

# Top Down Gun Pack

1.1

Sun Jan 10 2016 19:59:45



# Contents

<b>1</b>	<b>Namespace Index</b>	<b>1</b>
1.1	Packages . . . . .	1
<b>2</b>	<b>Hierarchical Index</b>	<b>3</b>
2.1	Class Hierarchy . . . . .	3
<b>3</b>	<b>Class Index</b>	<b>5</b>
3.1	Class List . . . . .	5
<b>4</b>	<b>Namespace Documentation</b>	<b>7</b>
4.1	TDGP Namespace Reference . . . . .	7
4.1.1	Enumeration Type Documentation . . . . .	8
4.1.1.1	Gun_Type . . . . .	8
4.1.1.2	SPECIAL_DAMAGE_TYPE . . . . .	8
4.2	TDGP.Demo Namespace Reference . . . . .	9
<b>5</b>	<b>Class Documentation</b>	<b>11</b>
5.1	TDGP.AnimationAudio Class Reference . . . . .	11
5.1.1	Detailed Description . . . . .	11
5.1.2	Member Function Documentation . . . . .	11
5.1.2.1	PlaySound() . . . . .	11
5.1.3	Member Data Documentation . . . . .	12
5.1.3.1	AudioClips . . . . .	12
5.1.3.2	InversePlayChance . . . . .	12
5.1.3.3	VolumeScale . . . . .	12
5.2	TDGP.DamageAnimationController Class Reference . . . . .	12
5.2.1	Detailed Description . . . . .	12
5.2.2	Member Function Documentation . . . . .	13
5.2.2.1	ApplyDamage(SPECIAL_DAMAGE_TYPE damageType, float dps, float time) . . . . .	13
5.2.3	Member Data Documentation . . . . .	13
5.2.3.1	ChainLightningRange . . . . .	13
5.2.3.2	IceCaseAnimation . . . . .	13
5.2.3.3	LightningAnimation . . . . .	13

5.2.3.4	MaxChainLightningEnemiesHit	13
5.3	TDGP.Destroy Class Reference	13
5.3.1	Detailed Description	14
5.3.2	Member Function Documentation	14
5.3.2.1	Execute()	14
5.4	TDGP.Disable Class Reference	14
5.4.1	Detailed Description	14
5.4.2	Member Function Documentation	14
5.4.2.1	Execute()	14
5.5	TDGP.EnableParentMovement Class Reference	14
5.5.1	Detailed Description	15
5.5.2	Member Function Documentation	15
5.5.2.1	EnableMovement()	15
5.6	TDGP.EnemyMovement Class Reference	15
5.6.1	Detailed Description	16
5.6.2	Member Function Documentation	16
5.6.2.1	SpawnComplete()	16
5.6.3	Member Data Documentation	16
5.6.3.1	MoveSpeed	16
5.6.4	Property Documentation	16
5.6.4.1	CanMove	16
5.7	TDGP.Demo.EnemySpawner Class Reference	16
5.7.1	Detailed Description	17
5.7.2	Member Function Documentation	17
5.7.2.1	EnemyRemoved()	17
5.7.3	Member Data Documentation	17
5.7.3.1	Enemy	17
5.7.3.2	KillCount	17
5.7.3.3	MaxEnemiesOnScreen	17
5.7.3.4	SpawnTime	17
5.8	TDGP.Demo.Environment Class Reference	17
5.8.1	Detailed Description	18
5.9	TDGP.Demo.FollowTarget Class Reference	18
5.9.1	Detailed Description	18
5.10	TDGP.Gun Class Reference	18
5.10.1	Detailed Description	19
5.10.2	Member Function Documentation	19
5.10.2.1	OnDrop()	19
5.10.2.2	OnPickup()	19
5.10.3	Member Data Documentation	19

5.10.3.1	GunType	19
5.11	TDGP.GunBarrel Class Reference	19
5.11.1	Detailed Description	20
5.11.2	Member Function Documentation	20
5.11.2.1	OnDrop()	20
5.11.2.2	OnFire()	20
5.11.2.3	OnPickup()	20
5.11.3	Member Data Documentation	21
5.11.3.1	FireSounds	21
5.11.3.2	LaunchSpeed	21
5.11.3.3	VolumeScale	21
5.12	TDGP.GunBarrelDelayedFire Class Reference	21
5.12.1	Detailed Description	21
5.12.2	Member Function Documentation	22
5.12.2.1	OnFire()	22
5.12.3	Member Data Documentation	22
5.12.3.1	MaxFireDelay	22
5.12.3.2	MinFireDelay	22
5.13	TDGP.GunBody Class Reference	22
5.13.1	Detailed Description	23
5.13.2	Member Function Documentation	23
5.13.2.1	OnDrop()	23
5.13.2.2	OnPickup()	23
5.13.3	Member Data Documentation	23
5.13.3.1	GunPosition	23
5.14	TDGP.GunClip Class Reference	23
5.14.1	Detailed Description	24
5.14.2	Member Function Documentation	24
5.14.2.1	GetBullet()	24
5.14.2.2	OnDrop()	24
5.14.2.3	OnPickup()	24
5.14.2.4	PoolObject(GameObject obj)	24
5.14.2.5	RequestBullet()	24
5.14.3	Member Data Documentation	25
5.14.3.1	BulletsInClip	25
5.14.3.2	ReloadSpeed	25
5.14.3.3	SupportReload	25
5.15	TDGP.GunComponent Class Reference	25
5.15.1	Detailed Description	26
5.15.2	Member Function Documentation	26

5.15.2.1	OnDrop()	26
5.15.2.2	OnPickup()	26
5.16	TDGP.GunLaserSight Class Reference	26
5.16.1	Detailed Description	27
5.16.2	Member Function Documentation	27
5.16.2.1	OnDrop()	27
5.16.2.2	OnPickup()	27
5.16.3	Member Data Documentation	27
5.16.3.1	Range	27
5.17	TDGP.GunMuzzle Class Reference	27
5.17.1	Detailed Description	28
5.17.2	Member Function Documentation	28
5.17.2.1	OnDrop()	28
5.17.2.2	OnPickup()	28
5.17.3	Member Data Documentation	28
5.17.3.1	FlashAnimationSpeed	28
5.18	TDGP.GunProjectile Class Reference	28
5.18.1	Detailed Description	29
5.18.2	Member Function Documentation	29
5.18.2.1	OnDrop()	29
5.18.2.2	OnPickup()	29
5.18.2.3	OnTriggerEnter2D(Collider2D other)	30
5.18.2.4	Update()	31
5.18.3	Member Data Documentation	31
5.18.3.1	DestroyOnEnemyImpact	31
5.18.3.2	MaxTimeAlive	31
5.18.4	Property Documentation	31
5.18.4.1	Owner	31
5.19	TDGP.GunStock Class Reference	31
5.19.1	Detailed Description	32
5.19.2	Member Function Documentation	32
5.19.2.1	OnDrop()	32
5.19.2.2	OnPickup()	32
5.19.3	Member Data Documentation	32
5.19.3.1	KnockBackForce	32
5.20	TDGP.GunTrigger Class Reference	32
5.20.1	Detailed Description	33
5.20.2	Member Function Documentation	33
5.20.2.1	Awake()	33
5.20.2.2	HandleClickInput()	33

5.20.2.3	HandleHoldInput()	33
5.20.2.4	OnDrop()	34
5.20.2.5	OnPickup()	34
5.20.3	Member Data Documentation	34
5.20.3.1	DelayBetweenProjectiles	34
5.20.3.2	KeyMapping	34
5.20.3.3	ShootType	34
5.21	TDGP.GunTriggerBurst Class Reference	34
5.21.1	Detailed Description	35
5.21.2	Member Function Documentation	35
5.21.2.1	Awake()	35
5.21.2.2	HandleClickInput()	35
5.21.2.3	HandleHoldInput()	35
5.21.3	Member Data Documentation	35
5.21.3.1	BulletsPerBurst	35
5.21.3.2	TimeBetweenBursts	35
5.22	TDGP.GunTriggerSingle Class Reference	36
5.22.1	Detailed Description	36
5.22.2	Member Function Documentation	36
5.22.2.1	Awake()	36
5.22.2.2	HandleClickInput()	36
5.22.2.3	HandleHoldInput()	36
5.22.2.4	OnPickup()	37
5.23	TDGP.Health Class Reference	37
5.23.1	Detailed Description	37
5.23.2	Member Function Documentation	37
5.23.2.1	ApplyDPS(float dps, float time)	37
5.23.2.2	OnHit(float damageAmount)	38
5.23.3	Member Data Documentation	38
5.23.3.1	MaxHealth	38
5.23.3.2	OnDeadAnimation	38
5.23.3.3	OnDeadSprites	38
5.23.3.4	OnHitSounds	38
5.24	TDGP.Holster Class Reference	38
5.24.1	Detailed Description	39
5.24.2	Member Function Documentation	39
5.24.2.1	OnWeaponPickup(Transform weapon)	39
5.25	TDGP.Demo.KillCount Class Reference	39
5.25.1	Detailed Description	39
5.26	TDGP.Demo.LoadSceneOnCollision Class Reference	39

5.26.1 Detailed Description . . . . .	40
5.27 TDGP.Demo.LookAtMouse Class Reference . . . . .	40
5.27.1 Detailed Description . . . . .	40
5.28 TDGP.MissileExplosion Class Reference . . . . .	40
5.28.1 Detailed Description . . . . .	41
5.29 TDGP.MissileTrail Class Reference . . . . .	41
5.29.1 Detailed Description . . . . .	41
5.29.2 Member Function Documentation . . . . .	41
5.29.2.1 OnFire() . . . . .	41
5.29.3 Member Data Documentation . . . . .	42
5.29.3.1 numOfPoints . . . . .	42
5.29.3.2 spread . . . . .	42
5.29.3.3 turnSpeed . . . . .	42
5.29.3.4 updateSpeed . . . . .	42
5.30 TDGP.Demo.PlayerBodyController Class Reference . . . . .	42
5.30.1 Detailed Description . . . . .	42
5.31 TDGP.Demo.PlayerMovementHandler Class Reference . . . . .	43
5.31.1 Detailed Description . . . . .	43
5.32 TDGP.SeekingGunProjectile Class Reference . . . . .	43
5.32.1 Detailed Description . . . . .	44
5.32.2 Member Function Documentation . . . . .	44
5.32.2.1 OnDrop() . . . . .	44
5.32.2.2 OnPickup() . . . . .	44
5.32.2.3 OnTriggerEnter2D(Collider2D other) . . . . .	44
5.32.2.4 Update() . . . . .	44
5.32.3 Member Data Documentation . . . . .	44
5.32.3.1 Damage . . . . .	44
5.32.3.2 DamageAnimation . . . . .	44
5.32.3.3 Velocity . . . . .	45
5.33 TDGP.SpecialGunProjectile Class Reference . . . . .	45
5.33.1 Detailed Description . . . . .	45
5.33.2 Member Function Documentation . . . . .	45
5.33.2.1 OnTriggerEnter2D(Collider2D other) . . . . .	45
5.33.3 Member Data Documentation . . . . .	46
5.33.3.1 DamagePerSecond . . . . .	46
5.33.3.2 DamageTime . . . . .	46
5.33.3.3 DamageType . . . . .	46
5.34 TDGP.StandardGunProjectile Class Reference . . . . .	46
5.34.1 Detailed Description . . . . .	47
5.34.2 Member Function Documentation . . . . .	47



