Pew Pew

1

v1.0.

Contents

Chapter 1

Namespace Index

1.1 Packages

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

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

Namespace Documentation

4.1 GameCore Namespace Reference

Classes

- interface AdjustableMoveSpeed

  Contract for any object that has an adjustable move speed.
- interface AdjustableShootSpeed

  Contract for any object that has an adjustable shoot speed.
- class AudioPlayer

  Attach to an audio source. Used to play instances of AudioClip.
- class AudioToggle

  Toggles game audio on button press.
- class BGMAudioPlayer

  Handles playing background music. Can fade between clips and queue pitch and volume lerp.
- class Blackhole

  Attached to a blackhole or repel. Controls health.
- class Bomb

  Responsible for animating the bomb and damaging enemies on explosion.
- interface BombListener

  Contract for any entity that can take damage from a bomb.
- class BombManager

  Spawns bomb within set bounds when player taps on screen.
- class BonusScorePowerUp

  Provides bonus particles.
- interface BossPart

  Contract for all boss parts.
- class BossPartDirectional

  Controls the directional boss part.
- class BossPartDropDown

  Controls the boss part drop down.
- class BossPartImpl

  Abstract concrete implementation of BossPart.
- class BossPartQuick

  Controls the boss part quick.
- class BossPartSeperateShip

Controls the boss part seperate ship.

- class BossPartShoot

  Controls the boss part shoot.
- class BossPartTop

  Controls the boss top part.
- class ButtonAnimationController

  Holds collections of ButtonAnimator. Responsible for playing the button aniamtions, with a small delay between each row.
- class ButtonAnimator

  Triggers individual shop button animations.
- class CameraShake

  Camera shake based on perlin noise.
- class ChallengeEnemyOnDeath

  Handles animation and particle emission on challenge enemies death.
- class ChallengeMovement

  Controls the challenge enemies movement.
- class ClassicMovement

  Controls the classic enemy movement.
- class CoroutineHandler

  Provides access to coroutines for instances that do not inherit form monobehaviour.
- class DamageEnemies

  Damages enemies on trigger enter.
- class DamagePlayer

  Damages player on trigger enter.
- class DirectionalMovement

  Controls the directional enemies.
- class DisableEffectWhenAnotherEffectorInScene

  Disables this particle effector when another effector is present in scene.
- class DoubleShotPowerUp

  Provides ability for player to shoot two projectiles in parallel for a specified amount of time.
- class DropDownMovement

  Controls the drop down enemymovement.
- class DropPowerUpOnDeath

  Spawns a powerup when this an entity with this script attached dies.
- class EnemyHealth

  Controls enemies health and taking damage.
- interface EnemyMove

  Contract for all enemies that can begin, pause, and resume actions.
- class EnemyMovement

  Controls standard enemy movement.
- class EnemyMoveReceiver

  Attach to an enemy object to receive moves from VerticalGroupMovement.
- class EnemyMoveRegister

  Adds attached EnemyMove to GameManager::EnemyMoves.
- class EnemyMoveSpeedAdjuster

  Adjusts enemies movement speed near round end.
- class EnemyQuickMovement

  Controls enemies quick movement.
- class EnemyShoot

  Shoots projectiles. Projectiles are pooled.
- interface EnemyShootStatusChange

Contract for any entity that can begin, pause, or resume shooting.

- class EnemyShootWhenRequested

  Provides functionality to request projectiles from a pool.
- class **Extensions**

  C# extensions.
- class ExtraBombActionable

  Provides player with an extra bomb when purchased.
- class ExtraFollowerActionable
- class ExtraLifeActionable

  Provides player with an extra life when purchased.
- class ExtraShotActionable

  Provides the player with an extra burst shot when purchased.
- class FadeOutText

  Lerps texts alpha over specified number of seconds.
- class FollowerHealth

  Handles followers life, taking damage, spawning projectiles, and destroying.
- class FPS

  Dispays frames per second counter on screen (useful for measuring performance on mobile devices).
- class GameManager

  Controls game flow. Starts game, initialises new rounds. Maintains list of entities within rounds to pause and resume movement.
- class GameOverUIHandler

  Shows the game over screen and handles UI requests from that screen.
- class GridStatus

  Stores persistent status of grid. Data is stored in PlayerPrefs. When a user disables/enables the grid, it is stored and loaded next time they play. As object is persistent, the grid status is carried from main menu scene to game scene.
- interface HitDeathInvoker

  Provides contract to providing a hook for an entities onDeath and onHit events.
- interface HitListener

  Contract for any entity that can take damage or react to damage.
- class HomingProjectile

  Enables a projectile to change heading based on players current location.
- class InfoScreenToggle

  Shows info screen and handles ui requests from that screen.
- class KeyboardInput

  Provides method to control player based on keyboard input.
- class MainMenuHandler

  Shows main menu screen and hanles UI requests from scene.
- class MenuEnemyDirector

  Directs enemies as part of the main menu scene.
- class MoveDown

  Move down movement state.
- class MoveLeft

  Move left movement state.
- class MovementGridForceApplication

  Applies a directional force to the background grid based on owners direction and velocity.
- interface MovementState

  Contract for a directional movement state.
- class MoveRight

  Move right movement state.
- class MoveUp

Move up movement state.

- class ObjectPool

  Generic object pool.
- class PauseHandler

  Shows pause screen and hanles UI events from that scene.
- class PlayerComponentDisabler

  Disables specified components when player is killed. Enables components when player is spawned.
- class PlayerController

  Updates player position based on input.
- class PlayerHealth

  Handles player health, applying damage, losing lives, and respawning.
- interface PlayerInput

  Contract for getting players next move.
- class PlayerItemUI

  Shows number of lives and bombs on the in-game UI the player currently has.
- class PlayerShoot

  Provides shoot functionality for the player. Projectiles are retrieved from a pool. Also provides burst functionality.
- class PlayerShootController

  Controls all player weapons. Enables the pausing and resuming of shooting i.e. between rounds, or when the player dies/respawns.
- class PlayerShootModules

  Provides functionality to add new shoot modules to player (when purchased through the store).
- class PointPopUpUI

  Shows pop up text when player collects a particle and updates players score.
- class PointsImages

  Handles enabling and disabling of points images (used to signify how many instances of an item have been purchased).
- class PointsText

  Attached to each points text. Handles text movement and fade out.
- interface PoolableProjectile

  Contract for any projectile that can be returned to a pool.
- interface PowerUp

  Contract for all in-game powerups.
- class PowerUpCollector

  Functionality for collecting and activating powerups.
- class PowerUpFallDown

  Attached to each powerup. Enables powerups to fall into a position where they can be picked up by the player.
- class PowerUpImpl

  The abstract base class for all powerups. Provides access to UI text system (to show powerup name) and any common fields.
- class PowerUpParticleExplosion

  Spawns particle explosion on particle pick up.
- class PowerUpSpawn

  Data class for powerup spawns.
- class Projectile

  The standard projectile. Is poolable and effected by blackhole and repel GameObjects.
- interface ProjectileReturn

  A contract for any entity that can pool a projectile.
- class Rotate

  Rotates GameObject on z axis.
- class Round

Responsible for round progression: starting and signifying to the GameManager that the round is complete.

- interface RoundEnemy

    Contract for an enemy that is part of a pround.

- class RoundEnemyImpl

    Round enemy implementation. Informs roundowner when entity has been killed or has escaped round.

- interface RoundManagement

    Contract for any class that can perform actions when a round or challenge round finishes.

- class RoundManager

    Starts current round and provides functionality to begin next round.

- interface RoundOwner

    Contract for any entity responsible for tracking enemies in a round.

- class RoundPersistentScore

    Persistently stores and retrieves the highest round the player has reached. Data is stored in PlayerPrefs.

- class RoundProgressHelper

    Ensures that if an enemy is offscreen for too long a period it is removed from the round.

- class RoundText

    Updates onscreen text to signify a round start or end.

- class ScaleOscillation

    Lerps between min and max scale over time.

- class Score

    Handles players score (points) for a specific run. Has functionality to add (when points are collected) and remove (when player purchases items at the shop) points. Updates the score UI incrementally.

- class ScreenBounds

    Holds data about the screen bounds.

- class ScreenBoundsBounceMovement

    Controls enemies that bouce around the screen.

- class Shield

    Provides shield functionality for player (when purchased through the in-game store).

- class ShieldActionable

    Provides a player with a shield (or additional shield) when purchased. A player can have up to four active shields. When a shield is destroyed the player can purchase the item again (at an increased cost).

- class ShootDamageActionable

    Increases the damage of players projectiles by one when purchased.

- class ShootModuleActionable

    Adds new shoot modules, followers, or shields when purchased.

- interface ShootRecoil

    Contract for any GameObject that can provide recoil.

- class ShootRecoilImpl

    Provides weapon recoil functionality. Where the gun is temporarily moved back by the force of a shot.

- interface ShootRequestable

    Contract for any entity that can shoot projectiles.

- class ShootSpeedPowerUp

    Increases players shooting speed temporaily when picked up.

- class ShopController

    Responsible for opening and closing shop, and updating whether items can be purchased.

- class ShopPurchaseAction

    Actions any shop purchase requests.

- interface ShopPurchaseActionable

    Contract for any item that can be purchased in the shop. Provides methods for performing action, checking if all actions have been performed, and checking if action can be performed.

- class ShopPurchaseActionableImpl

Base class for any shop purchase items. Provides access to and manipulation of all common shop purchase features, including: PointsImages, foreground overlay (enabled when item not currently purchasable), the text that displays the item cost, and the cost value.

- class Singleton

  Generic singleton base class.
- class SortingLayerExposer

  Exposes sorting layer of MeshRenderer.
- class SpeedBoostPowerUp

  Increases players movement speed temporarily.
- class SpriteFadeIn

  Lerps a sprites alpha from 0 to 1 over a set time defined by GameManager::ROUND_BEGIN_TIME.
- class SpriteOutline
- class StationaryContainer

  Holds all stationary enemies within a round.
- class StationaryMovement

  Controls stationary enemies fade in and collider enabled status.
- class TouchInput

  Provides a method to control the player based on mobile touch input.
- class UpgradeShipSpeedActionable

  Increases the players movement speed when purchased.
- class UpgradeShootSpeedActionable

  Decreases time between shots for player when purchased.
- class VerticalGroupMovement

  Controls groups of vertically moving enemies.
- class VerticalWrapAroundMovement

  Controls enemies that move vertically and wrap around the screen.
- class YIdleOscillation

  Y idle oscillation. No oscillation is performed.
- interface YMovementOscillation

  Contract for perfoming Oscillation.
- class YMovementOscillationImpl

  Implementation of Y Oscillation.

Enumerations

- enum MovementDirection { **Up**, **Left**, **Down**, **Right** }

  Possible enemy movement directions.

4.1.1 Enumeration Type Documentation

4.1.1.1 MovementDirection

```
enum GameCore.MovementDirection  [strong]
```

Possible enemy movement directions.

4.2 PE2D Namespace Reference

Classes

- class CircularArray

  Simplified version of the circular buffer found at: `http://geekswithblogs.net/blackrob/archive/2014/09/01/circu` `aspx`*. Generic storage, used to store particles.*
- class CustomParticle

  Main workhorse for the custom particles. Updates particles state (colour, position, velocity etc), handles interaction with effectors, and applys any screen constraints.
- class CustomParticleEmitter

  Base class for ParticleEmitterInRandomDirection and ParticleEmitterInObjectDirection. Add base classes to GameObjects to easily create particle emitters.
- class DemoConstraintSwitcher

  Switches between screen constraints in the demo scene.
- class DemoMouseController

  Spawns a circular explosion of particles on mouse click. Example of how to procedurally create particles.
- class DemoParticleEmitterSwitcher

  Switches between particle emitters in demo scene.
- class DemoSceneSwitcher

  Switches between demo scenes when enter key pressed.
- struct ParticleBuilder

  Holds the particle state. Passed to the ParticleFactory to build particles.
- class ParticleEffector

  Add to a gameobject to effect a particles movement.
- class ParticleEmitterInObjectDirection

  Emits particles based on objects rotation.
- class ParticleEmitterInRandomDirection

  Emits particles from objects position in a random direction.
- class ParticleFactory

  Creates and maintain an object pool of particles.
- class ParticleRenderer

  Simple renderer script for particles that disables the sprite renderer on enable and re-enables the srpite renderer after a time specified by ParticleRenderer::RENDERER_DELAY. Attach to the particle prefab to prevent occasional graphic glitches.
- class Pulsate

  Simple script used to pulse an objects size. Used in the demo scene for the effectors.
- class **StaticExtensions**

  Extensions for static classes. COntains a number of helper methods used throughout project.

Enumerations

- enum WrapAroundType { **None**, **WrapAround**, **Constrain** }

  Screen constraint type.
- enum EffectorType { **Attraction**, **Repel**, **BlackHole** }

  Effector types. Attraction pulls particles towards object, repel pushes particles away from object, and blackhole attracts objects until a certain point and then the particle encircles the object.

4.2.1 Enumeration Type Documentation

4.2.1.1 EffectorType

`enum PE2D.EffectorType [strong]`

Effector types. Attraction pulls particles towards object, repel pushes particles away from object, and blackhole attracts objects until a certain point and then the particle encircles the object.

4.2.1.2 WrapAroundType

`enum PE2D.WrapAroundType [strong]`

Screen constraint type.

4.3 WarpGrid Namespace Reference

Classes

- class Demo_Grid

  Used to demonstrate how to apply a force to an existing grid.
- class Grid

  Updates and displays warping grid. This is a conversion of an XNA project found here: https://gamedevelopment.tutsplus.com/tutorials/make-a-neon-vector-shooter-in-xna-the-warping-gr
- class Interpolate
- class PointMass

  A moveable point on the grid.
- struct Spring

  Connects two PointMass on a grid.

4.3.1 Detailed Description

Interpolation utility functions: easing, bezier, and catmull-rom. Consider using Unity's Animation curve editor and AnimationCurve class before scripting the desired behaviour using this utility.

Interpolation functionality available at different levels of abstraction. Low level access via individual easing functions (ex. EaseInOutCirc), Bezier(), and CatmullRom(). High level access using sequence generators, NewEase(), NewBezier(), and NewCatmullRom().

Sequence generators are typically used as follows:

IEnumerable<Vector3> sequence = Interpolate.New[Ease|Bezier|CatmulRom](configuration); foreach (Vector3 newPoint in sequence) { transform.position = newPoint; yield return WaitForSeconds(1.0f); }

Or:

IEnumerator<Vector3> sequence = Interpolate.New[Ease|Bezier|CatmulRom](configuration).GetEnumerator(); function Update() { if (sequence.MoveNext()) { transform.position = sequence.Current; } }

The low level functions work similarly to Unity's built in Lerp and it is up to you to track and pass in elapsedTime and duration on every call. The functions take this form (or the logical equivalent for Bezier() and CatmullRom()).

transform.position = ease(start, distance, elapsedTime, duration);

For convenience in configuration you can use the Ease(EaseType) function to look up a concrete easing function:

[SerializeField] Interpolate.EaseType easeType; // set using Unity's property inspector Interpolate.Function ease; // easing of a particular EaseType function Awake() { ease = Interpolate.Ease(easeType); }

Author

Fernando Zapata (fernando@cpudreams.com) Andrea85cs (andrea85cs@dynematica.it)

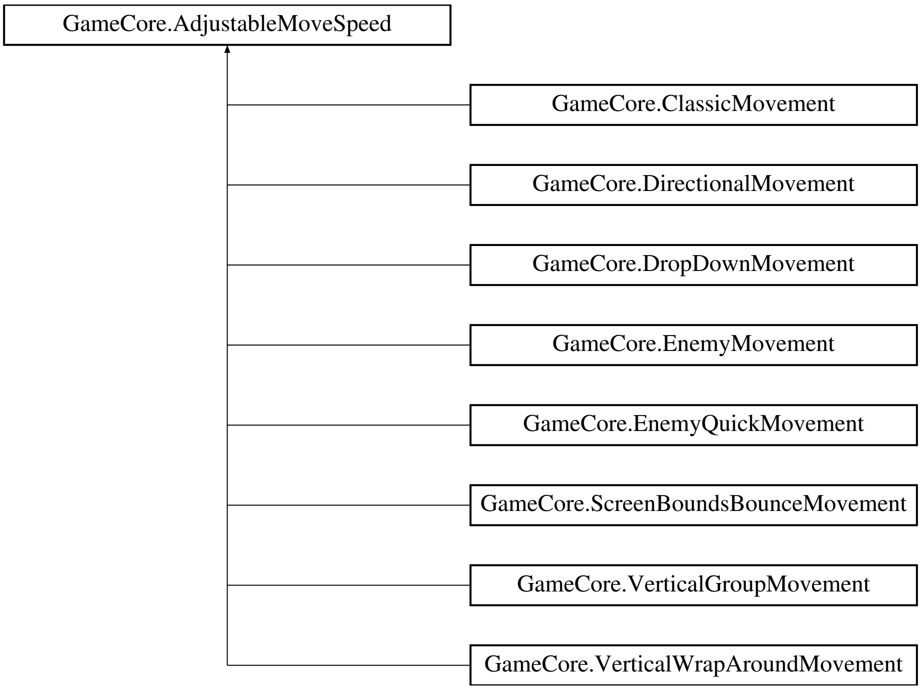
Chapter 5

Class Documentation

5.1 GameCore.AdjustableMoveSpeed Interface Reference

Contract for any object that has an adjustable move speed.

