

# My Project

Generated by Doxygen 1.8.6

Wed Apr 22 2015 22:34:19



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	BombScript Class Reference . . . . .	5
3.1.1	Member Function Documentation . . . . .	5
3.1.1.1	OnTriggerEnter . . . . .	5
3.1.1.2	Start . . . . .	6
3.1.1.3	Update . . . . .	6
3.1.2	Member Data Documentation . . . . .	6
3.1.2.1	BombEnemy . . . . .	6
3.1.2.2	EnemySpeed . . . . .	6
3.1.2.3	ExplosionSD . . . . .	6
3.2	Bullets Class Reference . . . . .	6
3.2.1	Member Function Documentation . . . . .	7
3.2.1.1	OnTriggerEnter . . . . .	7
3.2.1.2	Update . . . . .	7
3.2.2	Member Data Documentation . . . . .	7
3.2.2.1	Bulletspeed . . . . .	7
3.2.2.2	Dmg . . . . .	7
3.3	DestroyByTime Class Reference . . . . .	7
3.3.1	Member Function Documentation . . . . .	8
3.3.1.1	Start . . . . .	8
3.3.2	Member Data Documentation . . . . .	8
3.3.2.1	lifetime . . . . .	8
3.4	GameCamera Class Reference . . . . .	8
3.4.1	Member Function Documentation . . . . .	9
3.4.1.1	IncrementTowards . . . . .	9
3.4.1.2	SetTarget . . . . .	9

3.4.1.3	Update	9
3.4.2	Member Data Documentation	9
3.4.2.1	target	9
3.4.2.2	trackSpeed	9
3.5	GameController Class Reference	9
3.5.1	Member Function Documentation	10
3.5.1.1	Start	10
3.5.1.2	Update	10
3.5.2	Member Data Documentation	10
3.5.2.1	hazard	10
3.5.2.2	hazardCount	10
3.5.2.3	Object1	11
3.5.2.4	Object2	11
3.5.2.5	Object3	11
3.5.2.6	spawnValues	11
3.5.2.7	spawnWait	11
3.5.2.8	startWait	11
3.5.2.9	waveWait	11
3.6	GameManager Class Reference	11
3.6.1	Member Function Documentation	12
3.6.1.1	SpawnPlayer	12
3.6.1.2	Start	12
3.6.2	Member Data Documentation	12
3.6.2.1	cam	12
3.6.2.2	player	12
3.7	Health Class Reference	12
3.7.1	Member Function Documentation	13
3.7.1.1	isDead	13
3.7.1.2	ReceiveDmg	13
3.7.2	Member Data Documentation	13
3.7.2.1	currentHealth	13
3.8	MedPack Class Reference	13
3.8.1	Member Function Documentation	14
3.8.1.1	OnTriggerEnter	14
3.8.1.2	Update	14
3.8.2	Member Data Documentation	14
3.8.2.1	HealAMT	14
3.8.2.2	MedSpeed	14
3.9	ObjectBehavior2 Class Reference	14
3.10	PlayerController Class Reference	15

3.10.1	Member Function Documentation	16
3.10.1.1	Start	16
3.10.1.2	Update	16
3.10.2	Member Data Documentation	16
3.10.2.1	acceleration	16
3.10.2.2	amountToMove	16
3.10.2.3	currentSpeed	16
3.10.2.4	DeathSD	16
3.10.2.5	gravity	16
3.10.2.6	jumpHeight	16
3.10.2.7	JumpSD	16
3.10.2.8	Player	16
3.10.2.9	playerPhysics	16
3.10.2.10	RollSD	17
3.10.2.11	ShotSD	17
3.10.2.12	speed	17
3.10.2.13	targetSpeed	17
3.11	PlayerPhysics Class Reference	17
3.11.1	Member Function Documentation	18
3.11.1.1	Move	18
3.11.1.2	Start	19
3.11.2	Member Data Documentation	19
3.11.2.1	collisionMask	19
3.11.2.2	grounded	19
3.11.2.3	movementStopped	19
3.12	SpikeObstacle Class Reference	19
3.13	TestColliderandHealth Class Reference	19
3.13.1	Member Function Documentation	20
3.13.1.1	Update	20
3.13.2	Member Data Documentation	20
3.13.2.1	NewHealth	20
3.13.2.2	PreviousHealth	20



# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

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

MonoBehaviour	
BombScript . . . . .	5
Bullets . . . . .	6
DestroyByTime . . . . .	7
GameCamera . . . . .	8
GameController . . . . .	9
GameManager . . . . .	11
Health . . . . .	12
PlayerController . . . . .	15
MedPack . . . . .	13
ObjectBehavior2 . . . . .	14
PlayerPhysics . . . . .	17
SpikeObstacle . . . . .	19
TestColliderandHealth . . . . .	19





## Chapter 2

# Class Index

### 2.1 Class List

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

<a href="#">BombScript</a>	5
<a href="#">Bullets</a>	6
<a href="#">DestroyByTime</a>	7
<a href="#">GameCamera</a>	8
<a href="#">GameController</a>	9
<a href="#">GameManager</a>	11
<a href="#">Health</a>	12
<a href="#">MedPack</a>	13
<a href="#">ObjectBehavior2</a>	14
<a href="#">PlayerController</a>	15
<a href="#">PlayerPhysics</a>	17
<a href="#">SpikeObstacle</a>	19
<a href="#">TestColliderandHealth</a>	19

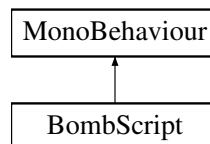


## Chapter 3

# 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

<i>col</i>	Col collision parameter to detect collisions
<i>return</i>	Tag "Player" collision causes <a href="#">Health</a> 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

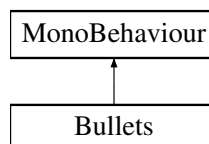
The explosion S.

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

<i>Hit</i>	Hit collision detect
<i>Return</i>	> 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

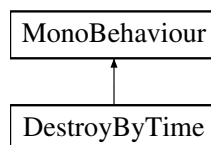
The dmg.

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]

Initiation, destroy object at lifetime

### 3.3.2 Member Data Documentation

#### 3.3.2.1 float DestroyByTime.lifetime

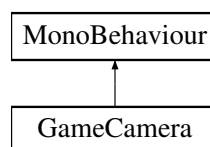
The lifetime of an object before being destroyed

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

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

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

Sets the target to aim for

##### Parameters

<i>t</i>	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`

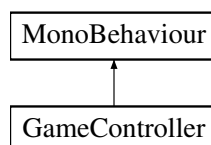
The track speed.

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

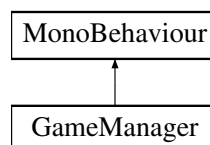
The wave wait.

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

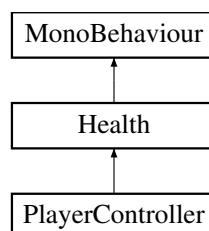
Gameobject spot to place player

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

<i>dmg</i>	Damage to take
------------	----------------

## 3.7.2 Member Data Documentation

### 3.7.2.1 float Health.currentHealth

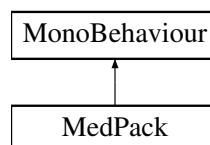
The current health.

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

<i>col</i>	Collider <i>col</i>
<i>Return</i>	If <i>col</i> == 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

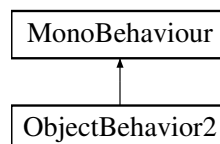
The medpack speed.

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

- MedPack.cs

## 3.9 ObjectBehavior2 Class Reference

Inheritance diagram for ObjectBehavior2:



## Public Attributes

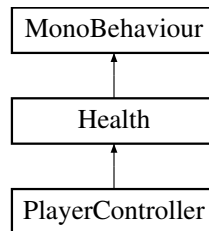
- float **speed** = -15f
- float **timeStamp**

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

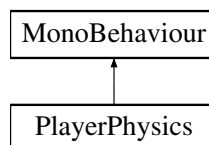
The target speed.

