
roundsReachedTitle
MultiStack::GameOverUI, 39

Save
MultiStack::DataPersistence, 25
SetTarget
MultiStack::CameraManager, 16
Shake
MultiStack::CameraShake, 17
shakeyShapes
MultiStack::ClickHandler, 21
Shape
MultiStack, 11
shape
MultiStack::OneShapeModifier, 50
MultiStack::PlayerPhysicsObject, 52
shapeOffsetHeight
MultiStack::TurnManager, 67
shouldPan
MultiStack::CameraManager, 16
shouldUpdate
MultiStack::TurnManager, 67
simplePrefab
MultiStack::PlayerPhysicsObject, 52
smashedAudioClip
MultiStack::GlassObject, 42
smashedObjectPrefabs
MultiStack::GlassObject, 42
SpawnStage
MultiStack::StageSelector, 61
stagePrefabs
MultiStack::StageSelector, 61
stageSelector
MultiStack::GameManager, 33
stageSpawned
MultiStack::StageSelector, 61
stageText
MultiStack::StageSelector, 61
StartNewRound
MultiStack::TurnManager, 66
StartPanning
MultiStack::CameraManager, 16

targetY
MultiStack::CameraManager, 16
text
MultiStack::GameText, 41
timeRemainingText
MultiStack::TurnManager, 66
title
MultiStack::GameOverUI, 39
titleText
MultiStack::MainMenu, 48

uiFlash
MultiStack::GameManager, 33
UpdateText
MultiStack::GameText, 40
useSpring
MultiStack::ClickHandler, 22

velocityRatio
MultiStack::ClickHandler, 22

weight
MultiStack::PlayerPhysicsObject, 53

yRandomOffset
MultiStack::CloudController, 24