Inheritance diagram for GameCore.AdjustableMoveSpeed:


GameCore.AdjustableMoveSpeed
GameCore.ClassicMovement
GameCore.DirectionalMovement
GameCore.DropDownMovement
GameCore.EnemyMovement
GameCore.EnemyQuickMovement
GameCore.ScreenBoundsBounceMovement
GameCore.VerticalGroupMovement
GameCore.VerticalWrapAroundMovement

Public Member Functions

- void **IncrementSpeed** ()

5.1.1 Detailed Description

Contract for any object that has an adjustable move speed.

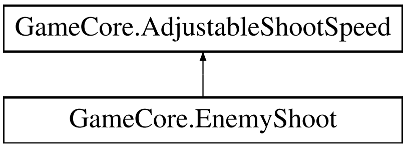
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/AdjustableMoveSpeed.cs

5.2 GameCore.AdjustableShootSpeed Interface Reference

Contract for any object that has an adjustable shoot speed.

Inheritance diagram for GameCore.AdjustableShootSpeed:



Public Member Functions

- void **IncrementSpeed** ()

5.2.1 Detailed Description

Contract for any object that has an adjustable shoot speed.

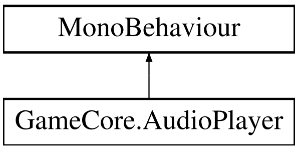
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/AdjustableMoveSpeed.cs

5.3 GameCore.AudioPlayer Class Reference

Attach to an audio source. Used to play instances of AudioClip.

Inheritance diagram for GameCore.AudioPlayer:



Public Member Functions

- void PlayInstance (AudioClip clip)

    Play AudioClip if not muted.

5.3.1 Detailed Description

Attach to an audio source. Used to play instances of AudioClip.

5.3.2 Member Function Documentation

5.3.2.1 PlayInstance()

```
void GameCore.AudioPlayer.PlayInstance (
             AudioClip clip )
```

Play AudioClip if not muted.

Parameters

clip	AudioClip to play

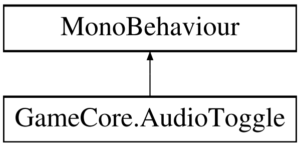
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Audio/AudioPlayer.cs

5.4 GameCore.AudioToggle Class Reference

Toggles game audio on button press.

Inheritance diagram for GameCore.AudioToggle:



Public Member Functions

- void Toggle ()

  Toggles audio playing and button image.

Public Attributes

- Sprite audioNonMutedImage

  Image to display when audio is not muted.
- Sprite audioMutedImage

  Image to display when audio is muted.

5.4.1 Detailed Description

Toggles game audio on button press.

5.4.2 Member Function Documentation

5.4.2.1 Toggle()

```
void GameCore.AudioToggle.Toggle ( )
```

Toggles audio playing and button image.

5.4.3 Member Data Documentation

5.4.3.1 audioMutedImage

```
Sprite GameCore.AudioToggle.audioMutedImage
```

Image to display when audio is muted.

5.4.3.2 audioNonMutedImage

```
Sprite GameCore.AudioToggle.audioNonMutedImage
```

Image to display when audio is not muted.

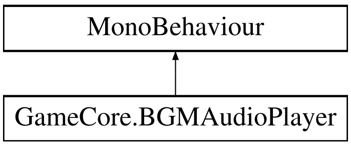
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Audio/AudioToggle.cs

5.5 GameCore.BGMAudioPlayer Class Reference

Handles playing background music. Can fade between clips and queue pitch and volume lerp.

Inheritance diagram for GameCore.BGMAudioPlayer:



Public Member Functions

- void ToggleAudio ()

  Toggles audio playing. Stores setting in PlayerPrefs and is loaded at game start.
- void PlayGameOverBGM ()

  Plays the game over AudioClip.
- void SetVolume (float vol, float seconds, AudioClip clipToChangeTo=null)

  Enques a volume change. This volume change is lerped to by seconds.
- void IncreasePitch (float amount)

  Increases the pitch by amount. Capped by maxBGMPitch.
- void SetPitch (float pitch, float seconds, AudioClip clipToChangeTo=null)

  Enques a pitch change. This pitch change is lerped to by seconds.
- void SwitchClips (AudioClip to, float seconds)

  Fade between the currently play clip and this clip.

Public Attributes

- AudioClip menuAudio

  The AudioClip to play at main menu.
- AudioClip [ ] gameAudio

  The AudioClips to play when gameplay scene is loaded. If more than one, a random clip is selected.
- AudioClip gameoverClip

  The AudioClip to play on game over.
- float switchAudioTrackLerpSecs = 2f

  The time in seconds the fade between audio tracks lasts.
- float maxBGMPitch = 1.1f

  The maximum audio pitch
- float maxBGMVolume = 0.7f

  The max background volume.

Properties

- bool muted `[get]`

  Whether this instance of BGMAudioPlayer is muted.

5.5.1 Detailed Description

Handles playing background music. Can fade between clips and queue pitch and volume lerp.

5.5.2 Member Function Documentation

5.5.2.1 IncreasePitch()

```
void GameCore.BGMAudioPlayer.IncreasePitch (
             float amount )
```

Increases the pitch by amount. Capped by maxBGMPitch.

Parameters

amount	Amount.

5.5.2.2 PlayGameOverBGM()

```
void GameCore.BGMAudioPlayer.PlayGameOverBGM ( )
```

Plays the game over AudioClip.

5.5.2.3 SetPitch()

```
void GameCore.BGMAudioPlayer.SetPitch (
             float pitch,
             float seconds,
             AudioClip clipToChangeTo = null )
```

Enques a pitch change. This pitch change is lerped to by seconds.

Parameters

pitch	Target pitch.
seconds	Seconds to spend lerping to pitch.
clipToChangeTo	Clip to change to. If null the current clip is continued.

5.5.2.4 SetVolume()

```
void GameCore.BGMAudioPlayer.SetVolume (
             float vol,
             float seconds,
             AudioClip clipToChangeTo = null )
```

Enques a volume change. This volume change is lerped to by seconds.

Parameters

vol	Target Volume.
seconds	Seconds to spend lerping to volume.
clipToChangeTo	Clip to change to. If null the current clip is continued.

5.5.2.5 SwitchClips()

```
void GameCore.BGMAudioPlayer.SwitchClips (
             AudioClip to,
             float seconds )
```

Fade between the currently play clip and this clip.

Parameters

to	The clip to fade to.
seconds	The time in seconds for the fade.

5.5.2.6 ToggleAudio()

```
void GameCore.BGMAudioPlayer.ToggleAudio ( )
```

Toggles audio playing. Stores setting in PlayerPrefs and is loaded at game start.

5.5.3 Member Data Documentation

5.5.3.1 gameAudio

```
AudioClip [] GameCore.BGMAudioPlayer.gameAudio
```

The AudioClips to play when gameplay scene is loaded. If more than one, a random clip is selected.

5.5.3.2 gameoverClip

```
AudioClip GameCore.BGMAudioPlayer.gameoverClip
```

The AudioClip to play on game over.

5.5.3.3 maxBGMPitch

```
float GameCore.BGMAudioPlayer.maxBGMPitch = 1.1f
```

The maximum audio pitch

5.5.3.4 maxBGMVolume

```
float GameCore.BGMAudioPlayer.maxBGMVolume = 0.7f
```

The max background volume.

5.5.3.5 menuAudio

```
AudioClip GameCore.BGMAudioPlayer.menuAudio
```

The AudioClip to play at main menu.

5.5.3.6 switchAudioTrackLerpSecs

```
float GameCore.BGMAudioPlayer.switchAudioTrackLerpSecs = 2f
```

The time in seconds the fade between audio tracks lasts.

5.5.4 Property Documentation

5.5.4.1 muted

```
bool GameCore.BGMAudioPlayer.muted  [get]
```

Whether this instance of BGMAudioPlayer is muted.

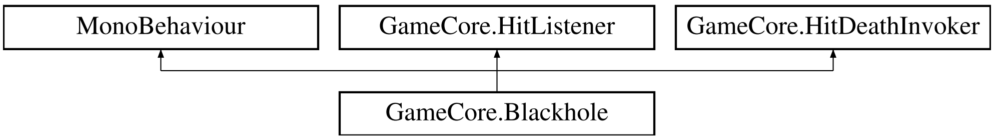
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Audio/BGMAudioPlayer.cs

5.6 GameCore.Blackhole Class Reference

Attached to a blackhole or repel. Controls health.

Inheritance diagram for GameCore.Blackhole:



Public Member Functions

- void ExplosionInRange (int damage)

    Applies explosive damage to object.
- void OnHit (int damage)

    Applies damage to object.
- void Kill (bool cameraShake)

    Kills this instance. Plays audio, raises on death event, spawns explosion, and destroys GameObject.
- void PlayOnDeathAudio ()

    Plays the on death audio.

Public Attributes

- int hitPoints = 1

  The starting health.
- AudioClip audioOnDeath

  The audio to play on death.
- AudioClip audioOnDamage

  The audio to play on damage.
- Color particleSpewColour

  The colour of the particles that are released by this object (if spewParticles is true).
- float percentageScaleDownWhenHit = 20f

  The percentage to scale down when hit.
- int numOfParticlesOnDeath = 30

  The number of particles to spawn on death.
- int numOfParticlesOnHit = 20

  The number of particles to spawn on hit.
- bool spewParticles = true

  Sets whether this instance should release particles whilst it is alive.

Properties

- Action onHit `[get, set]`

  Gets or sets the on hit action. This is called when the object has been damaged.
- Action onDeath `[get, set]`

  Gets or sets the on death action. This is called when the object is killed.

5.6.1 Detailed Description

Attached to a blackhole or repel. Controls health.

5.6.2 Member Function Documentation

5.6.2.1 ExplosionInRange()

```
void GameCore.Blackhole.ExplosionInRange (
             int damage )
```

Applies explosive damage to object.

Parameters

damage	Damage to apply.

5.6.2.2 Kill()

```
void GameCore.Blackhole.Kill (
             bool cameraShake )
```

Kills this instance. Plays audio, raises on death event, spawns explosion, and destroys GameObject.

Parameters

cameraShake	If set to `true` camera shake is applied.

5.6.2.3 OnHit()

```
void GameCore.Blackhole.OnHit (
             int damage )
```

Applies damage to object.

Parameters

damage	Damage to apply.

Implements GameCore.HitListener.

5.6.2.4 PlayOnDeathAudio()

```
void GameCore.Blackhole.PlayOnDeathAudio ( )
```

Plays the on death audio.

5.6.3 Member Data Documentation

5.6.3.1 audioOnDamage

```
AudioClip GameCore.Blackhole.audioOnDamage
```

The audio to play on damage.

5.6.3.2 audioOnDeath

```
AudioClip GameCore.Blackhole.audioOnDeath
```

The audio to play on death.

5.6.3.3 hitPoints

```
int GameCore.Blackhole.hitPoints = 1
```

The starting health.

5.6.3.4 numOfParticlesOnDeath

```
int GameCore.Blackhole.numOfParticlesOnDeath = 30
```

The number of particles to spawn on death.

5.6.3.5 numOfParticlesOnHit

```
int GameCore.Blackhole.numOfParticlesOnHit = 20
```

The number of particles to spawn on hit.

5.6.3.6 particleSpewColour

```
Color GameCore.Blackhole.particleSpewColour
```

The colour of the particles that are released by this object (if spewParticles is true).

5.6.3.7 percentageScaleDownWhenHit

```
float GameCore.Blackhole.percentageScaleDownWhenHit = 20f
```

The percentage to scale down when hit.

5.6.3.8 spewParticles

```
bool GameCore.Blackhole.spewParticles = true
```

Sets whether this instance should release particles whilst it is alive.

5.6.4 Property Documentation

5.6.4.1 onDeath

```
Action GameCore.Blackhole.onDeath  [get], [set]
```

Gets or sets the on death action. This is called when the object is killed.

The on death.

5.6.4.2 onHit

```
Action GameCore.Blackhole.onHit  [get], [set]
```

Gets or sets the on hit action. This is called when the object has been damaged.

The on hit action.

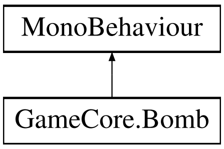
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Effectors/Blackhole.cs

5.7 GameCore.Bomb Class Reference

Responsible for animating the bomb and damaging enemies on explosion.

Inheritance diagram for GameCore.Bomb:



Public Member Functions

- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.

Public Attributes

- AudioClip audioToPlayOnExplode

  The audio to play on explosion.
- AudioClip audioToPlayOnFlash

  The audio to play on countdown to explosion.
- int damage = 1

  The damage to apply to enemies in proximity.
- float radius = 2.5f

  The radius. Enemies within this radius have damage applied to them.
- float secsToExplode = 5f

  The seconds from bomb placement to explosion.
- Color [ ] colors

  The colors to loop over while counting down to explosion.

5.7.1 Detailed Description

Responsible for animating the bomb and damaging enemies on explosion.

5.7.2 Member Function Documentation

5.7.2.1 Pause()

```
void GameCore.Bomb.Pause ( )
```

Pause this instance.

5.7.2.2 Resume()

```
void GameCore.Bomb.Resume ( )
```

Resume this instance.

5.7.3 Member Data Documentation

5.7.3.1 audioToPlayOnExplode

```
AudioClip GameCore.Bomb.audioToPlayOnExplode
```

The audio to play on explosion.

5.7.3.2 audioToPlayOnFlash

```
AudioClip GameCore.Bomb.audioToPlayOnFlash
```

The audio to play on countdown to explosion.

5.7.3.3 colors

```
Color [] GameCore.Bomb.colors
```

The colors to loop over while counting down to explosion.

5.7.3.4 damage

```
int GameCore.Bomb.damage = 1
```

The damage to apply to enemies in proximity.

5.7.3.5 radius

```
float GameCore.Bomb.radius = 2.5f
```

The radius. Enemies within this radius have damage applied to them.

5.7.3.6 secsToExplode

```
float GameCore.Bomb.secsToExplode = 5f
```

The seconds from bomb placement to explosion.

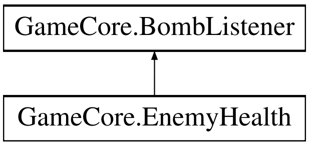
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Bombs/Bomb.cs

5.8 GameCore.BombListener Interface Reference

Contract for any entity that can take damage from a bomb.

Inheritance diagram for GameCore.BombListener:



Public Member Functions

- void ExplosionInRange (int damage)

  Apply damage from explosion.

Properties

- Transform owner `[get]`

  Gets the owner.

5.8.1 Detailed Description

Contract for any entity that can take damage from a bomb.

5.8.2 Member Function Documentation

5.8.2.1 ExplosionInRange()

```
void GameCore.BombListener.ExplosionInRange (
              int damage )
```

Apply damage from explosion.

Parameters

damage	Damage to apply.

Implemented in GameCore.EnemyHealth.

5.8.3 Property Documentation

5.8.3.1 owner

```
Transform GameCore.BombListener.owner  [get]
```

Gets the owner.

The owner.

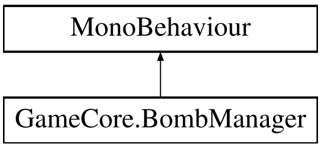
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyHealth.cs

5.9 GameCore.BombManager Class Reference

Spawns bomb within set bounds when player taps on screen.

Inheritance diagram for GameCore.BombManager:



Public Member Functions

- void IncrementBombCount ()

  Increments the bomb count.

Public Attributes

- int initialBombCount = 1

  The number of bombs the player starts with.
- GameObject bombPrefab

  The bomb prefab to spawn.

5.9.1 Detailed Description

Spawns bomb within set bounds when player taps on screen.

5.9.2 Member Function Documentation

5.9.2.1 IncrementBombCount()

```
void GameCore.BombManager.IncrementBombCount ( )
```

Increments the bomb count.

5.9.3 Member Data Documentation

5.9.3.1 bombPrefab

```
GameObject GameCore.BombManager.bombPrefab
```

The bomb prefab to spawn.

5.9.3.2 initialBombCount

```
int GameCore.BombManager.initialBombCount = 1
```

The number of bombs the player starts with.

