My Project

Generated by Doxygen 1.8.6

Wed Apr 22 2015 22:34:19

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

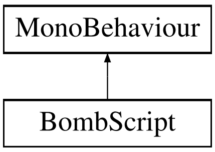
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 BombScript Class Reference

Inheritance diagram for BombScript:



Public Member Functions

- void Start ()

  Stuff to do at the start, gets attached animator and audio source
- void Update ()

  Updates per frame, moves object left per frame, checks if dead, if true, starts death animation& sound then destroys object
- void OnTriggerEnter (Collider col)

Public Attributes

- Animator BombEnemy

  The bomb enemy.
- AudioSource ExplosionSD

  The explosion S.
- float EnemySpeed = 0.1f

  The enemy speed.

3.1.1 Member Function Documentation

3.1.1.1 void BombScript.OnTriggerEnter (Collider *col*) `[inline]`

Raises the trigger enter event.

Parameters

col	Col collision parameter to detect collisions
return	Tag "Player" collision causes Health call, Dmg received & object destruct

3.1.1.2 void BombScript.Start ( ) [inline]

Stuff to do at the start, gets attached animator and audio source

3.1.1.3 void BombScript.Update ( ) [inline]

Updates per frame, moves object left per frame, checks if dead, if true, starts death animation& sound then destroys object

3.1.2 Member Data Documentation

3.1.2.1 Animator BombScript.BombEnemy

The bomb enemy.

3.1.2.2 float BombScript.EnemySpeed = 0.1f

The enemy speed.

3.1.2.3 AudioSource BombScript.ExplosionSD

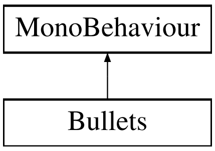
The explosion S.

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

- BombScript.cs

3.2 Bullets Class Reference

Inheritance diagram for Bullets:



Public Member Functions

- void Update ()

  Update called once per frame, moves object right, destroys object after certain time
- void OnTriggerEnter (Collider Hit)

  Collider trigger, on hit do something

Public Attributes

- float Bulletspeed = 5

  The bulletspeed.
- float Dmg = 5

  The dmg.

3.2.1 Member Function Documentation

3.2.1.1 void Bullets.OnTriggerEnter (Collider *Hit*) [inline]

Collider trigger, on hit do something

Parameters

Hit	Hit collision detect
Return	> If hit is obstacle, destroy object and this, if hit is enemy, enemy is damaged

3.2.1.2 void Bullets.Update ( ) [inline]

Update called once per frame, moves object right, destroys object after certain time

3.2.2 Member Data Documentation

3.2.2.1 float Bullets.Bulletspeed = 5

The bulletspeed.

3.2.2.2 float Bullets.Dmg = 5

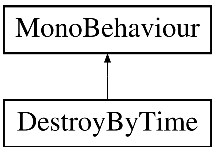
The dmg.

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

- Bullets.cs

3.3 DestroyByTime Class Reference

Inheritance diagram for DestroyByTime:



Public Member Functions

- void Start ()

  Inititation, destroy object at lifetime

Public Attributes

- float lifetime

  The lifetime of an object before being destroyed

3.3.1 Member Function Documentation

3.3.1.1 void DestroyByTime.Start ( ) [inline]

Inititation, destroy object at lifetime

3.3.2 Member Data Documentation

3.3.2.1 float DestroyByTime.lifetime

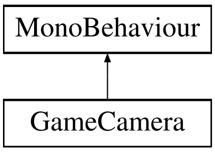
The lifetime of an object before being destroyed

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

- DestroyByTime.cs

3.4 GameCamera Class Reference

Inheritance diagram for GameCamera:



Public Member Functions

- void SetTarget (Transform t)

  Sets the target to aim for
- void Update ()

  Update per frame, transforms object to target vector
- float IncrementTowards (float n, float target, float a)

  Increments param n towards target by speed

Public Attributes

- Transform target

  vector target to aim for
- float trackSpeed = 10

  The track speed.

3.4.1 Member Function Documentation

3.4.1.1 float GameCamera.IncrementTowards (float *n*, float *target*, float *a*) [inline]

Increments param n towards target by speed

Returns

The towards.

Parameters

n	First vector, current position vector of cam
target	Second vector, target vector for cam
a	The alpha component.

3.4.1.2 void GameCamera.SetTarget (Transform *t*) [inline]

Sets the target to aim for

Parameters

t	vector to aim for

3.4.1.3 void GameCamera.Update ( ) [inline]

Update per frame, transforms object to target vector

3.4.2 Member Data Documentation

3.4.2.1 Transform GameCamera.target

vector target to aim for

3.4.2.2 float GameCamera.trackSpeed = 10

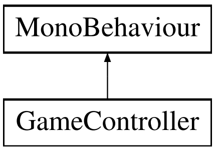
The track speed.