---

5.34.2.1 OnTriggerEnter2D(Collider2D other) . . . . .	47
5.34.3 Member Data Documentation . . . . .	47
5.34.3.1 Damage . . . . .	47
5.34.3.2 DamageAnimation . . . . .	47
5.35 TDGP.Demo.WalkingAnimationController Class Reference . . . . .	47
5.35.1 Detailed Description . . . . .	47
5.36 TDGP.Demo.WeaponSpawner Class Reference . . . . .	48
5.36.1 Detailed Description . . . . .	48
<b>Index</b>	<b>49</b>



# Chapter 1

## Namespace Index

### 1.1 Packages

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

<a href="#">TDGP</a> . . . . .	<a href="#">7</a>
<a href="#">TDGP.Demo</a> . . . . .	<a href="#">9</a>



## Chapter 2

# Hierarchical Index

### 2.1 Class Hierarchy

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

MonoBehaviour	
TDGP.AnimationAudio . . . . .	11
TDGP.DamageAnimationController . . . . .	12
TDGP.Demo.EnemySpawner . . . . .	16
TDGP.Demo.Environment . . . . .	17
TDGP.Demo.FollowTarget . . . . .	18
TDGP.Demo.KillCount . . . . .	39
TDGP.Demo.LoadSceneOnCollision . . . . .	39
TDGP.Demo.LookAtMouse . . . . .	40
TDGP.Demo.PlayerBodyController . . . . .	42
TDGP.Demo.PlayerMovementHandler . . . . .	43
TDGP.Demo.WalkingAnimationController . . . . .	47
TDGP.Demo.WeaponSpawner . . . . .	48
TDGP.Destroy . . . . .	13
TDGP.Disable . . . . .	14
TDGP.EnableParentMovement . . . . .	14
TDGP.EnemyMovement . . . . .	15
TDGP.Gun . . . . .	18
TDGP.GunComponent . . . . .	25
TDGP.GunBarrel . . . . .	19
TDGP.GunBarrelDelayedFire . . . . .	21
TDGP.GunBody . . . . .	22
TDGP.GunClip . . . . .	23
TDGP.GunLaserSight . . . . .	26
TDGP.GunMuzzle . . . . .	27
TDGP.GunProjectile . . . . .	28
TDGP.SeekGunProjectile . . . . .	43
TDGP.SpecialGunProjectile . . . . .	45
TDGP.StandardGunProjectile . . . . .	46
TDGP.GunStock . . . . .	31
TDGP.GunTrigger . . . . .	32
TDGP.GunTriggerBurst . . . . .	34
TDGP.GunTriggerSingle . . . . .	36
TDGP.Health . . . . .	37
TDGP.Holster . . . . .	38
TDGP.MissileExplosion . . . . .	40
TDGP.MissileTrail . . . . .	41



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">TDGP.AnimationAudio</a>	Plays Random audio clip contained in AudioClips. Called my animation. . . . .	11
<a href="#">TDGP.DamageAnimationController</a>	Controls animation and damage for special projectiles (currently ice and chain lightning). . . .	12
<a href="#">TDGP.Destroy</a>	Destroys gameobject. . . . .	13
<a href="#">TDGP.Disable</a>	Disables gameobject. . . . .	14
<a href="#">TDGP.EnableParentMovement</a>	Enables zombie movement when finished spawning. . . . .	14
<a href="#">TDGP.EnemyMovement</a>	Moves enemy towards player. . . . .	15
<a href="#">TDGP.Demo.EnemySpawner</a>	Spawns enemies for the example scene. . . . .	16
<a href="#">TDGP.Demo.Environment</a>	Generates environment for demo scene. . . . .	17
<a href="#">TDGP.Demo.FollowTarget</a>	<a href="#">Demo</a> Script. Moves ENemies towards player. . . . .	18
<a href="#">TDGP.Gun</a>	Calls all child GunComponents OnPickup and OnDrop methods when gun is picked up and dropped respectively. Stores the guns type i.e. one handed, two handed, dual wield. This is used by the <a href="#">Holster</a> script to update the players sprite. . . . .	18
<a href="#">TDGP.GunBarrel</a>	Responsibilities: launching projectiles, playing fire audio, and activating muzzle flash. . . . .	19
<a href="#">TDGP.GunBarrelDelayedFire</a>	<a href="#">Gun</a> barrel with delayed fire. Currently used by the chain gun to produce erratic firing. . . . .	21
<a href="#">TDGP.GunBody</a>	Responsibility: Updating the weapons relative position based on weapon type. . . . .	22
<a href="#">TDGP.GunClip</a>	Responsibilities: act as a object pool for the guns projectiles, provide projectiles when requested by the guns barrel, limit the maximum number of bullets fired by the gun, and reload bullets when requested by player. . . . .	23
<a href="#">TDGP.GunComponent</a>	The base class for all gun components. . . . .	25
<a href="#">TDGP.GunLaserSight</a>	Adds a laser sight to a weapon. Responsible for updating the lasers scale. . . . .	26
<a href="#">TDGP.GunMuzzle</a>	Handles gun muzzle flash. . . . .	27

<a href="#">TDGP.GunProjectile</a>	
The abstract base class for all gun projectiles. Responsible for handling collision, applying damage, and pooling projectile when max time alive has been reached. . . . .	28
<a href="#">TDGP.GunStock</a>	
Responsibility: Providing a knock back force when weapon fired. . . . .	31
<a href="#">TDGP.GunTrigger</a>	
Abstract base class for all gun triggers. Responsibility: Handling player input. . . . .	32
<a href="#">TDGP.GunTriggerBurst</a>	
Provides burst firing. . . . .	34
<a href="#">TDGP.GunTriggerSingle</a>	
Provides single bullet firing on trigger press. . . . .	36
<a href="#">TDGP.Health</a>	
Enemy health script. . . . .	37
<a href="#">TDGP.Holster</a>	
Weapon pickup script, also allows for weapon switching. Used for the demo scene but can be adapted for use by your player. . . . .	38
<a href="#">TDGP.Demo.KillCount</a>	
Shows text on screen for number of enemies killed in demo scene. . . . .	39
<a href="#">TDGP.Demo.LoadSceneOnCollision</a>	
Reloads scene if enemy touches player in demo scene. . . . .	39
<a href="#">TDGP.Demo.LookAtMouse</a>	
Demo script. Play looks at mouse. . . . .	40
<a href="#">TDGP.MissileExplosion</a>	
Missile explosion. Applies knockback force to enemy on explosion. . . . .	40
<a href="#">TDGP.MissileTrail</a>	
Missile trail. Uses line renderer to create a trail effect for the missile. . . . .	41
<a href="#">TDGP.Demo.PlayerBodyController</a>	
Updates player sprite based on currently held weapon. . . . .	42
<a href="#">TDGP.Demo.PlayerMovementHandler</a>	
Demo script. Updates players position based on input. . . . .	43
<a href="#">TDGP.SeekingGunProjectile</a>	
A projectile that seeks towards its target. . . . .	43
<a href="#">TDGP.SpecialGunProjectile</a>	
Projectile of special type i.e. ice or chain lightning. . . . .	45
<a href="#">TDGP.StandardGunProjectile</a>	
A standard gun projectile. . . . .	46
<a href="#">TDGP.Demo.WalkingAnimationController</a>	
Updates walking animation based on movement speed. . . . .	47
<a href="#">TDGP.Demo.WeaponSpawner</a>	
Demo Script. Spawns weapon at players location on button press. . . . .	48



## Chapter 4

# Namespace Documentation

### 4.1 TDGP Namespace Reference

#### Namespaces

- namespace [Demo](#)

#### Classes

- class [AnimationAudio](#)  
*Plays Random audio clip contained in AudioClips. Called my animation.*
- class [DamageAnimationController](#)  
*Controls animation and damage for special projectiles (currently ice and chain lightning).*
- class [Destroy](#)  
*Destroys gameobject.*
- class [Disable](#)  
*Disables gameobject.*
- class [EnableParentMovement](#)  
*Enables zombie movement when finished spawning.*
- class [EnemyMovement](#)  
*Moves enemy towards player.*
- class [Gun](#)  
*Calls all child GunComponents OnPickup and OnDrop methods when gun is picked up and dropped respectively. Stores the guns type i.e. one handed, two handed, dual wield. This is used by the [Holster](#) script to update the players sprite.*
- class [GunBarrel](#)  
*Responsibilities: launching projectiles, playing fire audio, and activating muzzle flash.*
- class [GunBarrelDelayedFire](#)  
*[Gun](#) barrel with delayed fire. Currently used by the chain gun to produce erratic firing.*
- class [GunBody](#)  
*Responsibility: Updating the weapons relative position based on weapon type.*
- class [GunClip](#)  
*Responsibilities: act as a object pool for the guns projectiles, provide projectiles when requested by the guns barrel, limit the maximum number of bullets fired by the gun, and reload bullets when requested by player.*
- class [GunComponent](#)  
*The base class for all gun components.*
- class [GunLaserSight](#)  
*Adds a laser sight to a weapon. Responsible for updating the lasers scale.*

- class [GunMuzzle](#)  
*Handles gun muzzle flash.*
- class [GunProjectile](#)  
*The abstract base class for all gun projectiles. Responsible for handling collision, applying damage, and pooling projectile when max time alive has been reached.*
- class [GunStock](#)  
*Responsibility: Providing a knock back force when weapon fired.*
- class [GunTrigger](#)  
*Abstract base class for all gun triggers. Responsibility: Handling player input.*
- class [GunTriggerBurst](#)  
*Provides burst firing.*
- class [GunTriggerSingle](#)  
*Provides single bullet firing on trigger press.*
- class [Health](#)  
*Enemy health script.*
- class [Holster](#)  
*Weapon pickup script, also allows for weapon switching. Used for the demo scene but can be adapted for use by your player.*
- class [MissileExplosion](#)  
*Missile explosion. Applies knockback force to enemy on explosion.*
- class [MissileTrail](#)  
*Missile trail. Uses line renderer to create a trail effect for the missile.*
- class [SeekingGunProjectile](#)  
*A projectile that seeks towards its target.*
- class [SpecialGunProjectile](#)  
*Projectile of special type i.e. ice or chain lightning.*
- class [StandardGunProjectile](#)  
*A standard gun projectile.*

## Enumerations

- enum [SPECIAL\\_DAMAGE\\_TYPE](#) { **ICE\_CASE**, **CHAIN\_LIGHTNING** }  
*Special damage types.*
- enum [Gun\\_Type](#) { **ONE\_HANDED**, **TWO\_HANDED**, **DUAL\_WIELD** }  
*The gun type.*

### 4.1.1 Enumeration Type Documentation

#### 4.1.1.1 enum TDGP.Gun\_Type [strong]

The gun type.

#### 4.1.1.2 enum TDGP.SPECIAL\_DAMAGE\_TYPE [strong]

Special damage types.