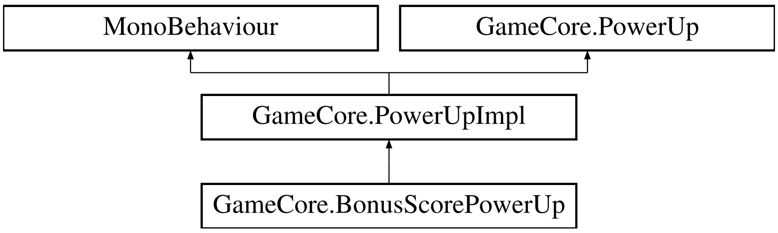
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Bombs/BombManager.cs

5.10 GameCore.BonusScorePowerUp Class Reference

Provides bonus particles.

Inheritance diagram for GameCore.BonusScorePowerUp:



Public Member Functions

- override void Perform (Transform player)

  Perform the specified powerup action. Spawns additional particles at powerup location.

Additional Inherited Members

5.10.1 Detailed Description

Provides bonus particles.

5.10.2 Member Function Documentation

5.10.2.1 Perform()

```
override void GameCore.BonusScorePowerUp.Perform (
            Transform player )  [virtual]
```

Perform the specified powerup action. Spawns additional particles at powerup location.

Parameters

player	Player tranform.

Implements GameCore.PowerUpImpl.

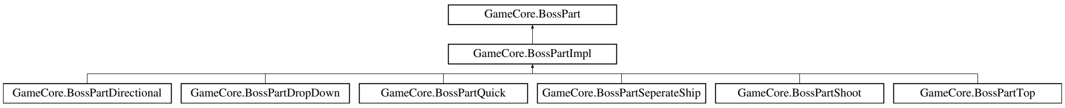
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/BonusScorePowerUp.cs

5.11 GameCore.BossPart Interface Reference

Contract for all boss parts.

Inheritance diagram for GameCore.BossPart:



Public Member Functions

- void **Activate** ()

5.11.1 Detailed Description

Contract for all boss parts.

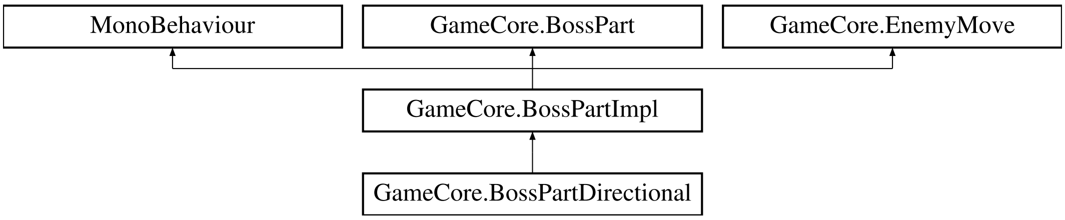
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartImpl.cs

5.12 GameCore.BossPartDirectional Class Reference

Controls the directional boss part.

Inheritance diagram for GameCore.BossPartDirectional:



Public Member Functions

- override void Pause ()

    Pause this instance.
- override void Resume ()

    Resume this instance.

Public Attributes

- float moveSpeed = 5f

    The movement speed.
- MovementDirection [ ] moveDirections

    The movement directions.
- float moveOffset = 2f

    The distance to move each time.
- float minDistToTarget = 0.1f

    The minimum distance part must be to target before next direction is calculated.
- float pauseOnTargetReach = 0.5f

    The seconds to pause when target.
- float rotateSpeed = 320.0f

    The rotate speed.
- int numOfProjectilesToRequest = 1

    The number of projectiles to request on each shoot attempt.

Protected Member Functions

- override void **Awake** ()
- override void **Start** ()
- override void **DoActivation** ()

Additional Inherited Members

5.12.1 Detailed Description

Controls the directional boss part.

5.12.2 Member Function Documentation

5.12.2.1 Pause()

```
override void GameCore.BossPartDirectional.Pause ( )  [virtual]
```

Pause this instance.

Implements GameCore.BossPartImpl.

5.12.2.2 Resume()

```
override void GameCore.BossPartDirectional.Resume ( )  [virtual]
```

Resume this instance.

Implements GameCore.BossPartImpl.

5.12.3 Member Data Documentation

5.12.3.1 minDistToTarget

```
float GameCore.BossPartDirectional.minDistToTarget = 0.1f
```

The minimum distance part must be to target before next direction is calculated.

5.12.3.2 moveDirections

```
MovementDirection [] GameCore.BossPartDirectional.moveDirections
```

The movement directions.

5.12.3.3 moveOffset

```
float GameCore.BossPartDirectional.moveOffset = 2f
```

The distance to move each time.

5.12.3.4 moveSpeed

```
float GameCore.BossPartDirectional.moveSpeed = 5f
```

The movement speed.

5.12.3.5 numOfProjectilesToRequest

```
int GameCore.BossPartDirectional.numOfProjectilesToRequest = 1
```

The number of projectiles to request on each shoot attempt.

5.12.3.6 pauseOnTargetReach

```
float GameCore.BossPartDirectional.pauseOnTargetReach = 0.5f
```

The seconds to pause when target.

5.12.3.7 rotateSpeed

```
float GameCore.BossPartDirectional.rotateSpeed = 320.0f
```

The rotate speed.

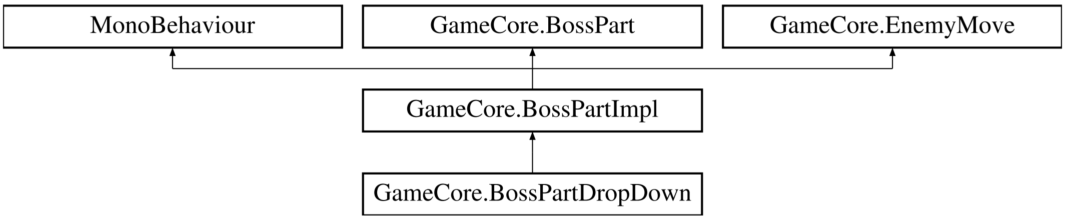
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartDirectional.cs

5.13 GameCore.BossPartDropDown Class Reference

Controls the boss part drop down.

Inheritance diagram for GameCore.BossPartDropDown:



Public Member Functions

- override void Pause ()

  Pause this instance.
- override void Resume ()

  Resume this instance.

Public Attributes

- float horMoveSpeed = 5f

  The horizontal movement speed.
- Vector2 minMaxTimeToDropDown = new Vector2(2f, 5f)

  The minimum and maximum time before the boss part drops down. A random number is selected between these two values.
- float dropSpeed

  The initial drop movement speed.
- float dropSpeedUp = 15f

  The drop speed increment.
- float bounceUpSpeed

  The speed the boss part moves up before dropping down.
- float bounceUpDistance = 0.6f

  THe distance to move up before dropping down.
- float secDelayBetweenProjShoot = 1f

  The second delay between projectile shoot requests.

Protected Member Functions

- override void **Awake** ()
- override void **DoActivation** ()

Additional Inherited Members

5.13.1 Detailed Description

Controls the boss part drop down.

5.13.2 Member Function Documentation

5.13.2.1 Pause()

```
override void GameCore.BossPartDropDown.Pause ( )  [virtual]
```

Pause this instance.

Implements GameCore.BossPartImpl.

5.13.2.2 Resume()

```
override void GameCore.BossPartDropDown.Resume ( )  [virtual]
```

Resume this instance.

Implements GameCore.BossPartImpl.

5.13.3 Member Data Documentation

5.13.3.1 bounceUpDistance

```
float GameCore.BossPartDropDown.bounceUpDistance = 0.6f
```

THe distance to move up before dropping down.

5.13.3.2 bounceUpSpeed

```
float GameCore.BossPartDropDown.bounceUpSpeed
```

The speed the boss part moves up before dropping down.

5.13.3.3 dropSpeed

```
float GameCore.BossPartDropDown.dropSpeed
```

The initial drop movement speed.

5.13.3.4 dropSpeedUp

```
float GameCore.BossPartDropDown.dropSpeedUp = 15f
```

The drop speed increment.

5.13.3.5 horMoveSpeed

```
float GameCore.BossPartDropDown.horMoveSpeed = 5f
```

The horizontal movement speed.

5.13.3.6 minMaxTimeToDropDown

```
Vector2 GameCore.BossPartDropDown.minMaxTimeToDropDown = new Vector2(2f, 5f)
```

The minimum and maximum time before the boss part drops down. A random number is selected between these two values.

5.13.3.7 secDelayBetweenProjShoot

```
float GameCore.BossPartDropDown.secDelayBetweenProjShoot = 1f
```

The second delay between projectile shoot requests.

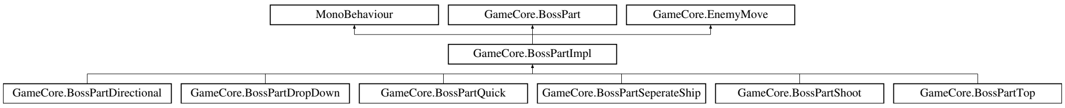
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartDropDown.cs

5.14 GameCore.BossPartImpl Class Reference

Abstract concrete implementation of BossPart.

Inheritance diagram for GameCore.BossPartImpl:



Public Member Functions

- void Begin ()

  Begin this instance.
- abstract void Pause ()

  Pause this instance.
- abstract void Resume ()

  Resume this instance.
- void Activate ()

  Activate this instance. Waits for player to spawn, flashes sprite, and then calls abstract DoActivation method (which is implementd in sub classes).

Public Attributes

- BossPartImpl next

  The next boss part. When this boss part dies, the next part is activated (if not null).
- bool isFirst = false

  Signifies if this instance is the first in the boss queue.

Protected Member Functions

- virtual void **Awake** ()
- virtual void **Start** ()
- abstract void **DoActivation** ()

Protected Attributes

- bool **m_HasBeenActivated** = false

5.14.1 Detailed Description

Abstract concrete implementation of BossPart.

5.14.2 Member Function Documentation

5.14.2.1 Activate()

```
void GameCore.BossPartImpl.Activate ( )
```

Activate this instance. Waits for player to spawn, flashes sprite, and then calls abstract DoActivation method (which is implementd in sub classes).

Implements GameCore.BossPart.

5.14.2.2 Begin()

```
void GameCore.BossPartImpl.Begin ( )
```

Begin this instance.

Implements GameCore.EnemyMove.

5.14.2.3 Pause()

```
abstract void GameCore.BossPartImpl.Pause ( )  [pure virtual]
```

Pause this instance.

Implements GameCore.EnemyMove.

Implemented in GameCore.BossPartDirectional, GameCore.BossPartDropDown, GameCore.BossPartTop, GameCore.BossPartQuick, GameCore.BossPartSeperateShip, and GameCore.BossPartShoot.

5.14.2.4 Resume()

```
abstract void GameCore.BossPartImpl.Resume ( )  [pure virtual]
```

Resume this instance.

Implements GameCore.EnemyMove.

Implemented in GameCore.BossPartDirectional, GameCore.BossPartDropDown, GameCore.BossPartTop, GameCore.BossPartQuick, GameCore.BossPartSeperateShip, and GameCore.BossPartShoot.

5.14.3 Member Data Documentation

5.14.3.1 isFirst

```
bool GameCore.BossPartImpl.isFirst = false
```

Signifies if this instance is the first in the boss queue.

5.14.3.2 next

```
BossPartImpl GameCore.BossPartImpl.next
```

The next boss part. When this boss part dies, the next part is activated (if not null).

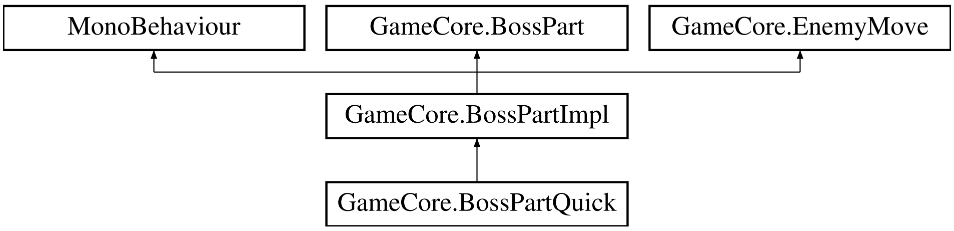
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartImpl.cs

5.15 GameCore.BossPartQuick Class Reference

Controls the boss part quick.

Inheritance diagram for GameCore.BossPartQuick:



Public Member Functions

- override void Pause ()

  Pause this instance.
- override void Resume ()

  Resume this instance.

Public Attributes

- MovementDirection movementDirection = MovementDirection.Left

  The movement direction.
- float moveSpeed = 10f

  The movement speed.
- float xShootPosition = 0f

  The X position on screen where the part should pause to shoot.
- float delayBeforeMoving = 0.5f

  The delay before moving after shooting.

Protected Member Functions

- override void **Awake** ()
- override void **Start** ()
- override void **DoActivation** ()

Additional Inherited Members

5.15.1 Detailed Description

Controls the boss part quick.

5.15.2 Member Function Documentation

5.15.2.1 Pause()

```
override void GameCore.BossPartQuick.Pause ( )  [virtual]
```

Pause this instance.

Implements GameCore.BossPartImpl.

5.15.2.2 Resume()

```
override void GameCore.BossPartQuick.Resume ( )  [virtual]
```

Resume this instance.

Implements GameCore.BossPartImpl.

5.15.3 Member Data Documentation

5.15.3.1 delayBeforeMoving

```
float GameCore.BossPartQuick.delayBeforeMoving = 0.5f
```

The delay before moving after shooting.

5.15.3.2 movementDirection

```
MovementDirection GameCore.BossPartQuick.movementDirection = MovementDirection.Left
```

The movement direction.

5.15.3.3 moveSpeed

```
float GameCore.BossPartQuick.moveSpeed = 10f
```

The movement speed.

5.15.3.4 xShootPosition

```
float GameCore.BossPartQuick.xShootPosition = 0f
```

The X position on screen where the part should pause to shoot.

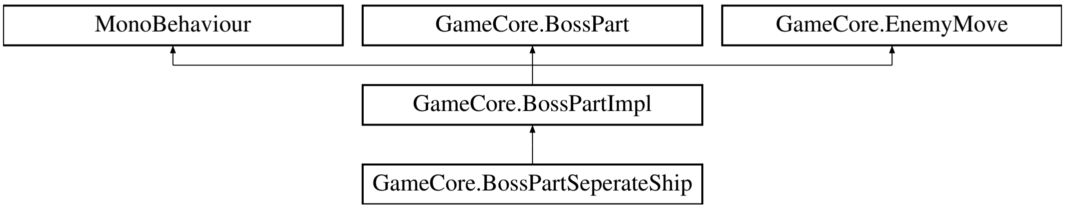
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartQuick.cs

5.16 GameCore.BossPartSeperateShip Class Reference

Controls the boss part seperate ship.

Inheritance diagram for GameCore.BossPartSeperateShip:



Public Member Functions

- override void Pause ()

  Pause this instance.
- override void Resume ()

  Resume this instance.

Public Attributes

- float moveSpeed = 10f

  The movement speed.
- LayerMask hitMask

  The hit mask for the screen bounds.

Protected Member Functions

- override void **Awake** ()
- override void **Start** ()
- override void **DoActivation** ()

Additional Inherited Members

5.16.1 Detailed Description

Controls the boss part seperate ship.

5.16.2 Member Function Documentation

5.16.2.1 Pause()

```
override void GameCore.BossPartSeperateShip.Pause ( ) [virtual]
```

Pause this instance.

Implements GameCore.BossPartImpl.

5.16.2.2 Resume()

```
override void GameCore.BossPartSeperateShip.Resume ( ) [virtual]
```

Resume this instance.

Implements GameCore.BossPartImpl.

5.16.3 Member Data Documentation

5.16.3.1 hitMask

```
LayerMask GameCore.BossPartSeperateShip.hitMask
```

The hit mask for the screen bounds.

5.16.3.2 moveSpeed

```
float GameCore.BossPartSeperateShip.moveSpeed = 10f
```

The movement speed.

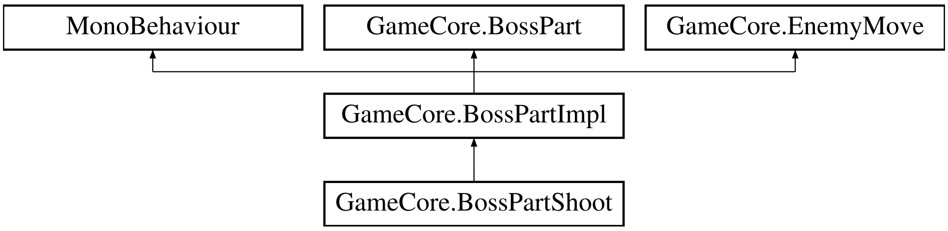
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartSeperateShip.cs

5.17 GameCore.BossPartShoot Class Reference

Controls the boss part shoot.

Inheritance diagram for GameCore.BossPartShoot:



Public Member Functions

- override void Pause ()

  Pause this instance.
- override void Resume ()

  Resume this instance.