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

- GameCamera.cs

3.5 GameController Class Reference

Inheritance diagram for GameController:



Public Member Functions

- void Update ()

  Update per frame, randomizer for spawning between objects
- void Start ()

  initializer to start time for waves, sets hazards to spawn, spawn time, wave time and position to spawn

Public Attributes

- GameObject Object1

  1st Placement for gameObject spawning
- GameObject Object2

  2nd Placement for gameObject spawning
- GameObject Object3

  3rd Placement for gameObject spawning
- GameObject hazard

  Hazard spot for randomizing between 3 objects
- Vector3 spawnValues

  The spawn values.
- int hazardCount

  The hazard count.
- float spawnWait

  The spawn wait.
- float startWait

  The start wait.
- float waveWait

  The wave wait.
- int choice

  Float Randomizer for objects summary>

3.5.1 Member Function Documentation

3.5.1.1 void GameController.Start ( ) [inline]

initializer to start time for waves, sets hazards to spawn, spawn time, wave time and position to spawn

3.5.1.2 void GameController.Update ( ) [inline]

Update per frame, randomizer for spawning between objects

3.5.2 Member Data Documentation

3.5.2.1 GameObject GameController.hazard

Hazard spot for randomizing between 3 objects

3.5.2.2 int GameController.hazardCount

The hazard count.

3.5.2.3 GameObject GameController.Object1

1st Placement for gameObject spawning

3.5.2.4 GameObject GameController.Object2

2nd Placement for gameObject spawning

3.5.2.5 GameObject GameController.Object3

3rd Placement for gameObject spawning

3.5.2.6 Vector3 GameController.spawnValues

The spawn values.

3.5.2.7 float GameController.spawnWait

The spawn wait.

3.5.2.8 float GameController.startWait

The start wait.

3.5.2.9 float GameController.waveWait

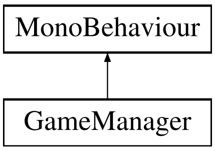
The wave wait.

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

- GameController.cs

3.6 GameManager Class Reference

Inheritance diagram for GameManager:



Public Member Functions

- void Start ()

  At start, get attached gameobject and spawn player
- void SpawnPlayer ()

  Spawns the player and sets camera on player

Public Attributes

- GameObject player

  Gameobject spot to place player
- GameCamera cam

  Gameobject camera

3.6.1 Member Function Documentation

3.6.1.1 void GameManager.SpawnPlayer ( ) [inline]

Spawns the player and sets camera on player

3.6.1.2 void GameManager.Start ( ) [inline]

At start, get attached gameobject and spawn player

3.6.2 Member Data Documentation

3.6.2.1 GameCamera GameManager.cam

Gameobject camera

3.6.2.2 GameObject GameManager.player

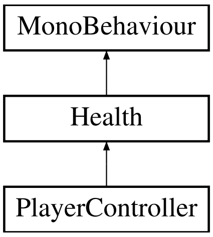
Gameobject spot to place player

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

- GameManager.cs

3.7 Health Class Reference

Inheritance diagram for Health:



Public Member Functions

- void ReceiveDmg (float dmg)

  Damage function, subtracts dmg from current health ///
- bool isDead ()

  Checks if the object is dead depending on current health ///

Public Attributes

- float currentHealth

  The current health.

3.7.1 Member Function Documentation

3.7.1.1 bool Health.isDead ( ) [inline]

Checks if the object is dead depending on current health ///

Returns

`true`, if dead was dead, `false` target is still alive.

3.7.1.2 void Health.ReceiveDmg (float *dmg*) [inline]

Damage function, subtracts dmg from current health ///

Parameters

dmg	Damage to take

3.7.2 Member Data Documentation

3.7.2.1 float Health.currentHealth

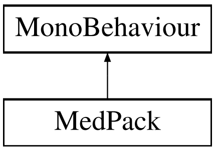
The current health.

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

- Health.cs

3.8 MedPack Class Reference

Inheritance diagram for MedPack:



Public Member Functions

- void Update ()

  Updates per frame, moves object left

- void OnTriggerEnter (Collider col)

  Object trigger on collision

Public Attributes

- float MedSpeed = 0.1f

    The medpack speed.
- float HealAMT = 5

    The total heal amount ///

3.8.1 Member Function Documentation

3.8.1.1 void MedPack.OnTriggerEnter (Collider *col*) [inline]

Object trigger on collision

Parameters

col	Collider col
Return	If col == player, player gets health and this is destroyed

3.8.1.2 void MedPack.Update ( ) [inline]

Updates per frame, moves object left

3.8.2 Member Data Documentation

3.8.2.1 float MedPack.HealAMT = 5

The total heal amount ///

3.8.2.2 float MedPack.MedSpeed = 0.1f

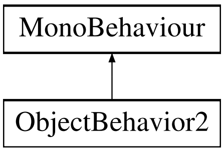
The medpack speed.