## 4.2 TDGP.Demo Namespace Reference

### Classes

- class [EnemySpawner](#)  
*Spawns enemies for the example scene.*
- class [Environment](#)  
*Generates environment for demo scene.*
- class [FollowTarget](#)  
*Demo Script. Moves ENemies towards player.*
- class [KillCount](#)  
*Shows text on screen for number of enemies killed in demo scene.*
- class [LoadSceneOnCollision](#)  
*Reloads scene if enemy touches player in demo scene.*
- class [LookAtMouse](#)  
*Demo script. Play looks at mouse.*
- class [PlayerBodyController](#)  
*Updates player sprite based on currently held weapon.*
- class [PlayerMovementHandler](#)  
*Demo script. Updates players position based on input.*
- class [WalkingAnimationController](#)  
*Updates walking animation based on movement speed.*
- class [WeaponSpawner](#)  
*Demo Script. Spawns weapon at players location on button press.*



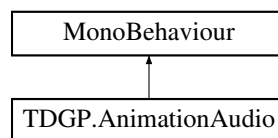
## Chapter 5

# Class Documentation

### 5.1 TDGP.AnimationAudio Class Reference

Plays Random audio clip contained in AudioClips. Called my animation.

Inheritance diagram for TDGP.AnimationAudio:



#### Public Member Functions

- void [PlaySound](#) ()  
*Plays random sound. Called by animation.*

#### Public Attributes

- AudioClip[] [AudioClips](#)  
*The bank of possible audio clips.*
- float [VolumeScale](#) = 1f  
*The volume scale.*
- float [InversePlayChance](#) = 0f  
*The chance a clip will play. Inverse i.e. a value of 0 will mean the clip will play every time and a value of 1 means the clip will never play.*

#### 5.1.1 Detailed Description

Plays Random audio clip contained in AudioClips. Called my animation.

#### 5.1.2 Member Function Documentation

##### 5.1.2.1 void TDGP.AnimationAudio.PlaySound ( )

Plays random sound. Called by animation.

### 5.1.3 Member Data Documentation

#### 5.1.3.1 AudioClip [ ] TDGP.AnimationAudio.AudioClips

The bank of possible audio clips.

#### 5.1.3.2 float TDGP.AnimationAudio.InversePlayChance = 0f

The chance a clip will play. Inverse i.e. a value of 0 will mean the clip will play every time and a value of 1 means the clip will never play.

#### 5.1.3.3 float TDGP.AnimationAudio.VolumeScale = 1f

The volume scale.

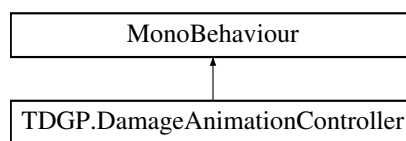
The documentation for this class was generated from the following file:

- AnimationAudio.cs

## 5.2 TDGP.DamageAnimationController Class Reference

Controls animation and damage for special projectiles (currently ice and chain lightning).

Inheritance diagram for TDGP.DamageAnimationController:



### Public Member Functions

- void [ApplyDamage](#) ([SPECIAL\\_DAMAGE\\_TYPE](#) damageType, float dps, float time)  
*Applies damage of type.*

### Public Attributes

- GameObject [IceCaseAnimation](#)  
*Ice case animation prefab.*
- GameObject [LightningAnimation](#)  
*Lightning animation prefab.*
- float [ChainLightningRange](#) = 3f  
*The chain lightning range. Enemies within this proximity will also be struck with chain lightning.*
- int [MaxChainLightningEnemiesHit](#) = 6  
*The maximum number of enemies that can be hit by chain lightning.*

#### 5.2.1 Detailed Description

Controls animation and damage for special projectiles (currently ice and chain lightning).

## 5.2.2 Member Function Documentation

5.2.2.1 void TDGP.DamageAnimationController.ApplyDamage ( SPECIAL\_DAMAGE\_TYPE *damageType*, float *dps*, float *time* )

Applies damage of type.

Parameters

<i>damageType</i>	Damage type.
<i>dps</i>	Damage per second.
<i>time</i>	Seconds damage occurs.

## 5.2.3 Member Data Documentation

5.2.3.1 float TDGP.DamageAnimationController.ChainLightningRange = 3f

The chain lightning range. Enemies within this proximity will also be struck with chain lightning.

5.2.3.2 GameObject TDGP.DamageAnimationController.IceCaseAnimation

Ice case animation prefab.

5.2.3.3 GameObject TDGP.DamageAnimationController.LightningAnimation

Lightning animation prefab.

5.2.3.4 int TDGP.DamageAnimationController.MaxChainLightningEnemiesHit = 6

The maximum number of enemies that can be hit by chain lightning.

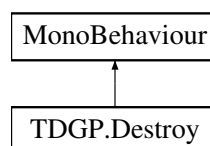
The documentation for this class was generated from the following file:

- DamageAnimationController.cs

## 5.3 TDGP.Destroy Class Reference

Destroys gameobject.

Inheritance diagram for TDGP.Destroy:



### Public Member Functions

- void [Execute](#) ()  
*Executes destroy. Called by animation.*

### 5.3.1 Detailed Description

Destroys gameobject.

### 5.3.2 Member Function Documentation

#### 5.3.2.1 void TDGP.Destroy.Execute ( )

Executes destroy. Called by animation.

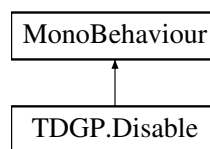
The documentation for this class was generated from the following file:

- Destroy.cs

## 5.4 TDGP.Disable Class Reference

Disables gameobject.

Inheritance diagram for TDGP.Disable:



### Public Member Functions

- void [Execute](#) ()  
*Executes disable. Called by animation.*

### 5.4.1 Detailed Description

Disables gameobject.

### 5.4.2 Member Function Documentation

#### 5.4.2.1 void TDGP.Disable.Execute ( )

Executes disable. Called by animation.

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

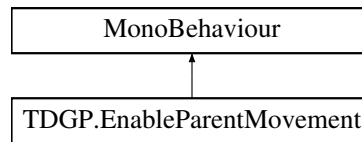
- Disable.cs

## 5.5 TDGP.EnableParentMovement Class Reference

Enables zombie movement when finished spawning.

Inheritance diagram for TDGP.EnableParentMovement:





### Public Member Functions

- void [EnableMovement](#) ()  
*Enables movement. Called by animation.*

#### 5.5.1 Detailed Description

Enables zombie movement when finished spawning.

#### 5.5.2 Member Function Documentation

##### 5.5.2.1 void TDGP.EnableParentMovement.EnableMovement ( )

Enables movement. Called by animation.

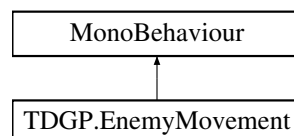
The documentation for this class was generated from the following file:

- EnableParentMovement.cs

## 5.6 TDGP.EnemyMovement Class Reference

Moves enemy towards player.

Inheritance diagram for TDGP.EnemyMovement:



### Public Member Functions

- void [SpawnComplete](#) ()  
*Called by animation, tells this instance that spawning has finished.*

### Public Attributes

- float [MoveSpeed](#)  
*The maximum movement speed.*

### Properties

- bool [CanMove](#) [set]  
*Sets a value indicating whether this instance can move.*

### 5.6.1 Detailed Description

Moves enemy towards player.

### 5.6.2 Member Function Documentation

#### 5.6.2.1 void TDGP.EnemyMovement.SpawnComplete ( )

Called by animation, tells this instance that spawning has finished.

### 5.6.3 Member Data Documentation

#### 5.6.3.1 float TDGP.EnemyMovement.MoveSpeed

The maximum movement speed.

### 5.6.4 Property Documentation

#### 5.6.4.1 bool TDGP.EnemyMovement.CanMove [set]

Sets a value indicating whether this instance can move.

`true` if this instance can move; otherwise, `false`.

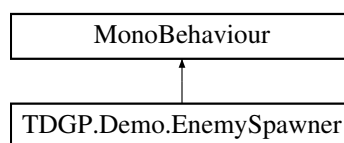
The documentation for this class was generated from the following file:

- EnemyMovement.cs

## 5.7 TDGP.Demo.EnemySpawner Class Reference

Spawns enemies for the example scene.

Inheritance diagram for TDGP.Demo.EnemySpawner:



### Public Member Functions

- void [EnemyRemoved](#) ()  
*Updates kill count.*

### Public Attributes

- GameObject [Enemy](#)  
*The enemy prefab.*
- float [SpawnTime](#) = 0.8f  
*Time between spawns.*
- int [MaxEnemiesOnScreen](#) = 10

*The maximum number of enemies on screen.*

- [KillCount KillCount](#)

*Reference to the kill text.*

### 5.7.1 Detailed Description

Spawns enemies for the example scene.

### 5.7.2 Member Function Documentation

#### 5.7.2.1 void TDGP.Demo.EnemySpawner.EnemyRemoved ( )

Updates kill count.

### 5.7.3 Member Data Documentation

#### 5.7.3.1 GameObject TDGP.Demo.EnemySpawner.Enemy

The enemy prefab.

#### 5.7.3.2 KillCount TDGP.Demo.EnemySpawner.KillCount

Reference to the kill text.

#### 5.7.3.3 int TDGP.Demo.EnemySpawner.MaxEnemiesOnScreen = 10

The maximum number of enemies on screen.

#### 5.7.3.4 float TDGP.Demo.EnemySpawner.SpawnTime = 0.8f

Time between spawns.

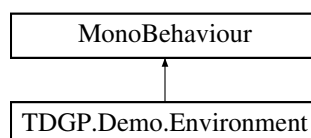
The documentation for this class was generated from the following file:

- EnemySpawner.cs

## 5.8 TDGP.Demo.Environment Class Reference

Generates environment for demo scene.

Inheritance diagram for TDGP.Demo.Environment:



## Public Attributes

- GameObject **Floor**
- GameObject **Wall\_N**
- GameObject **Wall\_NE**
- GameObject **Wall\_E**
- GameObject **Wall\_SE**
- GameObject **Wall\_S**
- GameObject **Wall\_SW**
- GameObject **Wall\_W**
- GameObject **Wall\_NW**
- Vector2 **RoomSize** = new Vector2 (15, 15)

### 5.8.1 Detailed Description

Generates environment for demo scene.

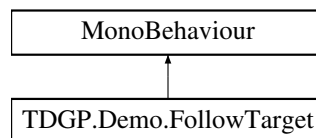
The documentation for this class was generated from the following file:

- Environment.cs

## 5.9 TDGP.Demo.FollowTarget Class Reference

[Demo](#) Script. Moves ENemies towards player.

Inheritance diagram for TDGP.Demo.FollowTarget:



## Public Attributes

- Transform **Target**

### 5.9.1 Detailed Description

[Demo](#) Script. Moves ENemies towards player.

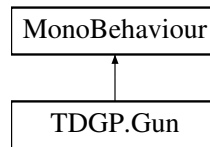
The documentation for this class was generated from the following file:

- FollowTarget.cs

## 5.10 TDGP.Gun Class Reference

Calls all child GunComponents OnPickup and OnDrop methods when gun is picked up and dropped respectively. Stores the guns type i.e. one handed, two handed, dual wield. This is used by the [Holster](#) script to update the players sprite.

Inheritance diagram for TDGP.Gun:



## Public Member Functions

- void [OnPickup](#) ()  
Calls *OnPickup* in all child componenets.
- void [OnDrop](#) ()  
Calls *OnDrop* in all child componenets.

## Public Attributes

- [Gun\\_Type](#) GunType  
The type of the gun.

### 5.10.1 Detailed Description

Calls all child GunComponents OnPickup and OnDrop methods when gun is picked up and dropped respectively. Stores the guns type i.e. one handed, two handed, dual wield. This is used by the [Holster](#) script to update the players sprite.

### 5.10.2 Member Function Documentation

#### 5.10.2.1 void TDGP.Gun.OnDrop ( )

Calls OnDrop in all child componenets.