Protected Member Functions

- override void **Awake** ()
- override void **Start** ()
- override void **DoActivation** ()

Additional Inherited Members

5.17.1 Detailed Description

Controls the boss part shoot.

5.17.2 Member Function Documentation

5.17.2.1 Pause()

```
override void GameCore.BossPartShoot.Pause ( ) [virtual]
```

Pause this instance.

Implements GameCore.BossPartImpl.

5.17.2.2 Resume()

```
override void GameCore.BossPartShoot.Resume ( ) [virtual]
```

Resume this instance.

Implements GameCore.BossPartImpl.

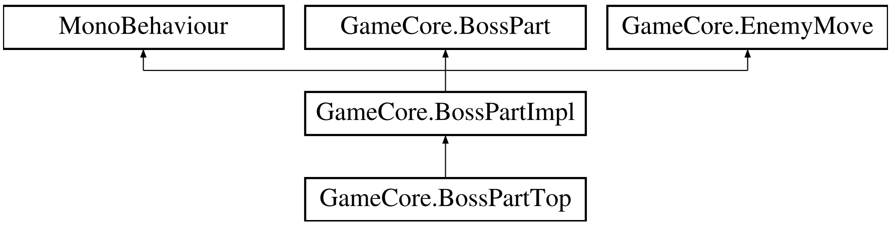
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartShoot.cs

5.18 GameCore.BossPartTop Class Reference

Controls the boss top part.

Inheritance diagram for GameCore.BossPartTop:



Public Member Functions

- override void Pause ()

  Pause this instance.
- override void Resume ()

  Resume this instance.

Public Attributes

- float moveSpeed = 5f

  The movement speed.
- LayerMask hitMask

  The layer mask for the screen bounds.
- float rotateSpeed = 5f

  The rotation speed.

Protected Member Functions

- override void **Awake** ()
- override void **Start** ()
- override void **DoActivation** ()

Additional Inherited Members

5.18.1 Detailed Description

Controls the boss top part.

5.18.2 Member Function Documentation

5.18.2.1 Pause()

```
override void GameCore.BossPartTop.Pause ( )  [virtual]
```

Pause this instance.

Implements GameCore.BossPartImpl.

5.18.2.2 Resume()

```
override void GameCore.BossPartTop.Resume ( )  [virtual]
```

Resume this instance.

Implements GameCore.BossPartImpl.

5.18.3 Member Data Documentation

5.18.3.1 hitMask

```
LayerMask GameCore.BossPartTop.hitMask
```

The layer mask for the screen bounds.

5.18.3.2 moveSpeed

```
float GameCore.BossPartTop.moveSpeed = 5f
```

The movement speed.

5.18.3.3 rotateSpeed

```
float GameCore.BossPartTop.rotateSpeed = 5f
```

The rotation speed.

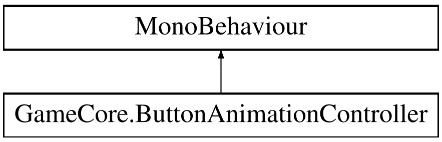
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Boss Parts/BossPartTop.cs

5.19 GameCore.ButtonAnimationController Class Reference

Holds collections of ButtonAnimator. Responsible for playing the button aniamtions, with a small delay between each row.

Inheritance diagram for GameCore.ButtonAnimationController:



Public Attributes

- ButtonAnimator [ ] buttonLeftAnimators

  The ButtonAnimator attached to the left column of shop buttons. These should be added from top to bottom i.e. the top left button is at index 0.
- ButtonAnimator [ ] buttonRightAnimators

  The ButtonAnimator attached to the right column of shop buttons. These should be added from top to bottom i.e. the top right button is at index 0.
- float delayBetweenAnimations = 0.2f

  The delay between animating each row.
- Action OnAnimationComplete

  Invoked when shop has finished animating.

5.19.1 Detailed Description

Holds collections of ButtonAnimator. Responsible for playing the button aniamtions, with a small delay between each row.

5.19.2 Member Data Documentation

5.19.2.1 buttonLeftAnimators

```
ButtonAnimator [] GameCore.ButtonAnimationController.buttonLeftAnimators
```

The ButtonAnimator attached to the left column of shop buttons. These should be added from top to bottom i.e. the top left button is at index 0.

5.19.2.2 buttonRightAnimators

```
ButtonAnimator [] GameCore.ButtonAnimationController.buttonRightAnimators
```

The ButtonAnimator attached to the right column of shop buttons. These should be added from top to bottom i.e. the top right button is at index 0.

5.19.2.3 delayBetweenAnimations

```
float GameCore.ButtonAnimationController.delayBetweenAnimations = 0.2f
```

The delay between animating each row.

5.19.2.4 OnAnimationComplete

```
Action GameCore.ButtonAnimationController.OnAnimationComplete
```

Invoked when shop has finished animating.

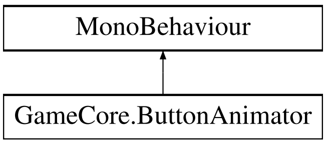
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/UI/ButtonAnimationController.cs

5.20 GameCore.ButtonAnimator Class Reference

Triggers individual shop button animations.

Inheritance diagram for GameCore.ButtonAnimator:



Public Member Functions

- void Animate ()

  Animate this instance.

5.20.1 Detailed Description

Triggers individual shop button animations.

5.20.2 Member Function Documentation

5.20.2.1 Animate()

```
void GameCore.ButtonAnimator.Animate ( )
```

Animate this instance.

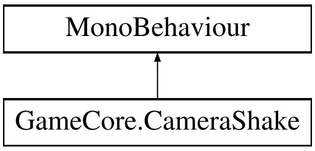
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/UI/ButtonAnimator.cs

5.21 GameCore.CameraShake Class Reference

Camera shake based on perlin noise.

Inheritance diagram for GameCore.CameraShake:



Public Member Functions

- void Begin (float duration, float magnitude)

  Begin this instance. Moves the camera round based on perlin noise, centred around cameras current position.

Public Attributes

- float globalMagDampener = 0.8f

  The global magnitude dampener. Any requested camera shake magnitude is multipled by this.
- float globalDurDampener = 1f

  The global duration dampener. Any requested camera shake duration is multipled by this.

5.21.1 Detailed Description

Camera shake based on perlin noise.

5.21.2 Member Function Documentation

5.21.2.1 Begin()

```
void GameCore.CameraShake.Begin (
             float duration,
             float magnitude )
```

Begin this instance. Moves the camera round based on perlin noise, centred around cameras current position.

5.21.3 Member Data Documentation

5.21.3.1 globalDurDampener

```
float GameCore.CameraShake.globalDurDampener = 1f
```

The global duration dampener. Any requested camera shake duration is multipled by this.

5.21.3.2 globalMagDampener

```
float GameCore.CameraShake.globalMagDampener = 0.8f
```

The global magnitude dampener. Any requested camera shake magnitude is multipled by this.

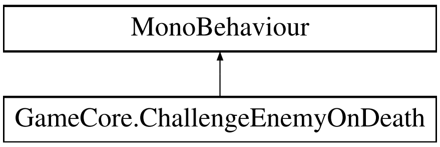
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Utilities/CameraShake.cs

5.22 GameCore.ChallengeEnemyOnDeath Class Reference

Handles animation and particle emission on challenge enemies death.

Inheritance diagram for GameCore.ChallengeEnemyOnDeath:



Public Attributes

- Color particleColour

    The particle colours to spew on death.

5.22.1 Detailed Description

Handles animation and particle emission on challenge enemies death.

5.22.2 Member Data Documentation

5.22.2.1 particleColour

```
Color GameCore.ChallengeEnemyOnDeath.particleColour
```

The particle colours to spew on death.

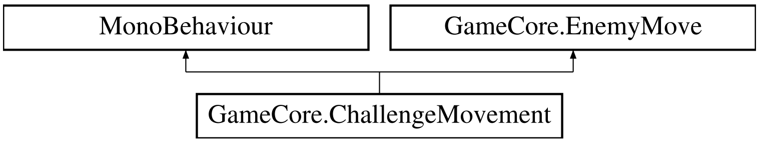
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/ChallengeEnemyOnDeath.cs

5.23 GameCore.ChallengeMovement Class Reference

Controls the challenge enemies movement.

Inheritance diagram for GameCore.ChallengeMovement:



Public Member Functions

- void Begin ()

    Begin this instance. Starts fade in.
- void Pause ()

    Pause this instance.
- void Resume ()

    Resume this instance.

Public Attributes

- MovementDirection moveDirection = MovementDirection.Left

  The movement direction.
- float moveSpeed = 3f

  The movement speed.
- bool oscillateY = false

  Bounce the charact on the y axis as it moves across the screen.
- Action onEscapedWave

  Invoked when enemy escapes wave.
- float startDelay = 0f

  The delay before the enemy starts moving.

5.23.1 Detailed Description

Controls the challenge enemies movement.

5.23.2 Member Function Documentation

5.23.2.1 Begin()

```
void GameCore.ChallengeMovement.Begin ( )
```

Begin this instance. Starts fade in.

Implements GameCore.EnemyMove.

5.23.2.2 Pause()

```
void GameCore.ChallengeMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.23.2.3 Resume()

```
void GameCore.ChallengeMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.23.3 Member Data Documentation

5.23.3.1 moveDirection

```
MovementDirection GameCore.ChallengeMovement.moveDirection = MovementDirection.Left
```

The movement direction.

5.23.3.2 moveSpeed

```
float GameCore.ChallengeMovement.moveSpeed = 3f
```

The movement speed.

5.23.3.3 onEscapedWave

```
Action GameCore.ChallengeMovement.onEscapedWave
```

Invoked when enemy escapes wave.

5.23.3.4 oscillateY

```
bool GameCore.ChallengeMovement.oscillateY = false
```

Bounce the charact on the y axis as it moves across the screen.

5.23.3.5 startDelay

```
float GameCore.ChallengeMovement.startDelay = 0f
```

The delay before the enemy starts moving.

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

- Pew Pew/Scripts/Enemies/Movement/ChallengeMovement.cs

5.24 PE2D.CircularArray< T > Class Template Reference

Simplified version of the circular buffer found at: `http://geekswithblogs.net/blackrob/archive/2014/09/01/ci` `aspx`. Generic storage, used to store particles.

Public Member Functions

- CircularArray (int capacity)

  Initializes a new instance of the PE2D.CircularArray'1 class.

Properties

- int Start `[get, set]`

  Pointer to first entry in array. Note this will not usually be 0.
- int Count `[get, set]`

  Current object count.
- int Capacity `[get]`

  Total object count.
- bool reachedCapacity `[get]`

  Gets a value indicating whether this PE2D.CircularArray'1 has reached capacity.
- T this[int i] `[get, set]`

  Gets or sets the PE2D.CircularArray'1 with the specified i.

5.24.1 Detailed Description

Simplified version of the circular buffer found at: `http://geekswithblogs.net/blackrob/archive/2014/09/01/ci` `aspx`. Generic storage, used to store particles.

5.24.2 Constructor & Destructor Documentation

5.24.2.1 CircularArray()

```
PE2D.CircularArray< T >.CircularArray (
            int capacity )
```

Initializes a new instance of the PE2D.CircularArray'1 class.

Parameters

capacity	Capacity.

5.24.3 Property Documentation

5.24.3.1 Capacity

```
int PE2D.CircularArray< T >.Capacity  [get]
```

Total object count.

The capacity.

5.24.3.2 Count

```
int PE2D.CircularArray< T >.Count  [get], [set]
```

Current object count.

The count.

5.24.3.3 reachedCapacity

```
bool PE2D.CircularArray< T >.reachedCapacity  [get]
```

Gets a value indicating whether this PE2D.CircularArray‘1 has reached capacity.

`true` if reached capacity; otherwise, `false`.

5.24.3.4 Start

```
int PE2D.CircularArray< T >.Start  [get], [set]
```

Pointer to first entry in array. Note this will not usually be 0.

The start.

5.24.3.5 this[int i]

```
T PE2D.CircularArray< T >.this[int i]  [get], [set]
```

Gets or sets the PE2D.CircularArray‘1 with the specified i.

Parameters

i	The index.

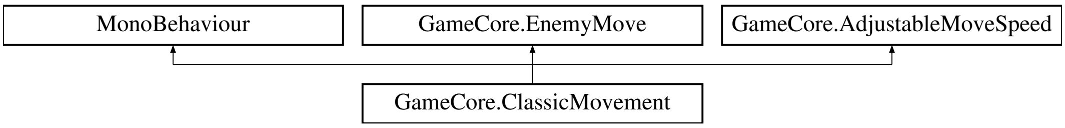
The documentation for this class was generated from the following file:

- pe2d/Helper/CircularArray.cs

5.25 GameCore.ClassicMovement Class Reference

Controls the classic enemy movement.

Inheritance diagram for GameCore.ClassicMovement:



Public Member Functions

- void Begin ()

  Begin this instance.
- void IncrementSpeed ()

  Increments the speed and yDrop amount.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.

Public Attributes

- float moveSpeed = 10f

  The movement speed.
- float moveSpeedInc = 2f

  The amount the movement speed is incremented when reaching end of round.
- MovementDirection initialMoveDirection

  The initial move direction.
- float yDrop = 0.1f

  The distance to drop by each time the enemy row reaches the screen edge.

5.25.1 Detailed Description

Controls the classic enemy movement.

5.25.2 Member Function Documentation

5.25.2.1 Begin()

```
void GameCore.ClassicMovement.Begin ( )
```

Begin this instance.

Implements GameCore.EnemyMove.

5.25.2.2 IncrementSpeed()

```
void GameCore.ClassicMovement.IncrementSpeed ( )
```

Increments the speed and yDrop amount.

Implements GameCore.AdjustableMoveSpeed.

5.25.2.3 Pause()

```
void GameCore.ClassicMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.25.2.4 Resume()

```
void GameCore.ClassicMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.25.3 Member Data Documentation

5.25.3.1 initialMoveDirection

```
MovementDirection GameCore.ClassicMovement.initialMoveDirection
```

The initial move direction.

5.25.3.2 moveSpeed

```
float GameCore.ClassicMovement.moveSpeed = 10f
```

The movement speed.

5.25.3.3 moveSpeedInc

```
float GameCore.ClassicMovement.moveSpeedInc = 2f
```

The amount the movement speed is incremented when reaching end of round.

5.25.3.4 yDrop

```
float GameCore.ClassicMovement.yDrop = 0.1f
```

The distance to drop by each time the enemy row reaches the screen edge.

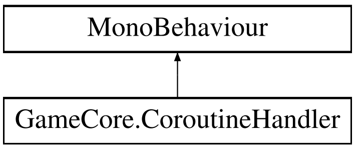
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/ClassicMovement.cs

5.26 GameCore.CoroutineHandler Class Reference

Provides access to coroutines for instances that do not inherit form monobehaviour.

Inheritance diagram for GameCore.CoroutineHandler:



Public Member Functions

- void RunCoroutine (IEnumerator routine)

  Runs the coroutine.

5.26.1 Detailed Description

Provides access to coroutines for instances that do not inherit form monobehaviour.

5.26.2 Member Function Documentation

5.26.2.1 RunCoroutine()

```
void GameCore.CoroutineHandler.RunCoroutine (
             IEnumerator routine )
```

Runs the coroutine.

Parameters

routine	Coroutine to execute.

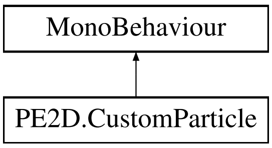
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Utilities/CoroutineHandler.cs

5.27 PE2D.CustomParticle Class Reference

Main workhorse for the custom particles. Updates particles state (colour, position, velocity etc), handles interaction with effectors, and applys any screen constraints.

Inheritance diagram for PE2D.CustomParticle:



Public Member Functions

- void **Pause** ()
- void **Resume** ()

Static Public Member Functions

- static void UpdateEffectorList ()

  Finds all effectors in scene. Static reference should only be called once for all particles on effector change.

Public Attributes

- bool shouldUpdateAlpha = true

  Update sprites alpha based on velovity.
- bool shouldUpdateScale = true

  Update sprites scale based on velicoty.

Properties

- bool **ignoreEffectors** `[set]`
- bool **canBeCollectedByPlayer** `[set]`
- ParticleBuilder state `[get, set]`

  Set the state of the particles. Also resets particles properties.
- float duration `[get, set]`

  Maximum duration of particles life. Life may be shorter dependent on velocity.
- float percentLife `[get, set]`

  Range (0, 1). 0 = time to remove from scene, 1 = just spawned.
- SpriteRenderer spriteRenderer `[get]`

  Gets the sprite renderer.

5.27.1 Detailed Description

Main workhorse for the custom particles. Updates particles state (colour, position, velocity etc), handles interaction with effectors, and applys any screen constraints.