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

- MedPack.cs

3.9 ObjectBehavior2 Class Reference

Inheritance diagram for ObjectBehavior2:



Public Attributes

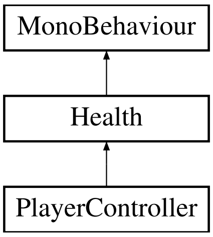
- float **speed** = -15f
- float **timeStamp**

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

- ObjectBehavior2.cs

3.10 PlayerController Class Reference

Inheritance diagram for PlayerController:



Public Member Functions

- void Start ()

  Gets player physics, animator and sets audio sources to listen
- void Update ()

  Update per frame. check if player is dead, sets player speed, checks for user input

Public Attributes

- float gravity = 20

  The gravity.
- float speed = 8

  The speed.
- float acceleration = 30

  The acceleration.
- float jumpHeight = 12

  The height of the jump.
- float currentSpeed

  The current speed.
- float targetSpeed

  The target speed.
- Vector2 amountToMove

  The amount to move.
- AudioSource ShotSD

  The shot Sound
- AudioSource RollSD

  The roll Sound
- AudioSource JumpSD

  The jump Sound
- AudioSource DeathSD

  The death Sound
- PlayerPhysics playerPhysics

  Gets player physics script
- Animator Player

  Gets player animator

3.10.1 Member Function Documentation

3.10.1.1 void PlayerController.Start ( ) [inline]

Gets player physics, animator and sets audio sources to listen

3.10.1.2 void PlayerController.Update ( ) [inline]

Update per frame. check if player is dead, sets player speed, checks for user input

3.10.2 Member Data Documentation

3.10.2.1 float PlayerController.acceleration = 30

The acceleration.

3.10.2.2 Vector2 PlayerController.amountToMove

The amount to move.

3.10.2.3 float PlayerController.currentSpeed

The current speed.

3.10.2.4 AudioSource PlayerController.DeathSD

The death Sound

3.10.2.5 float PlayerController.gravity = 20

The gravity.

3.10.2.6 float PlayerController.jumpHeight = 12

The height of the jump.

3.10.2.7 AudioSource PlayerController.JumpSD

The jump Sound

3.10.2.8 Animator PlayerController.Player

Gets player animator

3.10.2.9 **PlayerPhysics** PlayerController.playerPhysics

Gets player physics script

3.10.2.10 AudioSource PlayerController.RollSD

The roll Sound

3.10.2.11 AudioSource PlayerController.ShotSD

The shot Sound

3.10.2.12 float PlayerController.speed = 8

The speed.

3.10.2.13 float PlayerController.targetSpeed

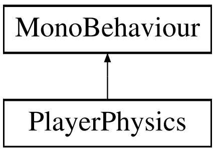
The target speed.

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

- PlayerController.cs

3.11 PlayerPhysics Class Reference

Inheritance diagram for PlayerPhysics:



Public Member Functions

- void Start ()

  Initialization, gets collider of player and its values(size and center)
- void Move (Vector2 moveAmount)

  Move the specified moveAmount.
- void **SetCollider** (Vector3 size, Vector3 centre)
- void **ResetCollider** ()

Public Attributes

- LayerMask collisionMask

  Set the layer for collisionmask, player will collide with this layer
- bool grounded

  Boolean to check if object is in contact with object from bottom
- bool movementStopped

  Boolean to check player is no longer moving

3.11.1 Member Function Documentation

3.11.1.1 void PlayerPhysics.Move (Vector2 *moveAmount*) `[inline]`

Move the specified moveAmount.

Parameters

moveAmount	Move amount.

3.11.1.2 void PlayerPhysics.Start ( ) [inline]

Initialization, gets collider of player and its values(size and center)

3.11.2 Member Data Documentation

3.11.2.1 LayerMask PlayerPhysics.collisionMask

Set the layer for collisionmask, player will collide with this layer

3.11.2.2 bool PlayerPhysics.grounded

Boolean to check if object is in contact with object from bottom

3.11.2.3 bool PlayerPhysics.movementStopped

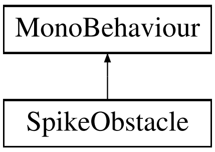
Boolean to check player is no longer moving

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

- PlayerPhysics.cs

3.12 SpikeObstacle Class Reference

Inheritance diagram for SpikeObstacle:



Public Attributes

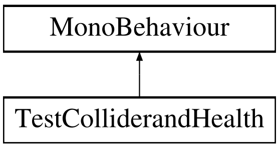
- float **ObstacleSpeed** = 10

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

- SpikeObstacle.cs

3.13 TestColliderandHealth Class Reference

Inheritance diagram for TestColliderandHealth:



Public Member Functions

- void Update ()

  Update per frame to move object left to collide with player

Public Attributes

- float PreviousHealth = 0

  Holder float to check previous health
- float NewHealth = 0

  Holder to check new health

3.13.1 Member Function Documentation

3.13.1.1 void TestColliderandHealth.Update ( ) [inline]

Update per frame to move object left to collide with player

3.13.2 Member Data Documentation

3.13.2.1 float TestColliderandHealth.NewHealth = 0

Holder to check new health

3.13.2.2 float TestColliderandHealth.PreviousHealth = 0

Holder float to check previous health

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

- TestColliderandHealth.cs

Index