#### 5.10.2.2 void TDGP.Gun.OnPickup ( )

Calls OnPickup in all child componenets.

### 5.10.3 Member Data Documentation

#### 5.10.3.1 Gun\_Type TDGP.Gun.GunType

The type of the gun.

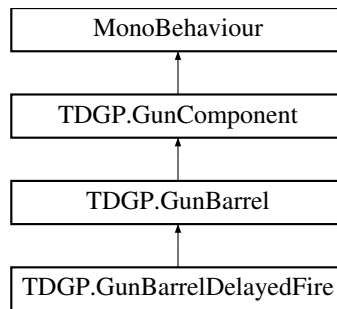
The documentation for this class was generated from the following file:

- Gun.cs

## 5.11 TDGP.GunBarrel Class Reference

Responsibilities: launching projectiles, playing fire audio, and activating muzzle flash.

Inheritance diagram for TDGP.GunBarrel:



## Public Member Functions

- virtual void [OnFire](#) ()  
*Requests bullet from gun clip and if returned, fires bullet based on barrels rotation.*
- override void [OnPickup](#) ()  
*Called by gun component on parent.*
- override void [OnDrop](#) ()  
*Called by gun component on parent. Disables muzzle flashes.*

## Public Attributes

- float [LaunchSpeed](#)  
*The speed at which projectile is launched.*
- float [VolumeScale](#) = 0.2f  
*The volume of the audio played when projectile is fired.*
- AudioClip[] [FireSounds](#)  
*Pool of possible sounds to play on fire.*

### 5.11.1 Detailed Description

Responsibilities: launching projectiles, playing fire audio, and activating muzzle flash.

### 5.11.2 Member Function Documentation

#### 5.11.2.1 override void TDGP.GunBarrel.OnDrop ( ) [virtual]

Called by gun component on parent. Disables muzzle flashes.

Implements [TDGP.GunComponent](#).

#### 5.11.2.2 virtual void TDGP.GunBarrel.OnFire ( ) [virtual]

Requests bullet from gun clip and if returned, fires bullet based on barrels rotation.

Reimplemented in [TDGP.GunBarrelDelayedFire](#).

#### 5.11.2.3 override void TDGP.GunBarrel.OnPickup ( ) [virtual]

Called by gun component on parent.

Implements [TDGP.GunComponent](#).

### 5.11.3 Member Data Documentation

#### 5.11.3.1 AudioClip [ ] TDGP.GunBarrel.FireSounds

Pool of possible sounds to play on fire.

#### 5.11.3.2 float TDGP.GunBarrel.LaunchSpeed

The speed at which projectile is launched.

#### 5.11.3.3 float TDGP.GunBarrel.VolumeScale = 0.2f

The volume of the audio played when projectile is fired.

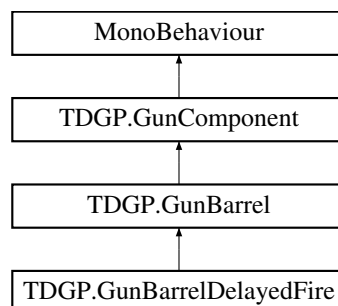
The documentation for this class was generated from the following file:

- GunBarrel.cs

## 5.12 TDGP.GunBarrelDelayedFire Class Reference

[Gun](#) barrel with delayed fire. Currently used by the chain gun to produce erratic firing.

Inheritance diagram for TDGP.GunBarrelDelayedFire:



### Public Member Functions

- override void [OnFire](#) ()

*Requests bullet from gun clip and if returned, fires bullet based on barrels rotation and min and max fire delay.*

### Public Attributes

- float [MinFireDelay](#) = 0.05f

*The minimum delay between the fire button pressed and shooting a projectile.*

- float [MaxFireDelay](#) = 0.15f

*The maximum delay between the fire button pressed and shooting a projectile.*

### 5.12.1 Detailed Description

[Gun](#) barrel with delayed fire. Currently used by the chain gun to produce erratic firing.

### 5.12.2 Member Function Documentation

#### 5.12.2.1 `override void TDGP.GunBarrelDelayedFire.OnFire ( ) [virtual]`

Requests bullet from gun clip and if returned, fires bullet based on barrels rotation and min and max fire delay.

Reimplemented from [TDGP.GunBarrel](#).

### 5.12.3 Member Data Documentation

#### 5.12.3.1 `float TDGP.GunBarrelDelayedFire.MaxFireDelay = 0.15f`

The maximum delay between the fire button pressed and shooting a projectile.

#### 5.12.3.2 `float TDGP.GunBarrelDelayedFire.MinFireDelay = 0.05f`

The minimum delay between the fire button pressed and shooting a projectile.

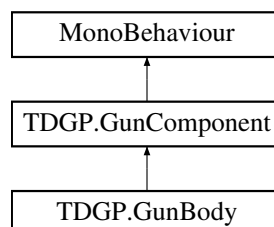
The documentation for this class was generated from the following file:

- `GunBarrelDelayedFire.cs`

## 5.13 TDGP.GunBody Class Reference

Responsibility: Updating the weapons relative position based on weapon type.

Inheritance diagram for TDGP.GunBody:



### Public Types

- enum **Gun\_Position** { **RIGHT\_ONE\_HANDED**, **RIGHT\_DUAL\_WIELD**, **LEFT\_DUAL\_WIELD**, **BOTH\_TWO\_HANDED** }

### Public Member Functions

- `override void OnPickup ()`  
*Called by gun component on parent. Sets local position and sprite sorting order.*
- `override void OnDrop ()`  
*Called by gun component on parent. Resets sprite sorting order.*

### Public Attributes

- Gun\_Position [GunPosition](#)  
*The guns relative position.*



### 5.13.1 Detailed Description

Responsibility: Updating the weapons relative position based on weapon type.

### 5.13.2 Member Function Documentation

#### 5.13.2.1 `override void TDGP.GunBody.OnDrop ( ) [virtual]`

Called by gun component on parent. Resets sprite sorting order.

Implements [TDGP.GunComponent](#).

#### 5.13.2.2 `override void TDGP.GunBody.OnPickup ( ) [virtual]`

Called by gun component on parent. Sets local position and sprite sorting order.

Implements [TDGP.GunComponent](#).

### 5.13.3 Member Data Documentation

#### 5.13.3.1 `Gun_Position` `TDGP.GunBody.GunPosition`

The guns relative position.

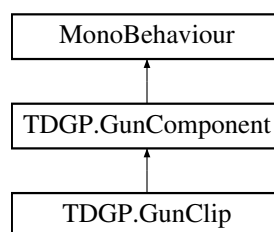
The documentation for this class was generated from the following file:

- `GunBody.cs`

## 5.14 TDGP.GunClip Class Reference

Responsibilities: act as a object pool for the guns projectiles, provide projectiles when requested by the guns barrel, limit the maximum number of bullets fired by the gun, and reload bullets when requested by player.

Inheritance diagram for TDGP.GunClip:



### Public Member Functions

- `void` [PoolObject](#) (`GameObject` obj)  
*Pools the object specified. Will not be pooled if there is no prefab of that type.*
- `GameObject` [GetBullet](#) ()  
*Request a bullet. Returns a bullet from the pool if present otherwise instantiates and returns new bullet.*
- `GameObject` [RequestBullet](#) ()  
*Requests a bullet. A bullet is only returned if: there are currently bullets in the clip, and the weapon is not currently being reloaded.*
- `override void` [OnPickup](#) ()

*Called by gun component on parent. Resets current bullets in clip.*

- override void [OnDrop](#) ( )

*Called by gun component on parent.*

## Public Attributes

- int [BulletsInClip](#) = 15

*The bullets in a clip.*

- bool [SupportReload](#) = false

*If true, this weapon can be reloaded.*

- float [ReloadSpeed](#) = 0.2f

*The time it takes to reload.*

### 5.14.1 Detailed Description

Responsibilities: act as a object pool for the guns projectiles, provide projectiles when requested by the guns barrel, limit the maximum number of bullets fired by the gun, and reload bullets when requested by player.

### 5.14.2 Member Function Documentation

#### 5.14.2.1 GameObject TDGP.GunClip.GetBullet ( )

Request a bullet. Returns a bullet from the pool if present totherwise instantiates and returns new bullet.

##### Returns

The bullet.

#### 5.14.2.2 override void TDGP.GunClip.OnDrop ( ) [virtual]

Called by gun component on parent.

Implements [TDGP.GunComponent](#).

#### 5.14.2.3 override void TDGP.GunClip.OnPickup ( ) [virtual]

Called by gun component on parent. Resets current bullets in clip.

Implements [TDGP.GunComponent](#).

#### 5.14.2.4 void TDGP.GunClip.PoolObject ( GameObject obj )

Pools the object specified. Will not be pooled if there is no prefab of that type.

##### Parameters

<i>obj</i>	Object to be pooled.
------------	----------------------

#### 5.14.2.5 GameObject TDGP.GunClip.RequestBullet ( )

Requests a bullet. A bullet is only returned if: there are currently bullets in the clip, and the weapon is not currently being reloaded.

**Returns**

The bullet.

**5.14.3 Member Data Documentation****5.14.3.1 int TDGP.GunClip.BulletsInClip = 15**

The bullets in a clip.

**5.14.3.2 float TDGP.GunClip.ReloadSpeed = 0.2f**

The time it takes to reload.

**5.14.3.3 bool TDGP.GunClip.SupportReload = false**

If true, this weapon can be reloaded.

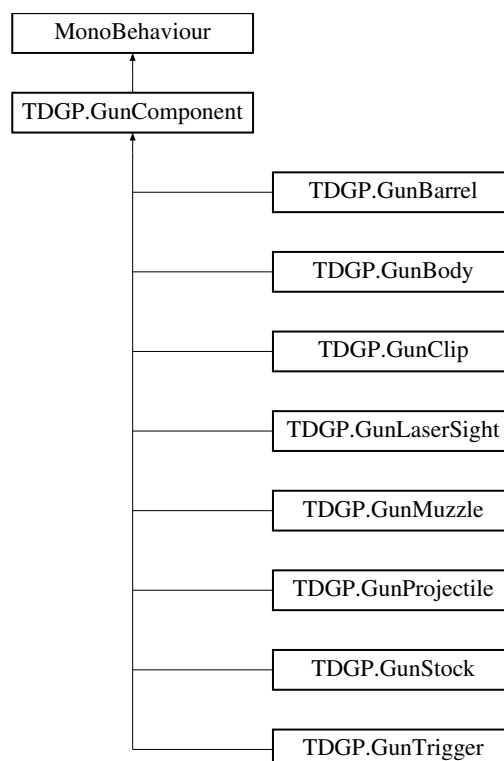
The documentation for this class was generated from the following file:

- GunClip.cs

**5.15 TDGP.GunComponent Class Reference**

The base class for all gun components.

Inheritance diagram for TDGP.GunComponent:



## Public Member Functions

- abstract void [OnPickup](#) ()  
*Raised when weapon picked up.*
- abstract void [OnDrop](#) ()  
*Raised when weapon dropped.*

### 5.15.1 Detailed Description

The base class for all gun components.

### 5.15.2 Member Function Documentation

#### 5.15.2.1 abstract void TDGP.GunComponent.OnDrop ( ) [pure virtual]

Raised when weapon dropped.

Implemented in [TDGP.GunClip](#), [TDGP.SeekingGunProjectile](#), [TDGP.GunProjectile](#), [TDGP.GunBarrel](#), [TDGP.GunTrigger](#), [TDGP.GunLaserSight](#), [TDGP.GunBody](#), [TDGP.GunMuzzle](#), and [TDGP.GunStock](#).