5.27.2 Member Function Documentation

5.27.2.1 UpdateEffectorList()

```
static void PE2D.CustomParticle.UpdateEffectorList ( )  [static]
```

Finds all effectors in scene. Static reference should only be called once for all particles on effector change.

5.27.3 Member Data Documentation

5.27.3.1 shouldUpdateAlpha

```
bool PE2D.CustomParticle.shouldUpdateAlpha = true
```

Update sprites alpha based on velovity.

5.27.3.2 shouldUpdateScale

```
bool PE2D.CustomParticle.shouldUpdateScale = true
```

Update sprites scale based on velicoty.

5.27.4 Property Documentation

5.27.4.1 duration

```
float PE2D.CustomParticle.duration  [get], [set]
```

Maximum duration of particles life. Life may be shorter dependent on velocity.

The duration.

5.27.4.2 percentLife

```
float PE2D.CustomParticle.percentLife  [get], [set]
```

Range (0, 1). 0 = time to remove from scene, 1 = just spawned.

The percent life.

5.27.4.3 spriteRenderer

```
SpriteRenderer PE2D.CustomParticle.spriteRenderer  [get]
```

Gets the sprite renderer.

The sprite renderer.

5.27.4.4 state

```
ParticleBuilder PE2D.CustomParticle.state  [get], [set]
```

Set the state of the particles. Also resets particles properties.

The state.

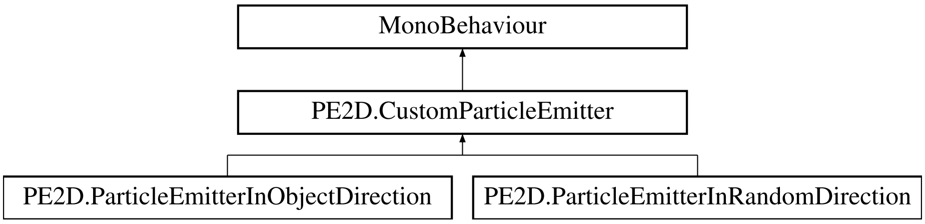
The documentation for this class was generated from the following file:

- pe2d/Particles/CustomParticle.cs

5.28 PE2D.CustomParticleEmitter Class Reference

Base class for ParticleEmitterInRandomDirection and ParticleEmitterInObjectDirection. Add base classes to GameObjects to easily create particle emitters.

Inheritance diagram for PE2D.CustomParticleEmitter:


MonoBehaviour
PE2D.CustomParticleEmitter
PE2D.ParticleEmitterInObjectDirection
PE2D.ParticleEmitterInRandomDirection

Public Member Functions

- void TurnOn ()

  Enables particle emission from this object.
- void TurnOff ()

  Disables particle emission from this object.

Public Attributes

- float timeBetweenProjectileRelease = 0f

  The time between projectile release, if equals 0 then particle is released with each call to update.
- Vector2 initialScale = new Vector2(2f, 1f)

  Initial scale of the particles released. Scale is also dependent on velocity.
- bool particlesEnabled = true

  Turns on/off particle generation from this GameObject.
- float duration = 90f

  The maximum duration for each particle. A particles life is also dependent on velocity.
- float velocityDampener = 0.94f

  THe rate at which to reduce particles velocity each time step.
- float lengthMultiplier = 40f

  The length multiplier for the particles.
- WrapAroundType wrapAround = WrapAroundType.None

  The screen constraint type.
- bool randomColour = false

  Particle will spawn as a random colour when enabled.
- Color particleColour

  Set the particles colour.
- bool clampMinLength

  Clamp the minimum length of a particle.
- float minLength

  The minimum length of a particle, only used if clampMinLength = true.
- bool clampMaxLength

  Clamp the maximum length of a particle.
- float maxLength

  The minimum length of a particle, only used if clampMaxLength = true.
- bool removeWhenVelocityReachesThreshold

  Will remove a particle if velocity reaches a threshold.
- float customVelocityThreshold

  The velocity at which a particle will be removed, only used if removeWhenVelocityReachesThreshold = true.
- bool removeWhenAlphaReachesThreshold

  Will remove the particle when its alpha reaches a specified threshold.
- float customAlphaThreshold

  The particles sprites alpha threshold at which a particle will be removed, only used if removeWhenAlphaReachesThreshold = true.

Protected Member Functions

- Color **GetRandomColour** ()
- abstract void **ReleaseParticle** ()

Protected Attributes

- ParticleBuilder **_cachedState**

5.28.1 Detailed Description

Base class for ParticleEmitterInRandomDirection and ParticleEmitterInObjectDirection. Add base classes to GameObjects to easily create particle emitters.

5.28.2 Member Function Documentation

5.28.2.1 TurnOff()

```
void PE2D.CustomParticleEmitter.TurnOff ( )
```

Disables particle emission from this object.

5.28.2.2 TurnOn()

```
void PE2D.CustomParticleEmitter.TurnOn ( )
```

Enables particle emission from this object.

5.28.3 Member Data Documentation

5.28.3.1 clampMaxLength

```
bool PE2D.CustomParticleEmitter.clampMaxLength
```

Clamp the maximum length of a particle.

5.28.3.2 clampMinLength

```
bool PE2D.CustomParticleEmitter.clampMinLength
```

Clamp the minimum length of a particle.

5.28.3.3 customAlphaThreshold

```
float PE2D.CustomParticleEmitter.customAlphaThreshold
```

The particles sprites alpha threshold at which a particle will be removed, only used if removeWhenAlphaReachesThreshold = true.

5.28.3.4 customVelocityThreshold

```
float PE2D.CustomParticleEmitter.customVelocityThreshold
```

The velocity at which a particle will be removed, only used if removeWhenVelocityReachesThreshold = true.

5.28.3.5 duration

```
float PE2D.CustomParticleEmitter.duration = 90f
```

The maximum duration for each particle. A particles life is also dependent on velocity.

5.28.3.6 initialScale

```
Vector2 PE2D.CustomParticleEmitter.initialScale = new Vector2(2f, 1f)
```

Initial scale of the particles released. Scale is also dependent on velocity.

5.28.3.7 lengthMultiplier

```
float PE2D.CustomParticleEmitter.lengthMultiplier = 40f
```

The length multiplier for the particles.

5.28.3.8 maxLength

```
float PE2D.CustomParticleEmitter.maxLength
```

The minimum length of a particle, only used if clampMaxLength = true.

5.28.3.9 minLength

```
float PE2D.CustomParticleEmitter.minLength
```

The minimum length of a particle, only used if clampMinLength = true.

5.28.3.10 particleColour

```
Color PE2D.CustomParticleEmitter.particleColour
```

Set the particles colour.

5.28.3.11 particlesEnabled

```
bool PE2D.CustomParticleEmitter.particlesEnabled = true
```

Turns on/off particle generation from this GameObject.

5.28.3.12 randomColour

```
bool PE2D.CustomParticleEmitter.randomColour = false
```

Particle will spawn as a random colour when enabled.

5.28.3.13 removeWhenAlphaReachesThreshold

```
bool PE2D.CustomParticleEmitter.removeWhenAlphaReachesThreshold
```

Will remove the particle when its alpha reaches a specified threshold.

5.28.3.14 removeWhenVelocityReachesThreshold

```
bool PE2D.CustomParticleEmitter.removeWhenVelocityReachesThreshold
```

Will remove a particle if velocity reaches a threshold.

5.28.3.15 timeBetweenProjectileRelease

```
float PE2D.CustomParticleEmitter.timeBetweenProjectileRelease = 0f
```

The time between projectile release, if equals 0 then particle is released with each call to update.

5.28.3.16 velocityDampener

```
float PE2D.CustomParticleEmitter.velocityDampener = 0.94f
```

THe rate at which to reduce particles velocity each time step.

5.28.3.17 wrapAround

```
WrapAroundType PE2D.CustomParticleEmitter.wrapAround = WrapAroundType.None
```

The screen constraint type.

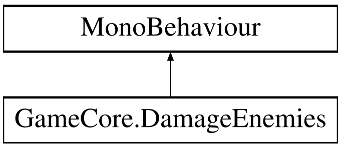
The documentation for this class was generated from the following file:

- pe2d/Particles/Emitters/CustomParticleEmitter.cs

5.29 GameCore.DamageEnemies Class Reference

Damages enemies on trigger enter.

Inheritance diagram for GameCore.DamageEnemies:



5.29.1 Detailed Description

Damages enemies on trigger enter.

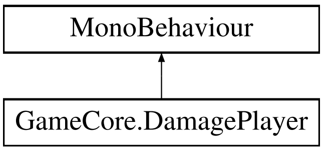
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Damage/DamageEnemies.cs

5.30 GameCore.DamagePlayer Class Reference

Damages player on trigger enter.

Inheritance diagram for GameCore.DamagePlayer:



5.30.1 Detailed Description

Damages player on trigger enter.

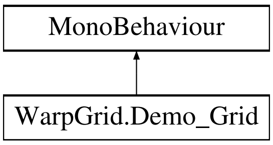
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Damage/DamagePlayer.cs

5.31 WarpGrid.Demo_Grid Class Reference

Used to demonstrate how to apply a force to an existing grid.

Inheritance diagram for WarpGrid.Demo_Grid:



Public Attributes

- Grid grid

    The grid to apply force to.

5.31.1 Detailed Description

Used to demonstrate how to apply a force to an existing grid.

5.31.2 Member Data Documentation

5.31.2.1 grid

```
Grid WarpGrid.Demo_Grid.grid
```

The grid to apply force to.

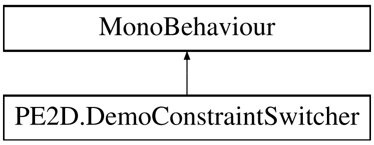
The documentation for this class was generated from the following file:

- UG/Scripts/Demo_Grid.cs

5.32 PE2D.DemoConstraintSwitcher Class Reference

Switches between screen constraints in the demo scene.

Inheritance diagram for PE2D.DemoConstraintSwitcher:



Public Attributes

- DemoMouseController **mouseController**
- Text **constraintText**

5.32.1 Detailed Description

Switches between screen constraints in the demo scene.

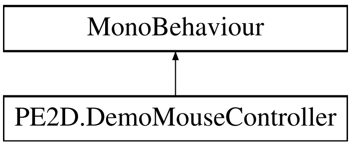
The documentation for this class was generated from the following file:

- pe2d/Demo/DemoConstraintSwitcher.cs

5.33 PE2D.DemoMouseController Class Reference

Spawns a circular explosion of particles on mouse click. Example of how to procedurally create particles.

Inheritance diagram for PE2D.DemoMouseController:



Public Attributes

- float **speedOffset** = .01f
- float **lengthMultiplier** = 40f
- int **numToSpawn** = 200
- WrapAroundType **wrapAround**

5.33.1 Detailed Description

Spawns a circular explosion of particles on mouse click. Example of how to procedurally create particles.

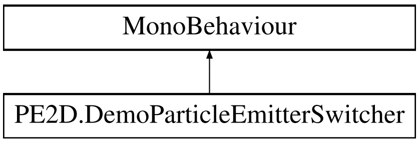
The documentation for this class was generated from the following file:

- pe2d/Demo/DemoMouseController.cs

5.34 PE2D.DemoParticleEmitterSwitcher Class Reference

Switches between particle emitters in demo scene.

Inheritance diagram for PE2D.DemoParticleEmitterSwitcher:



Public Attributes

- GameObject [ ] **particleEmitters**
- Text **emitterText**
- string **preEmitterString**
- string **postEmitterString**
- bool **updateEffectorsOnChange** = false

5.34.1 Detailed Description

Switches between particle emitters in demo scene.

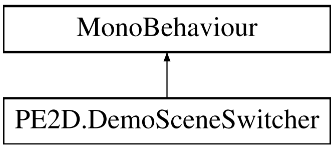
The documentation for this class was generated from the following file:

- pe2d/Demo/DemoParticleEmitterSwitcher.cs

5.35 PE2D.DemoSceneSwitcher Class Reference

Switches between demo scenes when enter key pressed.

Inheritance diagram for PE2D.DemoSceneSwitcher:



Public Attributes

- int **numberOfScenes** = 3

5.35.1 Detailed Description

Switches between demo scenes when enter key pressed.

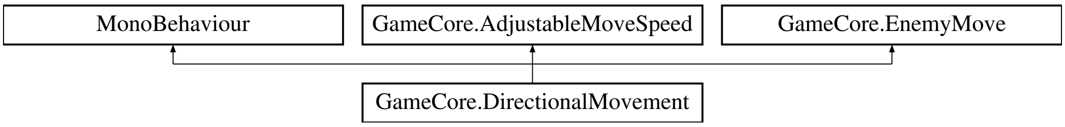
The documentation for this class was generated from the following file:

- pe2d/Demo/DemoSceneSwitcher.cs

5.36 GameCore.DirectionalMovement Class Reference

Controls the directional enemies.

Inheritance diagram for GameCore.DirectionalMovement:



Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void IncrementSpeed ()

  Increments the speed near round end.

Public Attributes

- float moveSpeed = 5f

  The movement speed.
- float moveSpeedIncrement = 2f

  The amount to increment the movement speed on round end.
- MovementDirection [ ] moveDirections

  The move directions. The enemy is moved in these directions in turn.
- float moveOffset = 2f

  The movement offset. The enemy is moved by this amount each time.
- float pauseOnTargetReach = 0.5f

  The time in seconds the enemy pauses when it reaches its destination before moving in the next direction.
- int numOfProjectilesToRequest = 1

  The number of projectiles to request when shooting.
- float rotateSpeed = 120.0f

  The rotatation speed.
- float delayedStart = 0f

  The seconds delay before an enemy starts moving.

5.36.1 Detailed Description

Controls the directional enemies.

5.36.2 Member Function Documentation

5.36.2.1 Begin()

```
void GameCore.DirectionalMovement.Begin ( )
```

Begin this instance. Starts fade in.

Implements GameCore.EnemyMove.

5.36.2.2 IncrementSpeed()

```
void GameCore.DirectionalMovement.IncrementSpeed ( )
```

Increments the speed near round end.

Implements GameCore.AdjustableMoveSpeed.

5.36.2.3 Pause()

```
void GameCore.DirectionalMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.36.2.4 Resume()

```
void GameCore.DirectionalMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.36.3 Member Data Documentation

5.36.3.1 delayedStart

```
float GameCore.DirectionalMovement.delayedStart = 0f
```

The seconds delay before an enemy starts moving.

5.36.3.2 moveDirections

```
MovementDirection [] GameCore.DirectionalMovement.moveDirections
```

The move directions. The enemy is moved in these directions in turn.

5.36.3.3 moveOffset

```
float GameCore.DirectionalMovement.moveOffset = 2f
```

The movement offset. The enemy is moved by this amount each time.

5.36.3.4 moveSpeed

```
float GameCore.DirectionalMovement.moveSpeed = 5f
```

The movement speed.

5.36.3.5 moveSpeedIncrement

```
float GameCore.DirectionalMovement.moveSpeedIncrement = 2f
```

The amount to increment the movement speed on round end.

5.36.3.6 numOfProjectilesToRequest

```
int GameCore.DirectionalMovement.numOfProjectilesToRequest = 1
```

The number of projectiles to request when shooting.

5.36.3.7 pauseOnTargetReach

```
float GameCore.DirectionalMovement.pauseOnTargetReach = 0.5f
```

The time in seconds the enemy pauses when it reaches its destination before moving in the next direction.

5.36.3.8 rotateSpeed

```
float GameCore.DirectionalMovement.rotateSpeed = 120.0f
```

The rotatation speed.

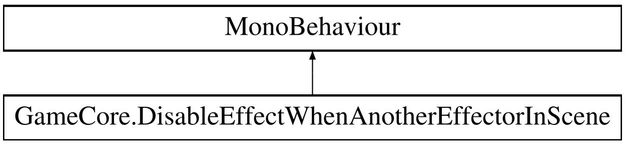
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/DirectionalMovement.cs

5.37 GameCore.DisableEffectWhenAnotherEffectorInScene Class Reference

Disables this particle effector when another effector is present in scene.

Inheritance diagram for GameCore.DisableEffectWhenAnotherEffectorInScene:



Public Member Functions

- void EnableEffector ()

  Enables the effector.
- void DisableEffector ()

  Disables the effector.

Public Attributes

- ParticleEffector particleEffector

  The particle effector to disable.

5.37.1 Detailed Description

Disables this particle effector when another effector is present in scene.

5.37.2 Member Function Documentation

5.37.2.1 DisableEffector()

```
void GameCore.DisableEffectWhenAnotherEffectorInScene.DisableEffector ( )
```

Disables the effector.

5.37.2.2 EnableEffector()

```
void GameCore.DisableEffectWhenAnotherEffectorInScene.EnableEffector ( )
```

Enables the effector.

5.37.3 Member Data Documentation

5.37.3.1 particleEffector

```
ParticleEffector GameCore.DisableEffectWhenAnotherEffectorInScene.particleEffector
```

The particle effector to disable.