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

<i>moveAmount</i>	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

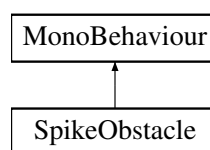
Boolean to check player is no longer moving

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

- PlayerPhysics.cs

## 3.12 SpikeObstacle Class Reference

Inheritance diagram for SpikeObstacle:



## Public Attributes

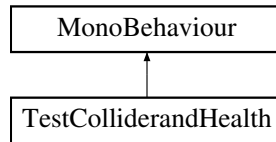
- float **ObstacleSpeed** = 10

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

- acceleration
  - PlayerController, 16
- amountToMove
  - PlayerController, 16
- BombEnemy
  - BombScript, 6
- BombScript, 5
  - BombEnemy, 6
  - EnemySpeed, 6
  - ExplosionSD, 6
  - OnTriggerEnter, 5
  - Start, 6
  - Update, 6
- Bullets, 6
  - Bulletspeed, 7
  - Dmg, 7
  - OnTriggerEnter, 7
  - Update, 7
- Bulletspeed
  - Bullets, 7
- cam
  - GameManager, 12
- collisionMask
  - PlayerPhysics, 19
- currentHealth
  - Health, 13
- currentSpeed
  - PlayerController, 16
- DeathSD
  - PlayerController, 16
- DestroyByTime, 7
  - lifetime, 8
  - Start, 8
- Dmg
  - Bullets, 7
- EnemySpeed
  - BombScript, 6
- ExplosionSD
  - BombScript, 6
- GameCamera, 8
  - IncrementTowards, 9
  - SetTarget, 9
  - target, 9
  - trackSpeed, 9
  - Update, 9
- GameController, 9
  - hazard, 10
  - hazardCount, 10
  - Object1, 10
  - Object2, 11
  - Object3, 11
  - spawnValues, 11
  - spawnWait, 11
  - Start, 10
  - startWait, 11
  - Update, 10
  - waveWait, 11
- GameManager, 11
  - cam, 12
  - player, 12
  - SpawnPlayer, 12
  - Start, 12
- gravity
  - PlayerController, 16
- grounded
  - PlayerPhysics, 19
- hazard
  - GameController, 10
- hazardCount
  - GameController, 10
- HealAMT
  - MedPack, 14
- Health, 12
  - currentHealth, 13
  - isDead, 13
  - ReceiveDmg, 13
- IncrementTowards
  - GameCamera, 9
- isDead
  - Health, 13
- jumpHeight
  - PlayerController, 16
- JumpSD
  - PlayerController, 16
- lifetime
  - DestroyByTime, 8
- MedPack, 13
  - HealAMT, 14
  - MedSpeed, 14
  - OnTriggerEnter, 14
  - Update, 14
- MedSpeed

- MedPack, 14
- Move
  - PlayerPhysics, 18
- movementStopped
  - PlayerPhysics, 19
- NewHealth
  - TestColliderandHealth, 20
- Object1
  - GameController, 10
- Object2
  - GameController, 11
- Object3
  - GameController, 11
- ObjectBehavior2, 14
- OnTriggerEnter
  - BombScript, 5
  - Bullets, 7
  - MedPack, 14
- Player
  - PlayerController, 16
- player
  - GameManager, 12
- PlayerController, 15
  - acceleration, 16
  - amountToMove, 16
  - currentSpeed, 16
  - DeathSD, 16
  - gravity, 16
  - jumpHeight, 16
  - JumpSD, 16
  - Player, 16
  - playerPhysics, 16
  - RollSD, 16
  - ShotSD, 17
  - speed, 17
  - Start, 16
  - targetSpeed, 17
  - Update, 16
- PlayerPhysics, 17
  - collisionMask, 19
  - grounded, 19
  - Move, 18
  - movementStopped, 19
  - Start, 19
- playerPhysics
  - PlayerController, 16
- PreviousHealth
  - TestColliderandHealth, 20
- ReceiveDmg
  - Health, 13
- RollSD
  - PlayerController, 16
- SetTarget
  - GameCamera, 9
- ShotSD
  - PlayerController, 17
- SpawnPlayer
  - GameManager, 12
- spawnValues
  - GameController, 11
- spawnWait
  - GameController, 11
- speed
  - PlayerController, 17
- SpikeObstacle, 19
- Start
  - BombScript, 6
  - DestroyByTime, 8
  - GameController, 10
  - GameManager, 12
  - PlayerController, 16
  - PlayerPhysics, 19
- startWait
  - GameController, 11
- target
  - GameCamera, 9
- targetSpeed
  - PlayerController, 17
- TestColliderandHealth, 19
  - NewHealth, 20
  - PreviousHealth, 20
  - Update, 20
- trackSpeed
  - GameCamera, 9
- Update
  - BombScript, 6
  - Bullets, 7
  - GameCamera, 9
  - GameController, 10
  - MedPack, 14
  - PlayerController, 16
  - TestColliderandHealth, 20
- waveWait
  - GameController, 11