#### 5.15.2.2 abstract void TDGP.GunComponent.OnPickup ( ) [pure virtual]

Raised when weapon picked up.

Implemented in [TDGP.GunClip](#), [TDGP.SeekingGunProjectile](#), [TDGP.GunProjectile](#), [TDGP.GunBarrel](#), [TDGP.GunTriggerSingle](#), [TDGP.GunTrigger](#), [TDGP.GunLaserSight](#), [TDGP.GunBody](#), [TDGP.GunMuzzle](#), and [TDGP.GunStock](#).

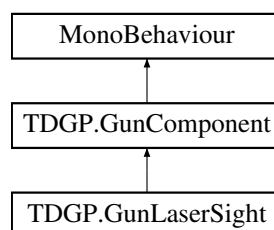
The documentation for this class was generated from the following file:

- GunComponent.cs

## 5.16 TDGP.GunLaserSight Class Reference

Adds a laser sight to a weapon. Responsible for updating the lasers scale.

Inheritance diagram for TDGP.GunLaserSight:



## Public Member Functions

- void **Awake** ()
- override void [OnPickup](#) ()  
*Raised when weapon picked up. Enables line renderer.*
- override void [OnDrop](#) ()  
*Raised when weapon dropped. Disables line renderer.*

## Public Attributes

- float [Range](#) = 5f  
*Total range of laser when no obstructions present.*

### 5.16.1 Detailed Description

Adds a laser sight to a weapon. Responsible for updating the lasers scale.

### 5.16.2 Member Function Documentation

#### 5.16.2.1 `override void TDGP.GunLaserSight.OnDrop ( ) [virtual]`

Raised when weapon dropped. Disables line renderer.

Implements [TDGP.GunComponent](#).

#### 5.16.2.2 `override void TDGP.GunLaserSight.OnPickup ( ) [virtual]`

Raised when weapon picked up. Enables line renderer.

Implements [TDGP.GunComponent](#).

### 5.16.3 Member Data Documentation

#### 5.16.3.1 `float TDGP.GunLaserSight.Range = 5f`

Total range of laser when no obstructions present.

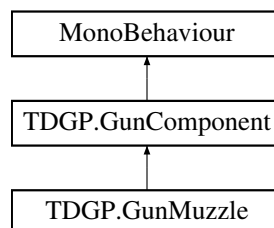
The documentation for this class was generated from the following file:

- GunLaserSight.cs

## 5.17 TDGP.GunMuzzle Class Reference

Handles gun muzzle flash.

Inheritance diagram for TDGP.GunMuzzle:



## Public Member Functions

- `override void OnPickup ( )`  
*Raised when weapon picked up.*
- `override void OnDrop ( )`  
*Raised when weapon dropped. Disables gameobject.*

## Public Attributes

- float [FlashAnimationSpeed](#) = 1f  
*The flash animation speed.*

### 5.17.1 Detailed Description

Handles gun muzzle flash.

### 5.17.2 Member Function Documentation

#### 5.17.2.1 override void TDGP.GunMuzzle.OnDrop ( ) [virtual]

Raised when weapon dropped. Disables gameobject.

Implements [TDGP.GunComponent](#).

#### 5.17.2.2 override void TDGP.GunMuzzle.OnPickup ( ) [virtual]

Raised when weapon picked up.

Implements [TDGP.GunComponent](#).

### 5.17.3 Member Data Documentation

#### 5.17.3.1 float TDGP.GunMuzzle.FlashAnimationSpeed = 1f

The flash animation speed.

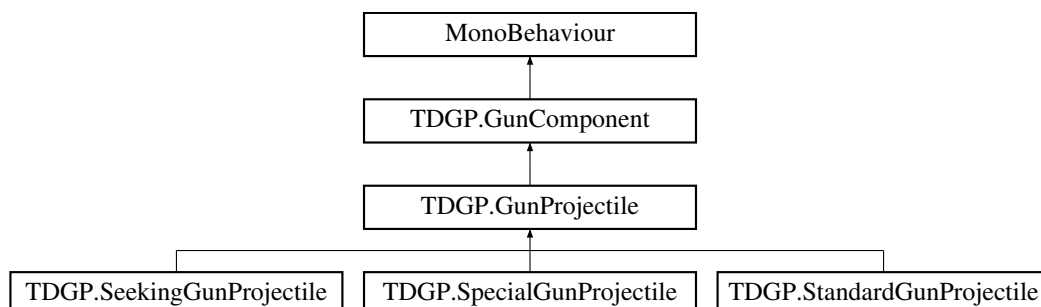
The documentation for this class was generated from the following file:

- GunMuzzle.cs

## 5.18 TDGP.GunProjectile Class Reference

The abstract base class for all gun projectiles. Responsible for handling collision, applying damage, and pooling projectile when max time alive has been reached.

Inheritance diagram for TDGP.GunProjectile:



## Public Member Functions

- virtual void **Awake** ()
- virtual void **Update** ()  
*Returns the projectile when/if maximum time alive is reached.*
- virtual void **OnTriggerEnter2D** (Collider2D other)  
*Returns projectile if it hits wall.*
- override void **OnPickup** ()  
*Raised when weapon picked up.*
- override void **OnDrop** ()  
*Raised when weapon dropped.*

## Public Attributes

- float **MaxTimeAlive** = 2f  
*The maximum time projectile can be alive. The projectile is pooled when this time has been reached.*
- bool **DestroyOnEnemyImpact** = true  
*If true, the projectile will be pooled when a collision with an enemy occurs.*

## Protected Member Functions

- void **ApplyDamage** (Collider2D other, float damage)
- void **InitDamageAnimation** (Collider2D other, GameObject animation)
- void **ReturnProjectile** ()

## Properties

- **GunClip Owner** [set]  
*The owner of this projectile.*

### 5.18.1 Detailed Description

The abstract base class for all gun projectiles. Responsible for handling collision, applying damage, and pooling projectile when max time alive has been reached.

### 5.18.2 Member Function Documentation

#### 5.18.2.1 override void TDGP.GunProjectile.OnDrop ( ) [virtual]

Raised when weapon dropped.

Implements [TDGP.GunComponent](#).

Reimplemented in [TDGP.SeekingGunProjectile](#).

#### 5.18.2.2 override void TDGP.GunProjectile.OnPickup ( ) [virtual]

Raised when weapon picked up.

Implements [TDGP.GunComponent](#).

Reimplemented in [TDGP.SeekingGunProjectile](#).

5.18.2.3 `virtual void TDGP.GunProjectile.OnTriggerEnter2D ( Collider2D other )` `[virtual]`

Returns projectile if it hits wall.



## Parameters

<i>other</i>	Other.
--------------	--------

Reimplemented in [TDGP.SeekingGunProjectile](#), [TDGP.StandardGunProjectile](#), and [TDGP.SpecialGunProjectile](#).

#### 5.18.2.4 virtual void TDGP.GunProjectile.Update ( ) [virtual]

Returns the projectile when/if maximum tile alive is reached.

Reimplemented in [TDGP.SeekingGunProjectile](#).

### 5.18.3 Member Data Documentation

#### 5.18.3.1 bool TDGP.GunProjectile.DestroyOnEnemyImpact = true

If true, the projectile will be pooled when a collision with an enemy occurs.

#### 5.18.3.2 float TDGP.GunProjectile.MaxTimeAlive = 2f

The maximum time projectile can be alive. The projectile is pooled when this time has been reached.

### 5.18.4 Property Documentation

#### 5.18.4.1 GunClip TDGP.GunProjectile.Owner [set]

The owner of this projectile.

The owner.

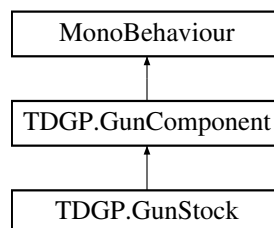
The documentation for this class was generated from the following file:

- GunProjectile.cs

## 5.19 TDGP.GunStock Class Reference

Responsibility: Providing a knock back force when weapon fired.

Inheritance diagram for TDGP.GunStock:



### Public Member Functions

- override void [OnPickup](#) ()  
*Raised when weapon picked up.*
- override void [OnDrop](#) ()  
*Raised when weapon dropped.*

## Public Attributes

- float `KnockBackForce` = 50f  
*The knock back force applied to the player on weapon fire.*

### 5.19.1 Detailed Description

Responsibility: Providing a knock back force when weapon fired.

### 5.19.2 Member Function Documentation

#### 5.19.2.1 `override void TDGP.GunStock.OnDrop ( ) [virtual]`

Raised when weapon dropped.

Implements [TDGP.GunComponent](#).

#### 5.19.2.2 `override void TDGP.GunStock.OnPickup ( ) [virtual]`

Raised when weapon picked up.

Implements [TDGP.GunComponent](#).

### 5.19.3 Member Data Documentation

#### 5.19.3.1 `float TDGP.GunStock.KnockBackForce = 50f`

The knock back force applied to the player on weapon fire.

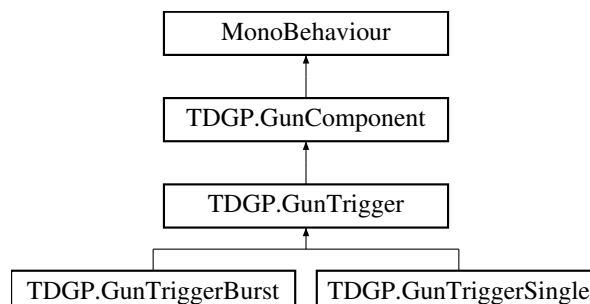
The documentation for this class was generated from the following file:

- GunStock.cs

## 5.20 TDGP.GunTrigger Class Reference

Abstract base class for all gun triggers. Responsibility: Handling player input.

Inheritance diagram for TDGP.GunTrigger:



## Public Types

- enum `TRIGGER_KEY_MAPPING` { `PRIMARY`, `SECONDARY` }
- enum `SHOOT_TYPE` { `CLICK`, `HOLD` }

## Public Member Functions

- virtual void [Awake](#) ()  
*Sets button mapping and retrieves list of barrels attached to gun.*
- abstract void [HandleClickInput](#) ()  
*Handles the click input.*
- abstract void [HandleHoldInput](#) ()  
*Handles the hold input.*
- override void [OnPickup](#) ()  
*Called by gun component on parent. Sets in use to true i.e. start listening for player input.*
- override void [OnDrop](#) ()  
*Called by gun component on parent. Sets in use to false i.e. stop listening for player input.*

## Public Attributes

- TRIGGER\_KEY\_MAPPING [KeyMapping](#)  
*The key mapping. Primary = right mouse click, secondary = left mouse click.*
- SHOOT\_TYPE [ShootType](#)  
*The shoot type i.e. click (the player has to keep clicking to shoot) or hold (the player can hold the mouse button down to continue to fire)*
- float [DelayBetweenProjectiles](#) = 0.1f  
*The delay between projectiles.*

## Protected Attributes

- string **buttonMapping**
- bool **inUse** = false
- List< [GunBarrel](#) > **barrels**

### 5.20.1 Detailed Description

Abstract base class for all gun triggers. Responsibility: Handling player input.

### 5.20.2 Member Function Documentation

#### 5.20.2.1 virtual void TDGP.GunTrigger.Awake ( ) [virtual]

Sets button mapping and retrieves list of barrels attached to gun.