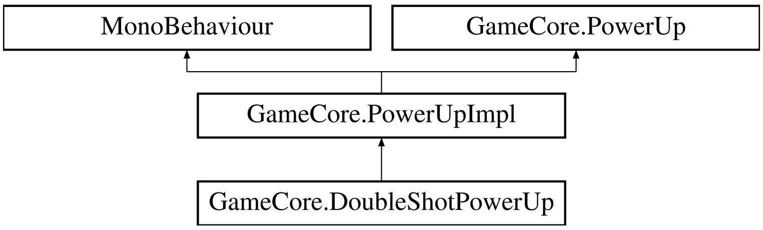
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Effectors/DisableEffectWhenAnotherEffectorInScene.cs

5.38 GameCore.DoubleShotPowerUp Class Reference

Provides ability for player to shoot two projectiles in parallel for a specified amount of time.

Inheritance diagram for GameCore.DoubleShotPowerUp:



Public Member Functions

- override void Perform (Transform player)

  Perform the specified powerup action. Finds main player shoot module and invokes PlayerShoot::DoubleShootingForSeconds.

Public Attributes

- float secPowerUp = 3f

  Powerup duration.

Additional Inherited Members

5.38.1 Detailed Description

Provides ability for player to shoot two projectiles in parallel for a specified amount of time.

5.38.2 Member Function Documentation

5.38.2.1 Perform()

```
override void GameCore.DoubleShotPowerUp.Perform (
             Transform player )  [virtual]
```

Perform the specified powerup action. Finds main player shoot module and invokes PlayerShoot::DoubleShootingForSeconds.

Parameters

player	Player tranform.

Implements GameCore.PowerUpImpl.

5.38.3 Member Data Documentation

5.38.3.1 secPowerUp

```
float GameCore.DoubleShotPowerUp.secPowerUp = 3f
```

Powerup duration.

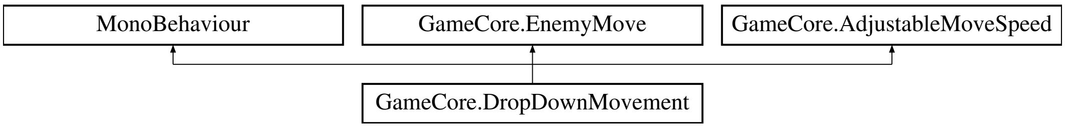
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/DoubleShotPowerUp.cs

5.39 GameCore.DropDownMovement Class Reference

Controls the drop down enemymovement.

Inheritance diagram for GameCore.DropDownMovement:



Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void IncrementSpeed ()

  Increments the drop, twitch, and bounce up speed.

Public Attributes

- float twitchSpeed

  The speed the enemy twitches while waiting to drop down.
- float twitchSpeedInc

  The amount to increment twitch speed when near round end.
- float twitchRange = 1.1f

  The twitch range. The enemy moves with a radius of this size when twitching.
- Vector2 minMaxSecsBetweenDrop

  The minimum and maximum seconds between dropping down.
- float dropSpeed

  The initial speed at which the enemy drops down.
- float dropSpeedInc

  The amount to increment the movement speed on the enemy.
- float dropSpeedUp = 15f

  The speed at which the enemy drops down.
- float bounceUpSpeed

  The speed the enemy bounces up.
- float bounceUpSpeedInc

  The amount to increase the enemies bounce up speed.
- float bounceUpDistance = 0.6f

  The distance to move up before falling down.

5.39.1 Detailed Description

Controls the drop down enemymovement.

5.39.2 Member Function Documentation

5.39.2.1 Begin()

```
void GameCore.DropDownMovement.Begin ( )
```

Begin this instance. Starts fade in.

Implements GameCore.EnemyMove.

5.39.2.2 IncrementSpeed()

```
void GameCore.DropDownMovement.IncrementSpeed ( )
```

Increments the drop, twitch, and bounce up speed.

Implements GameCore.AdjustableMoveSpeed.

5.39.2.3 Pause()

```
void GameCore.DropDownMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.39.2.4 Resume()

```
void GameCore.DropDownMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.39.3 Member Data Documentation

5.39.3.1 bounceUpDistance

```
float GameCore.DropDownMovement.bounceUpDistance = 0.6f
```

The distance to move up before falling down.

5.39.3.2 bounceUpSpeed

```
float GameCore.DropDownMovement.bounceUpSpeed
```

The speed the enemy bounces up.

5.39.3.3 bounceUpSpeedInc

```
float GameCore.DropDownMovement.bounceUpSpeedInc
```

The amount to increase the enemies bounce up speed.

5.39.3.4 dropSpeed

```
float GameCore.DropDownMovement.dropSpeed
```

The initial speed at which the enemy drops down.

5.39.3.5 dropSpeedInc

```
float GameCore.DropDownMovement.dropSpeedInc
```

The amount to increment the movement speed on the enemy.

5.39.3.6 dropSpeedUp

```
float GameCore.DropDownMovement.dropSpeedUp = 15f
```

The speed at which the enemy drops down.

5.39.3.7 minMaxSecsBetweenDrop

```
Vector2 GameCore.DropDownMovement.minMaxSecsBetweenDrop
```

The minimum and maximum seconds between dropping down.

5.39.3.8 twitchRange

```
float GameCore.DropDownMovement.twitchRange = 1.1f
```

The twitch range. The enemy moves with a radius of this size when twitching.

5.39.3.9 twitchSpeed

```
float GameCore.DropDownMovement.twitchSpeed
```

The speed the enemy twitches while waiting to drop down.

5.39.3.10 twitchSpeedInc

```
float GameCore.DropDownMovement.twitchSpeedInc
```

The amount to increment twitch speed when near round end.

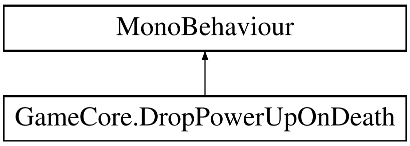
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/DropDownMovement.cs

5.40 GameCore.DropPowerUpOnDeath Class Reference

Spawns a powerup when this an entity with this script attached dies.

Inheritance diagram for GameCore.DropPowerUpOnDeath:



Public Attributes

- PowerUpSpawn [ ] powerUps
  The power ups that can be spawned. Consists of prefab and weighted spawn chance.

5.40.1 Detailed Description

Spawns a powerup when this an entity with this script attached dies.

5.40.2 Member Data Documentation

5.40.2.1 powerUps

```
PowerUpSpawn [ ] GameCore.DropPowerUpOnDeath.powerUps
```

The power ups that can be spawned. Consists of prefab and weighted spawn chance.

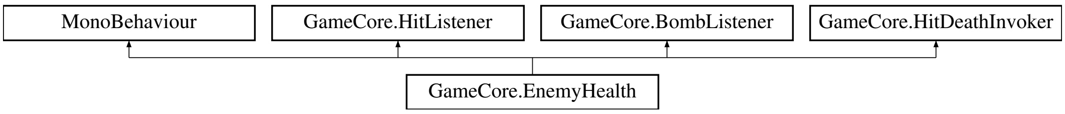
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/DropPowerUpOnDeath.cs

5.41 GameCore.EnemyHealth Class Reference

Controls enemies health and taking damage.

Inheritance diagram for GameCore.EnemyHealth:



Public Member Functions

- void ExplosionInRange (int damage)

    Apply damage from explosion.
- void OnHit (int damage)

    Raises the hit event. Damages entity.
- void Kill (bool cameraShake)

    Kill the entity.
- void PlayOnDeathAudio ()

    Plays the on death audio.

Public Attributes

- int hitPoints = 1

    The number of hit points for the enemy.
- AudioClip audioOnDeath

    An AudioClip to play when the enemy is destroyed.
- AudioClip audioOnDamage

    An AudioClip to play when the enemy takes damage.
- float percentageScaleDownWhenHit = 20f

    The percentage to scale down when hit.
- float destroyWhenBelowY = -40f

    Sets whether this instance should be destroyed when below this Y value.
- float camShakeMag = 0.1f

    The magnitude of the camera shake to apply on death.
- float camShakeSec = 0.1f

    The time in seconds of the camera shake on death.
- Color particleColour

    The colour of the particles to spew on death.
- float explosiveForceMulti = 1f

    A multiplier to apply to the Grid when entity is destroyed.
- Action onDestroyHook

    If not null, replaces the destroy method when enemy killed.
- int numOfParticlesOnDeath = 20

    The number of particles to spew on death.
- int numOfParticlesOnHit = 10

    The number of particles to spew when damaged.

Properties

- Action onHit `[get, set]`

  Gets or sets an action to perform on hit.
- Action onDeath `[get, set]`

  Gets or sets an action to perform on death.
- Transform owner `[get]`

  Gets the owner.

5.41.1 Detailed Description

Controls enemies health and taking damage.

5.41.2 Member Function Documentation

5.41.2.1 ExplosionInRange()

```
void GameCore.EnemyHealth.ExplosionInRange (
             int damage )
```

Apply damage from explosion.

Parameters

damage	Damage to apply.

Implements GameCore.BombListener.

5.41.2.2 Kill()

```
void GameCore.EnemyHealth.Kill (
             bool cameraShake )
```

Kill the entity.

Parameters

cameraShake	If set to `true` camera shake is applied.

5.41.2.3 OnHit()

```
void GameCore.EnemyHealth.OnHit (
             int damage )
```

Raises the hit event. Damages entity.

Parameters

damage	Damage taken.

Implements GameCore.HitListener.

5.41.2.4 PlayOnDeathAudio()

```
void GameCore.EnemyHealth.PlayOnDeathAudio ( )
```

Plays the on death audio.

5.41.3 Member Data Documentation

5.41.3.1 audioOnDamage

```
AudioClip GameCore.EnemyHealth.audioOnDamage
```

An AudioClip to play when the enemy takes damage.

5.41.3.2 audioOnDeath

```
AudioClip GameCore.EnemyHealth.audioOnDeath
```

An AudioClip to play when the enemy is destroyed.

5.41.3.3 camShakeMag

```
float GameCore.EnemyHealth.camShakeMag = 0.1f
```

The magnitude of the camera shake to apply on death.

5.41.3.4 camShakeSec

```
float GameCore.EnemyHealth.camShakeSec = 0.1f
```

The time in seconds of the camera shake on death.

5.41.3.5 destroyWhenBelowY

```
float GameCore.EnemyHealth.destroyWhenBelowY = -40f
```

Sets whether this instance should be destroyed when below this Y value.

5.41.3.6 explosiveForceMulti

```
float GameCore.EnemyHealth.explosiveForceMulti = 1f
```

A multiplier to apply to the Grid when entity is destroyed.

5.41.3.7 hitPoints

```
int GameCore.EnemyHealth.hitPoints = 1
```

The number of hit points for the enemy.

5.41.3.8 numOfParticlesOnDeath

```
int GameCore.EnemyHealth.numOfParticlesOnDeath = 20
```

The number of particles to spew on death.

5.41.3.9 numOfParticlesOnHit

```
int GameCore.EnemyHealth.numOfParticlesOnHit = 10
```

The number of particles to spew when damaged.

5.41.3.10 onDestroyHook

```
Action GameCore.EnemyHealth.onDestroyHook
```

If not null, replaces the destroy method when enemy killed.

5.41.3.11 particleColour

```
Color GameCore.EnemyHealth.particleColour
```

The colour of the particles to spew on death.

5.41.3.12 percentageScaleDownWhenHit

```
float GameCore.EnemyHealth.percentageScaleDownWhenHit = 20f
```

The percentage to scale down when hit.

5.41.4 Property Documentation

5.41.4.1 onDeath

```
Action GameCore.EnemyHealth.onDeath  [get], [set]
```

Gets or sets an action to perform on death.

The on death action.

5.41.4.2 onHit

```
Action GameCore.EnemyHealth.onHit  [get], [set]
```

Gets or sets an action to perform on hit.

The on hit action.

5.41.4.3 owner

```
Transform GameCore.EnemyHealth.owner  [get]
```

Gets the owner.

The owner.

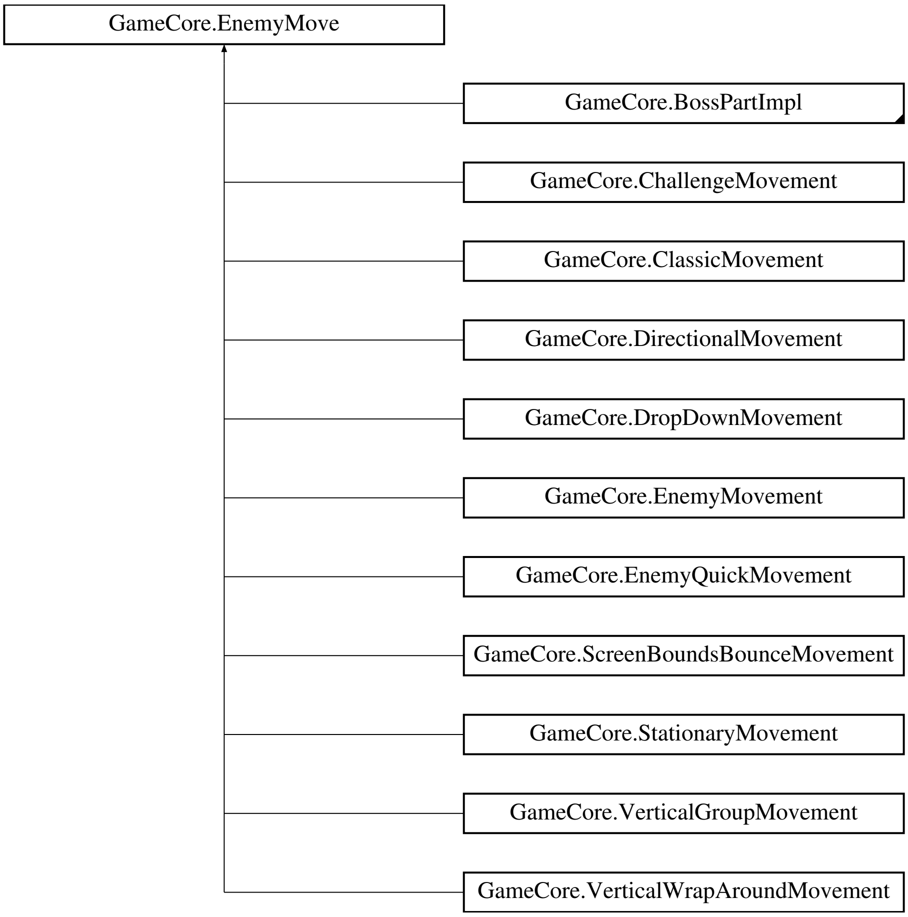
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyHealth.cs

5.42 GameCore.EnemyMove Interface Reference

Contract for all enemies that can begin, pause, and resume actions.

Inheritance diagram for GameCore.EnemyMove:


GameCore.EnemyMove
GameCore.BossPartImpl
GameCore.ChallengeMovement
GameCore.ClassicMovement
GameCore.DirectionalMovement
GameCore.DropDownMovement
GameCore.EnemyMovement
GameCore.EnemyQuickMovement
GameCore.ScreenBoundsBounceMovement
GameCore.StationaryMovement
GameCore.VerticalGroupMovement
GameCore.VerticalWrapAroundMovement

Public Member Functions

- void **Begin** ()
- void **Pause** ()
- void **Resume** ()

5.42.1 Detailed Description

Contract for all enemies that can begin, pause, and resume actions.

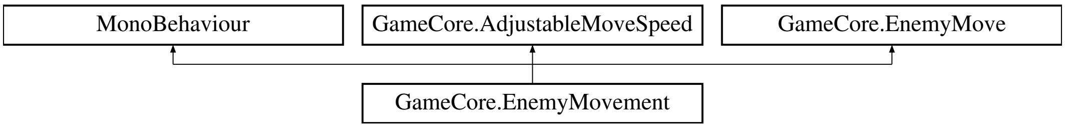
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/EnemyMovement.cs

5.43 GameCore.EnemyMovement Class Reference

Controls standard enemy movement.

Inheritance diagram for GameCore.EnemyMovement:



Public Member Functions

- void Begin ()

  Begin this instance.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void IncrementSpeed ()

  Increments the speed near round end.

Public Attributes

- float moveSpeed = 0.05f

  The movement speed.
- float moveSpeedAdjustment = 0.01f

  The amount to increase movement speed when near round end.
- MovementDirection initialMoveDir = MovementDirection.Left

  The initial movement direction.

5.43.1 Detailed Description

Controls standard enemy movement.

5.43.2 Member Function Documentation

5.43.2.1 Begin()

```
void GameCore.EnemyMovement.Begin ( )
```

Begin this instance.

Implements GameCore.EnemyMove.

5.43.2.2 IncrementSpeed()

```
void GameCore.EnemyMovement.IncrementSpeed ( )
```

Increments the speed near round end.

Implements GameCore.AdjustableMoveSpeed.

5.43.2.3 Pause()

```
void GameCore.EnemyMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.43.2.4 Resume()

```
void GameCore.EnemyMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.43.3 Member Data Documentation

5.43.3.1 initialMoveDir

```
MovementDirection GameCore.EnemyMovement.initialMoveDir = MovementDirection.Left
```

The initial movement direction.

5.43.3.2 moveSpeed

```
float GameCore.EnemyMovement.moveSpeed = 0.05f
```

The movement speed.

5.43.3.3 moveSpeedAdjustment

```
float GameCore.EnemyMovement.moveSpeedAdjustment = 0.01f
```

The amount to increase movement speed when near round end.

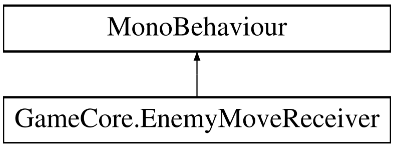
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/EnemyMovement.cs

5.44 GameCore.EnemyMoveReceiver Class Reference

Attach to an enemy object to receive moves from VerticalGroupMovement.

Inheritance diagram for GameCore.EnemyMoveReceiver:



Public Member Functions

- void DoMove (Vector2 move)

  Moves position by amount.

5.44.1 Detailed Description

Attach to an enemy object to receive moves from VerticalGroupMovement.

5.44.2 Member Function Documentation

5.44.2.1 DoMove()

```
void GameCore.EnemyMoveReceiver.DoMove (
             Vector2 move )
```

Moves position by amount.

Parameters

move	Amount to move.

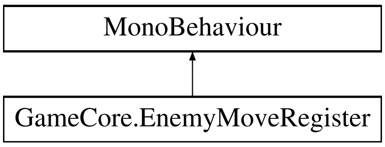
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/EnemyMoveReceiver.cs

5.45 GameCore.EnemyMoveRegister Class Reference

Adds attached EnemyMove to GameManager::EnemyMoves.

Inheritance diagram for GameCore.EnemyMoveRegister:



5.45.1 Detailed Description

Adds attached EnemyMove to GameManager::EnemyMoves.

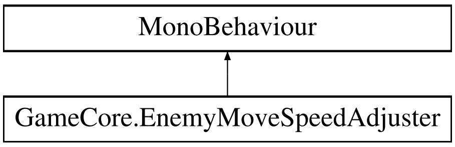
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/EnemyMoveRegister.cs

5.46 GameCore.EnemyMoveSpeedAdjuster Class Reference

Adjusts enemies movement speed near round end.

Inheritance diagram for GameCore.EnemyMoveSpeedAdjuster:



Public Attributes

- int enemeisRemainingPercentInc = 20

  The percentage of enemies remaining to trigger speed increase.
- float bgmPitchIncreaseOnRoundOver = 0.01f

  The amount to increase background audio on round over.

5.46.1 Detailed Description

Adjusts enemies movement speed near round end.

5.46.2 Member Data Documentation

5.46.2.1 bgmPitchIncreaseOnRoundOver

```
float GameCore.EnemyMoveSpeedAdjuster.bgmPitchIncreaseOnRoundOver = 0.01f
```

The amount to increase background audio on round over.

5.46.2.2 enemeisRemainingPercentInc

```
int GameCore.EnemyMoveSpeedAdjuster.enemeisRemainingPercentInc = 20
```

The percentage of enemies remaining to trigger speed increase.

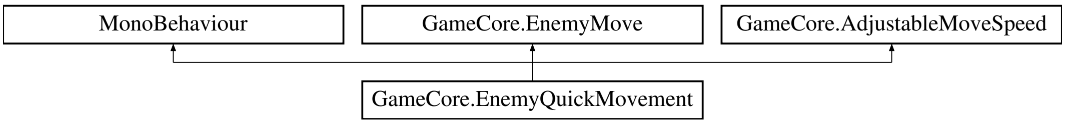
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/EnemyMoveSpeedAdjuster.cs

5.47 GameCore.EnemyQuickMovement Class Reference

Controls enemies quick movement.

Inheritance diagram for GameCore.EnemyQuickMovement:



Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void IncrementSpeed ()

  Increments the speed near round end.

Public Attributes

- float moveSpeed = 10f

  The movement speed.
- float moveSpeedAdjustment = 2f

  The amount to increase movement speed near round end.
- MovementDirection movementDirection = MovementDirection.Left

  The movement direction.
- float xTurnAroundPosition = 0f

  The x position on screen that enemy will rotate.
- float rotateSpeed = 120.0f

  The rotation speed.

5.47.1 Detailed Description

Controls enemies quick movement.

5.47.2 Member Function Documentation

5.47.2.1 Begin()

```
void GameCore.EnemyQuickMovement.Begin ( )
```

Begin this instance. Starts fade in.

Implements GameCore.EnemyMove.

5.47.2.2 IncrementSpeed()

```
void GameCore.EnemyQuickMovement.IncrementSpeed ( )
```

Increments the speed near round end.

Implements GameCore.AdjustableMoveSpeed.

5.47.2.3 Pause()

```
void GameCore.EnemyQuickMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.47.2.4 Resume()

```
void GameCore.EnemyQuickMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.47.3 Member Data Documentation

5.47.3.1 movementDirection

```
MovementDirection GameCore.EnemyQuickMovement.movementDirection = MovementDirection.Left
```

The movement direction.

5.47.3.2 moveSpeed

```
float GameCore.EnemyQuickMovement.moveSpeed = 10f
```

The movement speed.

5.47.3.3 moveSpeedAdjustment

```
float GameCore.EnemyQuickMovement.moveSpeedAdjustment = 2f
```

The amount to increase movement speed near round end.

5.47.3.4 rotateSpeed

```
float GameCore.EnemyQuickMovement.rotateSpeed = 120.0f
```

The rotation speed.

5.47.3.5 xTurnAroundPosition

```
float GameCore.EnemyQuickMovement.xTurnAroundPosition = 0f
```

The x position on screen that enemy will rotate.

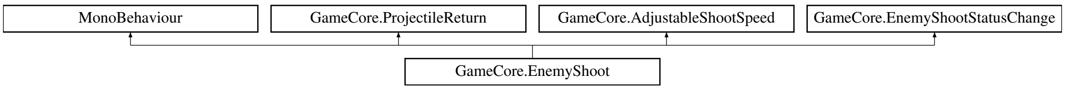
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/EnemyQuickMovement.cs

5.48 GameCore.EnemyShoot Class Reference

Shoots projectiles. Projectiles are pooled.

Inheritance diagram for GameCore.EnemyShoot:


MonoBehaviour
GameCore.ProjectileReturn
GameCore.AdjustableShootSpeed
GameCore.EnemyShootStatusChange
GameCore.EnemyShoot

Public Member Functions

- void Begin ()

  Begin shooting.
- void Pause ()

  Pause shooting.
- void Resume ()

  Resume shooting.
- void StopActivation ()

  Stops the next activation attempt of this module.
- void IncrementSpeed ()

  Decrements the time between shot requests.
- void PoolProjectile (Projectile projectile)

  Pools the projectile.

Public Attributes

- GameObject projectilePrefab

  The projectile prefab to spawn.
- float secsBetweenShot = 1.2f

  The seconds between shoot requests.
- float shootSpeedDecrement = 0.3f

  The amount to decrement time between shoot requests near round end.
- AudioClip audioOnShoot

  The audio to play on shoot.
- float beginShootingWhenBelowScreenY = 100f

  Sets the Y value to begin shooting. Enemies above this value will not shoot.
- int damage = 1

  The damage to apply when projectile hits player.
- Vector2 [ ] shootDirections

  The possible shoot directions.
- bool shootBasedOnRotation = false

  Sets whether projectiles are shot based on owners rotation.

5.48.1 Detailed Description

Shoots projectiles. Projectiles are pooled.

5.48.2 Member Function Documentation

5.48.2.1 Begin()

```
void GameCore.EnemyShoot.Begin ( )
```

Begin shooting.

Implements GameCore.EnemyShootStatusChange.

5.48.2.2 IncrementSpeed()

```
void GameCore.EnemyShoot.IncrementSpeed ( )
```

Decrements the time between shot requests.

Implements GameCore.AdjustableShootSpeed.

5.48.2.3 Pause()

```
void GameCore.EnemyShoot.Pause ( )
```

Pause shooting.

Implements GameCore.EnemyShootStatusChange.

5.48.2.4 PoolProjectile()

```
void GameCore.EnemyShoot.PoolProjectile (
            Projectile projectile )
```

Pools the projectile.

Parameters

projectile	Projectile.

Implements GameCore.ProjectileReturn.

5.48.2.5 Resume()

```
void GameCore.EnemyShoot.Resume ( )
```

Resume shooting.

Implements GameCore.EnemyShootStatusChange.

5.48.2.6 StopActivation()

```
void GameCore.EnemyShoot.StopActivation ( )
```

Stops the next activation attempt of this module.

5.48.3 Member Data Documentation

5.48.3.1 audioOnShoot

```
AudioClip GameCore.EnemyShoot.audioOnShoot
```

The audio to play on shoot.

5.48.3.2 beginShootingWhenBelowScreenY

```
float GameCore.EnemyShoot.beginShootingWhenBelowScreenY = 100f
```

Sets the Y value to begin shooting. Enemies above this value will not shoot.

5.48.3.3 damage

```
int GameCore.EnemyShoot.damage = 1
```

The damage to apply when projectile hits player.

5.48.3.4 projectilePrefab

```
GameObject GameCore.EnemyShoot.projectilePrefab
```

The projectile prefab to spawn.

5.48.3.5 secsBetweenShot

```
float GameCore.EnemyShoot.secsBetweenShot = 1.2f
```

The seconds between shoot requests.

5.48.3.6 shootBasedOnRotation

```
bool GameCore.EnemyShoot.shootBasedOnRotation = false
```

Sets whether projectiles are shot based on owners rotation.

5.48.3.7 shootDirections

```
Vector2 [] GameCore.EnemyShoot.shootDirections
```

The possible shoot directions.

5.48.3.8 shootSpeedDecrement

```
float GameCore.EnemyShoot.shootSpeedDecrement = 0.3f
```

The amount to decrement time between shoot requests near round end.

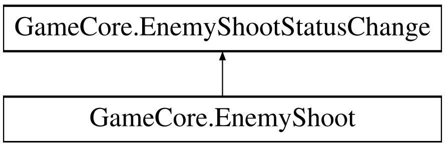
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyShoot.cs

5.49 GameCore.EnemyShootStatusChange Interface Reference

Contract for any entity that can begin, pause, or resume shooting.

Inheritance diagram for GameCore.EnemyShootStatusChange:



Public Member Functions

- void Begin ()

  Begin shooting.
- void Pause ()

  Pause shooting.
- void Resume ()

  Resume shooting.

5.49.1 Detailed Description

Contract for any entity that can begin, pause, or resume shooting.

5.49.2 Member Function Documentation

5.49.2.1 Begin()

```
void GameCore.EnemyShootStatusChange.Begin ( )
```

Begin shooting.

Implemented in GameCore.EnemyShoot.

5.49.2.2 Pause()

```
void GameCore.EnemyShootStatusChange.Pause ( )
```

Pause shooting.

Implemented in GameCore.EnemyShoot.

5.49.2.3 Resume()

```
void GameCore.EnemyShootStatusChange.Resume ( )
```

Resume shooting.

Implemented in GameCore.EnemyShoot.

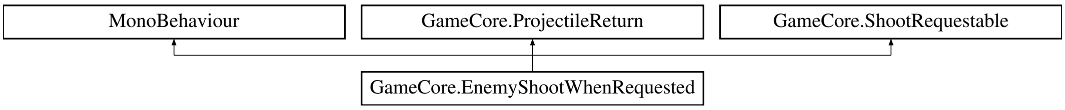
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyShoot.cs

5.50 GameCore.EnemyShootWhenRequested Class Reference

Provides functionality to request projectiles from a pool.

Inheritance diagram for GameCore.EnemyShootWhenRequested:


MonoBehaviour
GameCore.ProjectileReturn
GameCore.ShootRequestable
GameCore.EnemyShootWhenRequested

Public Member Functions

- void RequestShoot ()

  Requests to shoot a projectile. A projectile will be released as long as the pool returns a projectile.
- void PoolProjectile (Projectile projectile)

  Pools the projectile. Returns the projectile to a pool to be reused later.

Public Attributes

- GameObject projectilePrefab

  The projectile prefab.
- AudioClip audioOnShoot

  The audio to play on shoot.
- int damage = 1

  The projectiles damage.
- Vector2 [ ] shootDirections

  The possible shoot directions.
- int numProjectilesToPool = 4

  The number of projectiles to pool.

5.50.1 Detailed Description

Provides functionality to request projectiles from a pool.

5.50.2 Member Function Documentation

5.50.2.1 PoolProjectile()

```
void GameCore.EnemyShootWhenRequested.PoolProjectile (
            Projectile projectile )
```

Pools the projectile. Returns the projectile to a pool to be reused later.

Parameters

projectile	Projectile to disable and place into the pool.

Implements GameCore.ProjectileReturn.

5.50.2.2 RequestShoot()

```
void GameCore.EnemyShootWhenRequested.RequestShoot ( )
```

Requests to shoot a projectile. A projectile will be released as long as the pool returns a projectile.

Implements GameCore.ShootRequestable.

5.50.3 Member Data Documentation

5.50.3.1 audioOnShoot

```
AudioClip GameCore.EnemyShootWhenRequested.audioOnShoot
```

The audio to play on shoot.

5.50.3.2 damage

```
int GameCore.EnemyShootWhenRequested.damage = 1
```

The projectiles damage.

5.50.3.3 numProjectilesToPool

```
int GameCore.EnemyShootWhenRequested.numProjectilesToPool = 4
```

The number of projectiles to pool.

5.50.3.4 projectilePrefab

```
GameObject GameCore.EnemyShootWhenRequested.projectilePrefab
```

The projectile prefab.

5.50.3.5 shootDirections

```
Vector2 [ ] GameCore.EnemyShootWhenRequested.shootDirections
```

The possible shoot directions.

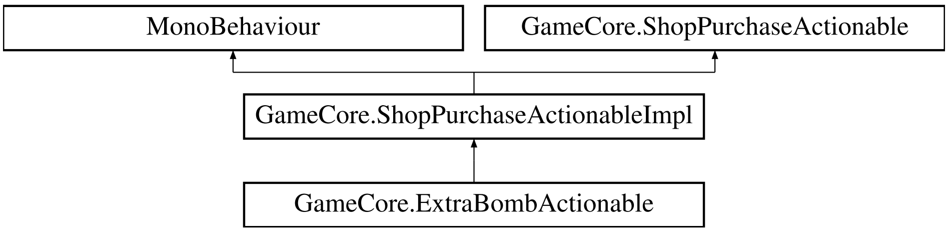
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyShootWhenRequested.cs

5.51 GameCore.ExtraBombActionable Class Reference

Provides player with an extra bomb when purchased.

Inheritance diagram for GameCore.ExtraBombActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ExtraBombActionable

Public Member Functions

- override void DoAction ()

  Increments bomb count.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

5.51.1 Detailed Description

Provides player with an extra bomb when purchased.

5.51.2 Member Function Documentation

5.51.2.1 DoAction()

```
override void GameCore.ExtraBombActionable.DoAction ( )  [virtual]
```

Increments bomb count.

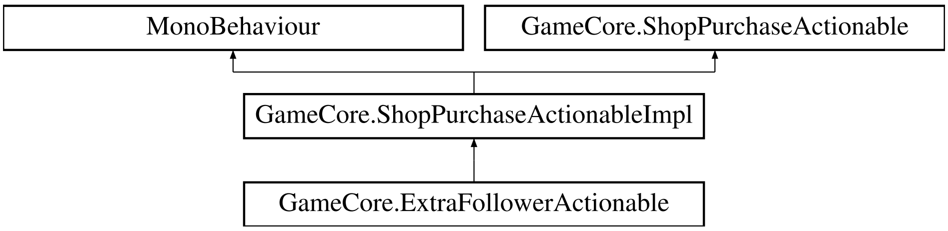
Reimplemented from GameCore.ShopPurchaseActionableImpl.

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

- Pew Pew/Scripts/Shop/ExtraBombActionable.cs

5.52 GameCore.ExtraFollowerActionable Class Reference

Inheritance diagram for GameCore.ExtraFollowerActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ExtraFollowerActionable

Additional Inherited Members

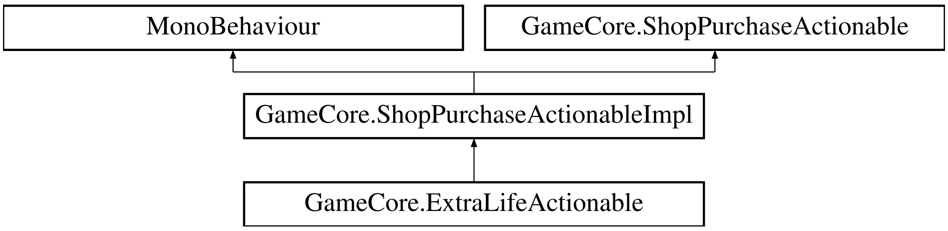
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ExtraFollowerActionable.cs

5.53 GameCore.ExtraLifeActionable Class Reference

Provides player with an extra life when purchased.