Reimplemented in [TDGP.GunTriggerBurst](#), and [TDGP.GunTriggerSingle](#).

#### 5.20.2.2 abstract void TDGP.GunTrigger.HandleClickInput ( ) [pure virtual]

Handles the click input.

Implemented in [TDGP.GunTriggerSingle](#), and [TDGP.GunTriggerBurst](#).

#### 5.20.2.3 abstract void TDGP.GunTrigger.HandleHoldInput ( ) [pure virtual]

Handles the hold input.

Implemented in [TDGP.GunTriggerSingle](#), and [TDGP.GunTriggerBurst](#).

#### 5.20.2.4 override void TDGP.GunTrigger.OnDrop ( ) [virtual]

Called by gun component on parent. Sets in use to false i.e. stop listening for player input.

Implements [TDGP.GunComponent](#).

#### 5.20.2.5 override void TDGP.GunTrigger.OnPickup ( ) [virtual]

Called by gun component on parent. Sets in use to true i.e. start listening for player input.

Implements [TDGP.GunComponent](#).

Reimplemented in [TDGP.GunTriggerSingle](#).

### 5.20.3 Member Data Documentation

#### 5.20.3.1 float TDGP.GunTrigger.DelayBetweenProjectiles = 0.1f

The delay between projectiles.

#### 5.20.3.2 TRIGGER\_KEY\_MAPPING TDGP.GunTrigger.KeyMapping

The key mapping. Primary = right mouse click, secondary = left mouse click.

#### 5.20.3.3 SHOOT\_TYPE TDGP.GunTrigger.ShootType

The shoot type i.e. click (the player has to keep clicking to shoot) or hold (the player can hold the mouse button down to continue to fire)

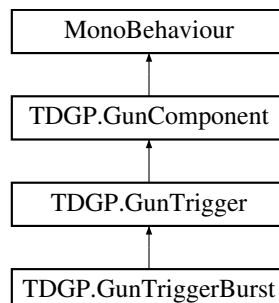
The documentation for this class was generated from the following file:

- GunTrigger.cs

## 5.21 TDGP.GunTriggerBurst Class Reference

Provides burst firing.

Inheritance diagram for TDGP.GunTriggerBurst:



### Public Member Functions

- override void [Awake](#) ()  
*Sets button mapping and retrieves list of barrels attached to gun.*

- override void [HandleClickInput](#) ()  
*If not already firing burst, a burst of projectiles will be fired.*
- override void [HandleHoldInput](#) ()  
*If not already firing burst, a burst of projectiles will be fired.*

## Public Attributes

- float [TimeBetweenBursts](#) = 0.3f  
*The time between bursts of projectiles.*
- int [BulletsPerBurst](#) = 3  
*The bullets per burst.*

## Additional Inherited Members

### 5.21.1 Detailed Description

Provides burst firing.

### 5.21.2 Member Function Documentation

#### 5.21.2.1 override void TDGP.GunTriggerBurst.Awake ( ) [virtual]

Sets button mapping and retrieves list of barrels attached to gun.

Reimplemented from [TDGP.GunTrigger](#).

#### 5.21.2.2 override void TDGP.GunTriggerBurst.HandleClickInput ( ) [virtual]

If not already firing burst, a burst of projectiles will be fired.

Implements [TDGP.GunTrigger](#).

#### 5.21.2.3 override void TDGP.GunTriggerBurst.HandleHoldInput ( ) [virtual]

If not already firing burst, a burst of projectiles will be fired.

Implements [TDGP.GunTrigger](#).

### 5.21.3 Member Data Documentation

#### 5.21.3.1 int TDGP.GunTriggerBurst.BulletsPerBurst = 3

The bullets per burst.

#### 5.21.3.2 float TDGP.GunTriggerBurst.TimeBetweenBursts = 0.3f

The time between bursts of projectiles.

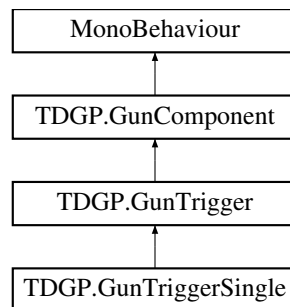
The documentation for this class was generated from the following file:

- [GunTriggerBurst.cs](#)

## 5.22 TDGP.GunTriggerSingle Class Reference

Provides signle bullet firing on trigger press.

Inheritance diagram for TDGP.GunTriggerSingle:



### Public Member Functions

- override void [Awake](#) ()  
*Sets button mapping and retrieves list of barrels attached to gun.*
- override void [HandleClickInput](#) ()  
*Tells each barrel attached to the gun to fire.*
- override void [HandleHoldInput](#) ()  
*Tells each barrel attached to the gun to fire.*
- override void [OnPickup](#) ()  
*Called by gun component on parent. Resets current shoot speed.*

### Additional Inherited Members

#### 5.22.1 Detailed Description

Provides signle bullet firing on trigger press.

#### 5.22.2 Member Function Documentation

##### 5.22.2.1 override void TDGP.GunTriggerSingle.Awake ( ) [virtual]

Sets button mapping and retrieves list of barrels attached to gun.

Reimplemented from [TDGP.GunTrigger](#).

##### 5.22.2.2 override void TDGP.GunTriggerSingle.HandleClickInput ( ) [virtual]

Tells each barrel attached to the gun to fire.

Implements [TDGP.GunTrigger](#).

##### 5.22.2.3 override void TDGP.GunTriggerSingle.HandleHoldInput ( ) [virtual]

Tells each barrel attached to the gun to fire.

Implements [TDGP.GunTrigger](#).

#### 5.22.2.4 override void TDGP.GunTriggerSingle.OnPickup ( ) [virtual]

Called by gun component on parent. Resets current shoot speed.

Reimplemented from [TDGP.GunTrigger](#).

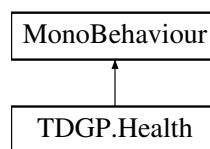
The documentation for this class was generated from the following file:

- GunTriggerSingle.cs

## 5.23 TDGP.Health Class Reference

Enemy health script.

Inheritance diagram for TDGP.Health:



### Public Member Functions

- void [OnHit](#) (float damageAmount)  
*Reduces health, plays hit sound, and kills enemy if health less than or equal to zero.*
- void [ApplyDPS](#) (float dps, float time)  
*Applies damage per second.*

### Public Attributes

- float [MaxHealth](#) = 10f  
*The starting health of enemies in the demo scene.*
- AudioClip[] [OnHitSounds](#)  
*Sound pool of possible sounds to play when hit.*
- GameObject [OnDeadAnimation](#)  
*Animation to play when enemy killed.*
- GameObject[] [OnDeadSprites](#)  
*Pool of sprites to place when enemy killed.*

### 5.23.1 Detailed Description

Enemy health script.

### 5.23.2 Member Function Documentation

#### 5.23.2.1 void TDGP.Health.ApplyDPS ( float dps, float time )

Applies damage per second.

## Parameters

<i>dps</i>	Damage per second.
<i>time</i>	Time.

5.23.2.2 void TDGP.Health.OnHit ( float *damageAmount* )

Reduces health, plays hit sound, and kills enemy if health less than or equal to zero.

## Parameters

<i>damageAmount</i>	Damage amount.
---------------------	----------------

## 5.23.3 Member Data Documentation

## 5.23.3.1 float TDGP.Health.MaxHealth = 10f

The starting health of enemies in the demo scene.

## 5.23.3.2 GameObject TDGP.Health.OnDeadAnimation

Animation to play when enemy killed.

## 5.23.3.3 GameObject [ ] TDGP.Health.OnDeadSprites

Pool of sprites to place when enemy killed.

## 5.23.3.4 AudioClip [ ] TDGP.Health.OnHitSounds

Sound pool of possible sounds to play when hit.

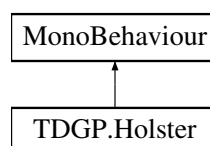
The documentation for this class was generated from the following file:

- Health.cs

## 5.24 TDGP.Holster Class Reference

Weapon pickup script, also allows for weapon switching. Used for the demo scene but can be adapted for use by your player.

Inheritance diagram for TDGP.Holster:



## Public Member Functions

- void [OnWeaponPickup](#) (Transform weapon)

*Called when a weapon is picked up. Attaches weapon to player in a holster slot. If the holster is full it destroys the weapon in the current slot.*



### 5.24.1 Detailed Description

Weapon pickup script, also allows for weapon switching. Used for the demo scene but can be adapted for use by your player.

### 5.24.2 Member Function Documentation

#### 5.24.2.1 void TDGP.Holster.OnWeaponPickup ( Transform *weapon* )

Called when a weapon is picked up. Attaches weapon to player in a holster slot. If the holster is full it destroys the weapon in the current slot.

Parameters

<i>weapon</i>	Weapon.
---------------	---------

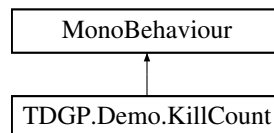
The documentation for this class was generated from the following file:

- Holster.cs

## 5.25 TDGP.Demo.KillCount Class Reference

Shows text on screen for number of enemies killed in demo scene.

Inheritance diagram for TDGP.Demo.KillCount:



### Public Member Functions

- void **EnemyKilled** ()

### 5.25.1 Detailed Description

Shows text on screen for number of enemies killed in demo scene.

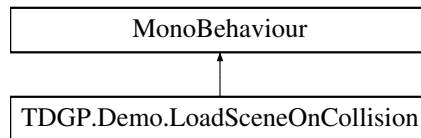
The documentation for this class was generated from the following file:

- KillCount.cs

## 5.26 TDGP.Demo.LoadSceneOnCollision Class Reference

Reloads scene if enemy touches player in demo scene.

Inheritance diagram for TDGP.Demo.LoadSceneOnCollision:



### Public Attributes

- int **SceneNumber** = 0

#### 5.26.1 Detailed Description

Reloads scene if enemy touches player in demo scene.

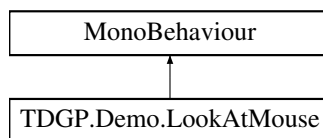
The documentation for this class was generated from the following file:

- LoadSceneOnCollision.cs

### 5.27 TDGP.Demo.LookAtMouse Class Reference

[Demo](#) script. Play looks at mouse.

Inheritance diagram for TDGP.Demo.LookAtMouse:



#### 5.27.1 Detailed Description

[Demo](#) script. Play looks at mouse.

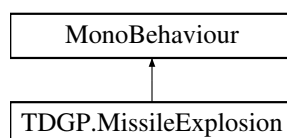
The documentation for this class was generated from the following file:

- LookAtMouse.cs

### 5.28 TDGP.MissileExplosion Class Reference

Missile explosion. Applies knockback force to enemy on explosion.

Inheritance diagram for TDGP.MissileExplosion:



## Public Attributes

- float **PushBackForce** = 10f

### 5.28.1 Detailed Description

Missile explosion. Applies knockback force to enemy on explosion.

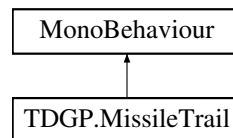
The documentation for this class was generated from the following file:

- MissileExplosion.cs

## 5.29 TDGP.MissileTrail Class Reference

Missile trail. Uses line renderer to create a trail effect for the missile.

Inheritance diagram for TDGP.MissileTrail:



## Public Member Functions

- void **OnFire** ()  
*Updates line based on projectiles movement.*

## Public Attributes

- int **numOfPoints** = 10  
*The number of points on line.*
- float **updateSpeed** = 0.25f  
*The update speed.*
- float **turnSpeed** = 0.25f  
*The turn speed.*
- float **spread** = 0.2f  
*The spread of the line.*

### 5.29.1 Detailed Description

Missile trail. Uses line renderer to create a trail effect for the missile.

### 5.29.2 Member Function Documentation

#### 5.29.2.1 void TDGP.MissileTrail.OnFire ( )

Updates line based on projectiles movement.

### 5.29.3 Member Data Documentation

#### 5.29.3.1 `int TDGP.MissileTrail.numOfPoints = 10`

The number of points on line.

#### 5.29.3.2 `float TDGP.MissileTrail.spread = 0.2f`

The spread of the line.

#### 5.29.3.3 `float TDGP.MissileTrail.turnSpeed = 0.25f`

The turn speed.

#### 5.29.3.4 `float TDGP.MissileTrail.updateSpeed = 0.25f`

The update speed.

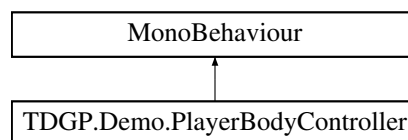
The documentation for this class was generated from the following file:

- `MissileTrail.cs`

## 5.30 TDGP.Demo.PlayerBodyController Class Reference

Updates player sprite based on currently held weapon.

Inheritance diagram for `TDGP.Demo.PlayerBodyController`:



### Public Member Functions

- void **PickedUpOneHanded** ()
- void **PickedUpTwoHandedWeapon** ()
- void **PickedUpDualWieldWeapon** ()

### Public Attributes

- Sprite **OneHandedWeaponBody**
- Sprite **TwoHandedWeaponBody**
- Sprite **DualWieldWeaponBody**

#### 5.30.1 Detailed Description

Updates player sprite based on currently held weapon.

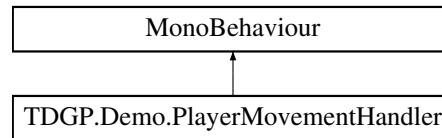
The documentation for this class was generated from the following file:

- `PlayerBodyController.cs`

## 5.31 TDGP.Demo.PlayerMovementHandler Class Reference

[Demo](#) script. Updates players position based on input.

Inheritance diagram for TDGP.Demo.PlayerMovementHandler:



### Public Attributes

- float **MoveSpeed** = 2.5f

#### 5.31.1 Detailed Description

[Demo](#) script. Updates players position based on input.

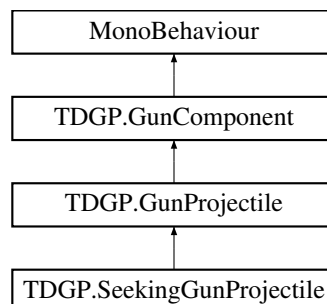
The documentation for this class was generated from the following file:

- PlayerMovementHandler.cs

## 5.32 TDGP.SeekingGunProjectile Class Reference

A projectile that seeks towards its target.

Inheritance diagram for TDGP.SeekingGunProjectile:



### Public Member Functions

- override void **Awake** ()
- override void [OnTriggerEnter2D](#) (Collider2D other)  
*Returns projectile if it hits wall or enemy. Instantiates damage animation and damages object (if enemy).*
- override void [Update](#) ()  
*Rotates towards closest target.*
- Vector2 **GetForce** ()
- override void [OnPickup](#) ()  
*Raised when weapon picked up.*
- override void [OnDrop](#) ()  
*Raised when weapon dropped.*

## Public Attributes

- float [Damage](#) = 10f  
*Damage on collision.*
- float [Velocity](#) = 40f  
*Projectiles velocity.*
- GameObject [DamageAnimation](#)  
*Animation on collision with enemy.*

## Additional Inherited Members

### 5.32.1 Detailed Description

A projectile that seeks towards its target.

### 5.32.2 Member Function Documentation

#### 5.32.2.1 override void TDGP.SeekingGunProjectile.OnDrop ( ) [virtual]

Raised when weapon dropped.

Reimplemented from [TDGP.GunProjectile](#).

#### 5.32.2.2 override void TDGP.SeekingGunProjectile.OnPickup ( ) [virtual]

Raised when weapon picked up.

Reimplemented from [TDGP.GunProjectile](#).

#### 5.32.2.3 override void TDGP.SeekingGunProjectile.OnTriggerEnter2D ( Collider2D *other* ) [virtual]

Returns projectile if it hits wall or enemy. Instantiates damage animation and damages object (if enemy).

Parameters

<i>other</i>	Other.
--------------	--------

Reimplemented from [TDGP.GunProjectile](#).

#### 5.32.2.4 override void TDGP.SeekingGunProjectile.Update ( ) [virtual]

Rotates towards closest target.

Reimplemented from [TDGP.GunProjectile](#).

### 5.32.3 Member Data Documentation

#### 5.32.3.1 float TDGP.SeekingGunProjectile.Damage = 10f

Damage on collision.

#### 5.32.3.2 GameObject TDGP.SeekingGunProjectile.DamageAnimation

Animation on collision with enemy.

## 5.32.3.3 float TDGP.SeekGunProjectile.Velocity = 40f

Projectiles velocity.

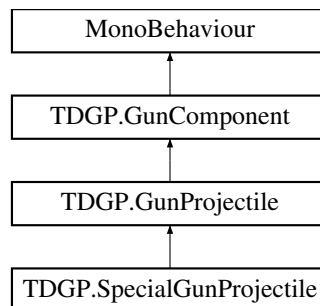
The documentation for this class was generated from the following file:

- SeekingGunProjectile.cs

## 5.33 TDGP.SpecialGunProjectile Class Reference

Projectile of special type i.e. ice or chain lightning.

Inheritance diagram for TDGP.SpecialGunProjectile:



### Public Member Functions

- override void [OnTriggerEnter2D](#) (Collider2D other)  
*Returns projectile if it hits wall. Applies special damage if hits enemy.*

### Public Attributes

- float [DamagePerSecond](#) = 0.5f  
*The damage per second.*
- float [DamageTime](#) = 1f  
*The damage time.*
- [SPECIAL\\_DAMAGE\\_TYPE](#) DamageType  
*The type of damage to apply.*

### Additional Inherited Members

#### 5.33.1 Detailed Description

Projectile of special type i.e. ice or chain lightning.

#### 5.33.2 Member Function Documentation

##### 5.33.2.1 override void TDGP.SpecialGunProjectile.OnTriggerEnter2D ( Collider2D other ) [virtual]

Returns projectile if it hits wall. Applies special damage if hits enemy.

## Parameters

<i>other</i>	Other.
--------------	--------

Reimplemented from [TDGP.GunProjectile](#).

### 5.33.3 Member Data Documentation

#### 5.33.3.1 float TDGP.SpecialGunProjectile.DamagePerSecond = 0.5f

The damage per second.

#### 5.33.3.2 float TDGP.SpecialGunProjectile.DamageTime = 1f

The damage time.

#### 5.33.3.3 SPECIAL\_DAMAGE\_TYPE TDGP.SpecialGunProjectile.DamageType

The type of damage to apply.

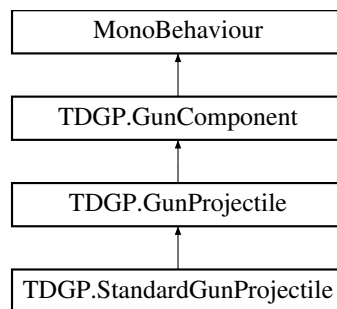
The documentation for this class was generated from the following file:

- [SpecialGunProjectile.cs](#)

## 5.34 TDGP.StandardGunProjectile Class Reference

A standard gun projectile.

Inheritance diagram for TDGP.StandardGunProjectile:



### Public Member Functions

- override void **Awake** ()
- override void [OnTriggerEnter2D](#) (Collider2D other)  
*Returns projectile if it hits wall. Applies damage if hits enemy.*

### Public Attributes

- float [Damage](#)  
*Damage on projectile hit.*
- GameObject [DamageAnimation](#)  
*The damage animation prefab.*



## Additional Inherited Members

### 5.34.1 Detailed Description

A standard gun projectile.

### 5.34.2 Member Function Documentation

5.34.2.1 `override void TDGP.StandardGunProjectile.OnTriggerEnter2D ( Collider2D other )` `[virtual]`

Returns projectile if it hits wall. Applys damage if hits enemy.

Parameters

<i>other</i>	Other.
--------------	--------

Reimplemented from [TDGP.GunProjectile](#).

### 5.34.3 Member Data Documentation

5.34.3.1 `float TDGP.StandardGunProjectile.Damage`

Damage on projectile hit.

5.34.3.2 `GameObject TDGP.StandardGunProjectile.DamageAnimation`

The damage animation prefab.

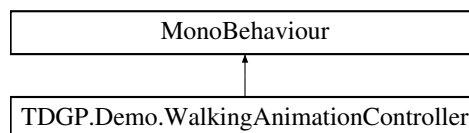
The documentation for this class was generated from the following file:

- StandardGunProjectile.cs

## 5.35 TDGP.Demo.WalkingAnimationController Class Reference

Updates walking animation based on movement speed.

Inheritance diagram for TDGP.Demo.WalkingAnimationController:



### 5.35.1 Detailed Description

Updates walking animation based on movement speed.

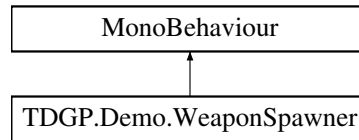
The documentation for this class was generated from the following file:

- WalkingAnimationController.cs

## 5.36 TDGP.Demo.WeaponSpawner Class Reference

**Demo** Script. Spawns weapon at players location on button press.

Inheritance diagram for TDGP.Demo.WeaponSpawner:



### Public Attributes

- Transform **Player**
- GameObject[] **WeaponPrefabs**

### 5.36.1 Detailed Description

**Demo** Script. Spawns weapon at players location on button press.

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

- WeaponSpawner.cs

# Index

- ApplyDPS
  - TDGP::Health, [37](#)
- ApplyDamage
  - TDGP::DamageAnimationController, [13](#)
- AudioClips
  - TDGP::AnimationAudio, [12](#)
- Awake
  - TDGP::GunTrigger, [33](#)
  - TDGP::GunTriggerBurst, [35](#)
  - TDGP::GunTriggerSingle, [36](#)
- BulletsInClip
  - TDGP::GunClip, [25](#)
- BulletsPerBurst
  - TDGP::GunTriggerBurst, [35](#)
- CanMove
  - TDGP::EnemyMovement, [16](#)
- ChainLightningRange
  - TDGP::DamageAnimationController, [13](#)
- Damage
  - TDGP::SeekingGunProjectile, [44](#)
  - TDGP::StandardGunProjectile, [47](#)
- DamageAnimation
  - TDGP::SeekingGunProjectile, [44](#)
  - TDGP::StandardGunProjectile, [47](#)
- DamagePerSecond
  - TDGP::SpecialGunProjectile, [46](#)
- DamageTime
  - TDGP::SpecialGunProjectile, [46](#)
- DamageType
  - TDGP::SpecialGunProjectile, [46](#)
- DelayBetweenProjectiles
  - TDGP::GunTrigger, [34](#)
- DestroyOnEnemyImpact
  - TDGP::GunProjectile, [31](#)
- EnableMovement
  - TDGP::EnableParentMovement, [15](#)
- Enemy
  - TDGP::Demo::EnemySpawner, [17](#)
- EnemyRemoved
  - TDGP::Demo::EnemySpawner, [17](#)
- Execute
  - TDGP::Destroy, [14](#)
  - TDGP::Disable, [14](#)
- FireSounds
  - TDGP::GunBarrel, [21](#)
- FlashAnimationSpeed
  - TDGP::GunMuzzle, [28](#)
- GetBullet
  - TDGP::GunClip, [24](#)
- Gun\_Type
  - TDGP, [8](#)
- GunPosition
  - TDGP::GunBody, [23](#)
- GunType
  - TDGP::Gun, [19](#)
- HandleClickInput
  - TDGP::GunTrigger, [33](#)
  - TDGP::GunTriggerBurst, [35](#)
  - TDGP::GunTriggerSingle, [36](#)
- HandleHoldInput
  - TDGP::GunTrigger, [33](#)
  - TDGP::GunTriggerBurst, [35](#)
  - TDGP::GunTriggerSingle, [36](#)
- IceCaseAnimation
  - TDGP::DamageAnimationController, [13](#)
- InversePlayChance
  - TDGP::AnimationAudio, [12](#)
- KeyMapping
  - TDGP::GunTrigger, [34](#)
- KillCount
  - TDGP::Demo::EnemySpawner, [17](#)
- KnockBackForce
  - TDGP::GunStock, [32](#)
- LaunchSpeed
  - TDGP::GunBarrel, [21](#)
- LightningAnimation
  - TDGP::DamageAnimationController, [13](#)
- MaxChainLightningEnemiesHit
  - TDGP::DamageAnimationController, [13](#)
- MaxEnemiesOnScreen
  - TDGP::Demo::EnemySpawner, [17](#)
- MaxFireDelay
  - TDGP::GunBarrelDelayedFire, [22](#)
- MaxHealth
  - TDGP::Health, [38](#)
- MaxTimeAlive
  - TDGP::GunProjectile, [31](#)
- MinFireDelay
  - TDGP::GunBarrelDelayedFire, [22](#)
- MoveSpeed
  - TDGP::EnemyMovement, [16](#)

- numOfPoints
  - TDGP::MissileTrail, [42](#)
- OnDeadAnimation
  - TDGP::Health, [38](#)
- OnDeadSprites
  - TDGP::Health, [38](#)
- OnDrop
  - TDGP::Gun, [19](#)
  - TDGP::GunBarrel, [20](#)
  - TDGP::GunBody, [23](#)
  - TDGP::GunClip, [24](#)
  - TDGP::GunComponent, [26](#)
  - TDGP::GunLaserSight, [27](#)
  - TDGP::GunMuzzle, [28](#)
  - TDGP::GunProjectile, [29](#)
  - TDGP::GunStock, [32](#)
  - TDGP::GunTrigger, [33](#)
  - TDGP::SeekingGunProjectile, [44](#)
- OnFire
  - TDGP::GunBarrel, [20](#)
  - TDGP::GunBarrelDelayedFire, [22](#)
  - TDGP::MissileTrail, [41](#)
- OnHit
  - TDGP::Health, [38](#)
- OnHitSounds
  - TDGP::Health, [38](#)
- OnPickup
  - TDGP::Gun, [19](#)
  - TDGP::GunBarrel, [20](#)
  - TDGP::GunBody, [23](#)
  - TDGP::GunClip, [24](#)
  - TDGP::GunComponent, [26](#)
  - TDGP::GunLaserSight, [27](#)
  - TDGP::GunMuzzle, [28](#)
  - TDGP::GunProjectile, [29](#)
  - TDGP::GunStock, [32](#)
  - TDGP::GunTrigger, [34](#)
  - TDGP::GunTriggerSingle, [36](#)
  - TDGP::SeekingGunProjectile, [44](#)
- OnTriggerEnter2D
  - TDGP::GunProjectile, [29](#)
  - TDGP::SeekingGunProjectile, [44](#)
  - TDGP::SpecialGunProjectile, [45](#)
  - TDGP::StandardGunProjectile, [47](#)
- OnWeaponPickup
  - TDGP::Holster, [39](#)
- Owner
  - TDGP::GunProjectile, [31](#)
- PlaySound
  - TDGP::AnimationAudio, [11](#)
- PoolObject
  - TDGP::GunClip, [24](#)
- Range
  - TDGP::GunLaserSight, [27](#)
- ReloadSpeed
  - TDGP::GunClip, [25](#)
- RequestBullet
  - TDGP::GunClip, [24](#)
- SPECIAL\_DAMAGE\_TYPE
  - TDGP, [8](#)
- ShootType
  - TDGP::GunTrigger, [34](#)
- SpawnComplete
  - TDGP::EnemyMovement, [16](#)
- SpawnTime
  - TDGP::Demo::EnemySpawner, [17](#)
- spread
  - TDGP::MissileTrail, [42](#)
- SupportReload
  - TDGP::GunClip, [25](#)
- TDGP, [7](#)
  - Gun\_Type, [8](#)
  - SPECIAL\_DAMAGE\_TYPE, [8](#)
- TDGP.AnimationAudio, [11](#)
- TDGP.DamageAnimationController, [12](#)
- TDGP.Demo, [9](#)
- TDGP.Demo.EnemySpawner, [16](#)
- TDGP.Demo.Environment, [17](#)
- TDGP.Demo.FollowTarget, [18](#)
- TDGP.Demo.KillCount, [39](#)
- TDGP.Demo.LoadSceneOnCollision, [39](#)
- TDGP.Demo.LookAtMouse, [40](#)
- TDGP.Demo.PlayerBodyController, [42](#)
- TDGP.Demo.PlayerMovementHandler, [43](#)
- TDGP.Demo.WalkingAnimationController, [47](#)
- TDGP.Demo.WeaponSpawner, [48](#)
- TDGP.Destroy, [13](#)
- TDGP.Disable, [14](#)
- TDGP.EnableParentMovement, [14](#)
- TDGP.EnemyMovement, [15](#)
- TDGP.Gun, [18](#)
- TDGP.GunBarrel, [19](#)
- TDGP.GunBarrelDelayedFire, [21](#)
- TDGP.GunBody, [22](#)
- TDGP.GunClip, [23](#)
- TDGP.GunComponent, [25](#)
- TDGP.GunLaserSight, [26](#)
- TDGP.GunMuzzle, [27](#)
- TDGP.GunProjectile, [28](#)
- TDGP.GunStock, [31](#)
- TDGP.GunTrigger, [32](#)
- TDGP.GunTriggerBurst, [34](#)
- TDGP.GunTriggerSingle, [36](#)
- TDGP.Health, [37](#)
- TDGP.Holster, [38](#)
- TDGP.MissileExplosion, [40](#)
- TDGP.MissileTrail, [41](#)
- TDGP.SeekingGunProjectile, [43](#)
- TDGP.SpecialGunProjectile, [45](#)
- TDGP.StandardGunProjectile, [46](#)
- TDGP::AnimationAudio
  - AudioClips, [12](#)
  - InversePlayChance, [12](#)

- PlaySound, 11
- VolumeScale, 12
- TDGP::DamageAnimationController
  - ApplyDamage, 13
  - ChainLightningRange, 13
  - IceCaseAnimation, 13
  - LightningAnimation, 13
  - MaxChainLightningEnemiesHit, 13
- TDGP::Demo::EnemySpawner
  - Enemy, 17
  - EnemyRemoved, 17
  - KillCount, 17
  - MaxEnemiesOnScreen, 17
  - SpawnTime, 17
- TDGP::Destroy
  - Execute, 14
- TDGP::Disable
  - Execute, 14
- TDGP::EnableParentMovement
  - EnableMovement, 15
- TDGP::EnemyMovement
  - CanMove, 16
  - MoveSpeed, 16
  - SpawnComplete, 16
- TDGP::Gun
  - GunType, 19
  - OnDrop, 19
  - OnPickup, 19
- TDGP::GunBarrel
  - FireSounds, 21
  - LaunchSpeed, 21
  - OnDrop, 20
  - OnFire, 20
  - OnPickup, 20
  - VolumeScale, 21
- TDGP::GunBarrelDelayedFire
  - MaxFireDelay, 22
  - MinFireDelay, 22
  - OnFire, 22
- TDGP::GunBody
  - GunPosition, 23
  - OnDrop, 23
  - OnPickup, 23
- TDGP::GunClip
  - BulletsInClip, 25
  - GetBullet, 24
  - OnDrop, 24
  - OnPickup, 24
  - PoolObject, 24
  - ReloadSpeed, 25
  - RequestBullet, 24
  - SupportReload, 25
- TDGP::GunComponent
  - OnDrop, 26
  - OnPickup, 26
- TDGP::GunLaserSight
  - OnDrop, 27
  - OnPickup, 27
- Range, 27
- TDGP::GunMuzzle
  - FlashAnimationSpeed, 28
  - OnDrop, 28
  - OnPickup, 28
- TDGP::GunProjectile
  - DestroyOnEnemyImpact, 31
  - MaxTimeAlive, 31
  - OnDrop, 29
  - OnPickup, 29
  - OnTriggerEnter2D, 29
  - Owner, 31
  - Update, 31
- TDGP::GunStock
  - KnockBackForce, 32
  - OnDrop, 32
  - OnPickup, 32
- TDGP::GunTrigger
  - Awake, 33
  - DelayBetweenProjectiles, 34
  - HandleClickInput, 33
  - HandleHoldInput, 33
  - KeyMapping, 34
  - OnDrop, 33
  - OnPickup, 34
  - ShootType, 34
- TDGP::GunTriggerBurst
  - Awake, 35
  - BulletsPerBurst, 35
  - HandleClickInput, 35
  - HandleHoldInput, 35
  - TimeBetweenBursts, 35
- TDGP::GunTriggerSingle
  - Awake, 36
  - HandleClickInput, 36
  - HandleHoldInput, 36
  - OnPickup, 36
- TDGP::Health
  - ApplyDPS, 37
  - MaxHealth, 38
  - OnDeadAnimation, 38
  - OnDeadSprites, 38
  - OnHit, 38
  - OnHitSounds, 38
- TDGP::Holster
  - OnWeaponPickup, 39
- TDGP::MissileTrail
  - numOfPoints, 42
  - OnFire, 41
  - spread, 42
  - turnSpeed, 42
  - updateSpeed, 42
- TDGP::SeekingGunProjectile
  - Damage, 44
  - DamageAnimation, 44
  - OnDrop, 44
  - OnPickup, 44
  - OnTriggerEnter2D, 44

- Update, [44](#)
- Velocity, [44](#)
- TDGP::SpecialGunProjectile
  - DamagePerSecond, [46](#)
  - DamageTime, [46](#)
  - DamageType, [46](#)
  - OnTriggerEnter2D, [45](#)
- TDGP::StandardGunProjectile
  - Damage, [47](#)
  - DamageAnimation, [47](#)
  - OnTriggerEnter2D, [47](#)
- TimeBetweenBursts
  - TDGP::GunTriggerBurst, [35](#)
- turnSpeed
  - TDGP::MissileTrail, [42](#)
- Update
  - TDGP::GunProjectile, [31](#)
  - TDGP::SeekingGunProjectile, [44](#)
- updateSpeed
  - TDGP::MissileTrail, [42](#)
- Velocity
  - TDGP::SeekingGunProjectile, [44](#)
- VolumeScale
  - TDGP::AnimationAudio, [12](#)
  - TDGP::GunBarrel, [21](#)