Inheritance diagram for GameCore.ExtraLifeActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ExtraLifeActionable

Public Member Functions

- override void DoAction ()

  Increments players life.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

5.53.1 Detailed Description

Provides player with an extra life when purchased.

5.53.2 Member Function Documentation

5.53.2.1 DoAction()

```
override void GameCore.ExtraLifeActionable.DoAction ( )  [virtual]
```

Increments players life.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

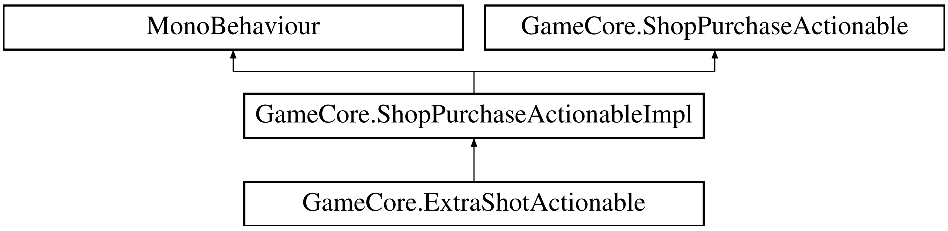
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ExtraLifeActionable.cs

5.54 GameCore.ExtraShotActionable Class Reference

Provides the player with an extra burst shot when purchased.

Inheritance diagram for GameCore.ExtraShotActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ExtraShotActionable

Public Member Functions

- override void DoAction ()

  Increments players burst shot.

Public Attributes

- PlayerShoot **playerShoot**

Additional Inherited Members

5.54.1 Detailed Description

Provides the player with an extra burst shot when purchased.

5.54.2 Member Function Documentation

5.54.2.1 DoAction()

```
override void GameCore.ExtraShotActionable.DoAction ( )  [virtual]
```

Increments players burst shot.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

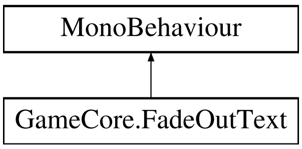
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ExtraShotActionable.cs

5.55 GameCore.FadeOutText Class Reference

Lerps texts alpha over specified number of seconds.

Inheritance diagram for GameCore.FadeOutText:



Public Attributes

- float secsToFadeOut

  The seconds to wait before beginning fade out.
- float fadeOutTime

  The time to spend fading out (seconds).

5.55.1 Detailed Description

Lerps texts alpha over specified number of seconds.

5.55.2 Member Data Documentation

5.55.2.1 fadeOutTime

```
float GameCore.FadeOutText.fadeOutTime
```

The time to spend fading out (seconds).

5.55.2.2 secsToFadeOut

```
float GameCore.FadeOutText.secsToFadeOut
```

The seconds to wait before beginning fade out.

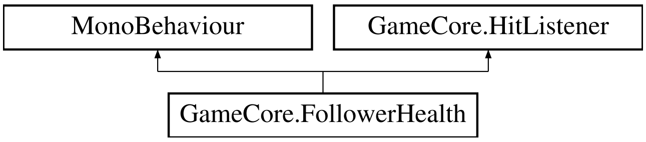
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/UI/FadeOutText.cs

5.56 GameCore.FollowerHealth Class Reference

Handles followers life, taking damage, spawning projectiles, and destroying.

Inheritance diagram for GameCore.FollowerHealth:



Public Member Functions

- void OnHit (int damage)

  Raises the hit event. Applies damage to follower. Destroys follower if current health reaches zero.

Public Attributes

- int maxHealth = 2

  The maximum starting health.
- int numOfParticlesOnDeath = 20

  The number of particles to spawn on death.
- int numOfParticlesOnDamage = 10

  The number of particles to spawn when damage is taken.
- Color particleColour

  The colour of spawned particles.
- float percentageScaleDownWhenHit = 10f

  The percentage to scale down when hit.

5.56.1 Detailed Description

Handles followers life, taking damage, spawning projectiles, and destroying.

5.56.2 Member Function Documentation

5.56.2.1 OnHit()

```
void GameCore.FollowerHealth.OnHit (
             int damage )
```

Raises the hit event. Applies damage to follower. Destroys follower if current health reaches zero.

Parameters

damage	Damage taken.

Implements GameCore.HitListener.

5.56.3 Member Data Documentation

5.56.3.1 maxHealth

```
int GameCore.FollowerHealth.maxHealth = 2
```

The maximum starting health.

5.56.3.2 numOfParticlesOnDamage

```
int GameCore.FollowerHealth.numOfParticlesOnDamage = 10
```

The number of particles to spawn when damage is taken.

5.56.3.3 numOfParticlesOnDeath

```
int GameCore.FollowerHealth.numOfParticlesOnDeath = 20
```

The number of particles to spawn on death.

5.56.3.4 particleColour

```
Color GameCore.FollowerHealth.particleColour
```

The colour of spawned particles.

5.56.3.5 percentageScaleDownWhenHit

```
float GameCore.FollowerHealth.percentageScaleDownWhenHit = 10f
```

The percentage to scale down when hit.

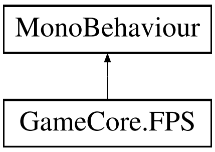
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Followers/FollowerHealth.cs

5.57 GameCore.FPS Class Reference

Dispays frames per second counter on screen (useful for measuring performance on mobile devices).

Inheritance diagram for GameCore.FPS:



5.57.1 Detailed Description

Dispays frames per second counter on screen (useful for measuring performance on mobile devices).

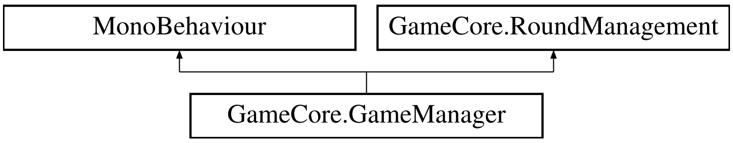
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Debug/FPS.cs

5.58 GameCore.GameManager Class Reference

Controls game flow. Starts game, initialises new rounds. Maintains list of entities within rounds to pause and resume movement.

Inheritance diagram for GameCore.GameManager:



Public Member Functions

- void OnPlayerDied ()

  Called when player dies. Finds all enemies below a certain y value (set by minimumYToKillEnemyOnPlayerDeath) and destroys them. The remaining enmies movement and shooting are paused.
- void PauseCurrentRoundEntities ()

  Pauses all current round enemies moving and shooting.
- void ResumeCurrentRoundEntities ()

  Resumes current round enemies moving and shooting.
- void OnPlayerRespawned ()

  Called when player respawns. Resumes enemy movement and shooting and updates effectors status.
- void OnPlayerDeathGameOver ()

  Called when player has died and has no lives remaining. Shows game over text and shpws game over screen.
- void OnRoundsComplete ()

Change this method to change what happens when player completes the game. Currently the game over screen is displayed.

- void OnRoundOver ()

  Opens shop when all particles have left scene.
- void OnBossRoundOver ()

  Shows boss wave complete text and opens show when particles have left scene.
- void OnChallengeRoundOver (int enemiesKilled, int enemiesInRound)

  Shows challenge wave complete percentage calculation and opens shop when particles have left scene.
- void **CloseShop** ()

Public Attributes

- Action onRoundStart

  Invoked when a round starts.
- GameObject shop

  Disabled for release. Outputs current round index to screen for debug purposes.
- GameObject player

  The parent object for the player/
- GameObject [ ] roundPrefabs

  Collection of rounds. The rounds are played sequentially.
- float minimumYToKillEnemyOnPlayerDeath = 2f

  When player dies, all enemies below this y position are killed. This prevents an enemy from killing the player as soon as they respawn.
- GameOverUIHandler gameOverHandler

  The class responsible for showing and updating the game over screen.
- AudioClip audioOnWaveComplete

  The audio clip to play on wave complete.
- PauseHandler pause

  The class responsible for showing and updating the pause screne.

Static Public Attributes

- static List< EnemyMove > EnemyMoves = new List<EnemyMove>()

  Collection of EnemyMove in current round. Used to pause and resume enemy movement.
- static List< EnemyShoot > EnemyShoots = new List<EnemyShoot>()

  Collection of EnemyShoots in current round. Used to pause and resume enemy shooting.
- static readonly float ROUND_BEGIN_TIME = 1f

  The time after the shop is closed and a new round begins.

Properties

- bool IsPlaying `[get]`

  Gets a value indicating whether this instance is playing.
- Round currentRound `[get]`

  Gets the current round.
- int currentRoundIndex `[get]`

  Gets the index of the current round.

5.58.1 Detailed Description

Controls game flow. Starts game, initialises new rounds. Maintains list of entities within rounds to pause and resume movement.

5.58.2 Member Function Documentation

5.58.2.1 OnBossRoundOver()

```
void GameCore.GameManager.OnBossRoundOver ( )
```

Shows boss wave complete text and opens show when particles have left scene.

5.58.2.2 OnChallengeRoundOver()

```
void GameCore.GameManager.OnChallengeRoundOver (
             int enemiesKilled,
             int enemiesInRound )
```

Shows challenge wave complete percentage calculation and opens shop when particles have left scene.

Parameters

enemiesKilled	Enemies killed.
enemiesInRound	Enemies in round.

Implements GameCore.RoundManagement.

5.58.2.3 OnPlayerDeathGameOver()

```
void GameCore.GameManager.OnPlayerDeathGameOver ( )
```

Called when player has died and has no lives remaining. Shows game over text and shpws game over screen.

5.58.2.4 OnPlayerDied()

```
void GameCore.GameManager.OnPlayerDied ( )
```

Called when player dies. Finds all enemies below a certain y value (set by minimumYToKillEnemyOnPlayerDeath) and destroys them. The remaining enmies movement and shooting are paused.

5.58.2.5 OnPlayerRespawned()

```
void GameCore.GameManager.OnPlayerRespawned ( )
```

Called when player respawns. Resumes enemy movement and shooting and updates effectors status.

5.58.2.6 OnRoundOver()

```
void GameCore.GameManager.OnRoundOver ( )
```

Opens shop when all particles have left scene.

Implements GameCore.RoundManagement.

5.58.2.7 OnRoundsComplete()

```
void GameCore.GameManager.OnRoundsComplete ( )
```

Change this method to change what happens when player completes the game. Currently the game over screen is displayed.

5.58.2.8 PauseCurrentRoundEntities()

```
void GameCore.GameManager.PauseCurrentRoundEntities ( )
```

Pauses all current round enemies moving and shooting.

5.58.2.9 ResumeCurrentRoundEntities()

```
void GameCore.GameManager.ResumeCurrentRoundEntities ( )
```

Resumes current round enemies moving and shooting.

5.58.3 Member Data Documentation

5.58.3.1 audioOnWaveComplete

```
AudioClip GameCore.GameManager.audioOnWaveComplete
```

The audio clip to play on wave complete.

5.58.3.2 EnemyMoves

```
List<EnemyMove> GameCore.GameManager.EnemyMoves = new List<EnemyMove>()  [static]
```

Collection of EnemyMove in current round. Used to pause and resume enemy movement.

5.58.3.3 EnemyShoots

```
List<EnemyShoot> GameCore.GameManager.EnemyShoots = new List<EnemyShoot>()  [static]
```

Collection of EnemyShoots in current round. Used to pause and resume enemy shooting.

5.58.3.4 gameOverHandler

```
GameOverUIHandler GameCore.GameManager.gameOverHandler
```

The class responsible for showing and updating the game over screen.

5.58.3.5 minimumYToKillEnemyOnPlayerDeath

```
float GameCore.GameManager.minimumYToKillEnemyOnPlayerDeath = 2f
```

When player dies, all enemies below this y position are killed. This prevents an enemy from killing the player as soon as they respawn.

5.58.3.6 onRoundStart

```
Action GameCore.GameManager.onRoundStart
```

Invoked when a round starts.

5.58.3.7 pause

```
PauseHandler GameCore.GameManager.pause
```

The class responsible for showing and updating the pause screne.

5.58.3.8 player

```
GameObject GameCore.GameManager.player
```

The parent object for the player/

5.58.3.9 ROUND_BEGIN_TIME

```
readonly float GameCore.GameManager.ROUND_BEGIN_TIME = 1f  [static]
```

The time after the shop is closed and a new round begins.

5.58.3.10 roundPrefabs

```
GameObject [ ] GameCore.GameManager.roundPrefabs
```

Collection of rounds. The rounds are played sequentially.

5.58.3.11 shop

```
GameObject GameCore.GameManager.shop
```

Disabled for release. Outputs current round index to screen for debug purposes.

The parent object for the shop UI.

5.58.4 Property Documentation

5.58.4.1 currentRound

```
Round GameCore.GameManager.currentRound  [get]
```

Gets the current round.

The current round.

5.58.4.2 currentRoundIndex

```
int GameCore.GameManager.currentRoundIndex  [get]
```

Gets the index of the current round.

The index of the current round.

5.58.4.3 IsPlaying

```
bool GameCore.GameManager.IsPlaying  [get]
```

Gets a value indicating whether this instance is playing.

`true` if this instance is playing; otherwise, `false`.

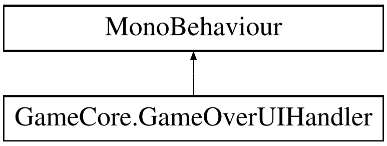
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/GameManager.cs

5.59 GameCore.GameOverUIHandler Class Reference

Shows the game over screen and handles UI requests from that screen.

Inheritance diagram for GameCore.GameOverUIHandler:



Public Member Functions

- void Show ()

  Show this instance. Updates currentRoundText and highestRoundText. Pauses entities in scene.
- void Restart ()

  Button event. Reloads game scene.
- void MainMenu ()

  Button event. Loads main menu scene.

Public Attributes

- Text currentRoundText

  The text used to show the current round number.
- Text highestRoundText

  The text used to show the highest round.
- GameObject [ ] objectsToHide

  Objects to hide before showing the game over screen.

5.59.1 Detailed Description

Shows the game over screen and handles UI requests from that screen.

5.59.2 Member Function Documentation

5.59.2.1 MainMenu()

```
void GameCore.GameOverUIHandler.MainMenu ( )
```

Button event. Loads main menu scene.

5.59.2.2 Restart()

```
void GameCore.GameOverUIHandler.Restart ( )
```

Button event. Reloads game scene.

5.59.2.3 Show()

```
void GameCore.GameOverUIHandler.Show ( )
```

Show this instance. Updates currentRoundText and highestRoundText. Pauses entities in scene.

5.59.3 Member Data Documentation

5.59.3.1 currentRoundText

```
Text GameCore.GameOverUIHandler.currentRoundText
```

The text used to show the current round number.

5.59.3.2 highestRoundText

```
Text GameCore.GameOverUIHandler.highestRoundText
```

The text used to show the highest round.

5.59.3.3 objectsToHide

```
GameObject [] GameCore.GameOverUIHandler.objectsToHide
```

Objects to hide before showing the game over screen.

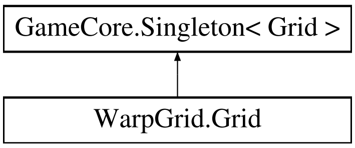
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/UI/GameOverUIHandler.cs

5.60 WarpGrid.Grid Class Reference

Updates and displays warping grid. This is a conversion of an XNA project found here: https://gamedevelopment.tutsplus.com/tutorials/make-a-neon-vector-shooter-in-xna-the-warping-

Inheritance diagram for WarpGrid.Grid:



Public Types

- enum DrawMethod { **Quick**, **Smooth** }

  Smooth method creates smoother curves but is more expensive. The DrawMethod::Quick is recommended for mobile devices.

Public Member Functions

- void CreateGrid ()

  Creates the grid. Grid is built offscreen and then moved onto the screen to prevent instantiated unpositioned lines from showing onscreen.
- void DisableGrid ()

  Destroys grid if created.
- void ApplyDirectedForce (Vector2 force, Vector2 position, float radius)

  Applies a directed force at position.
- void ApplyImplosiveForce (float force, Vector2 position, float radius)

  Applies an implosive force at position.
- void ApplyExplosiveForce (float force, Vector2 position, float radius)

  Applies an explosive force at position.

Public Attributes

- Rect size

  The grid size. The grid is computationally expensive so make sure you create the smallest grid possible.
- Vector2 spacing

  The spacing between each line.
- float minLineWidth = 0.01f

  The minimum width of the line. This width is used in the inner squares.
- float maxLineWidth = 0.03f

  The width of the max line. These lines are used in the outter squares.
- int maxInstantiatedLines = 1520

  Limits the maximum number of lines in the grid. If this number is too low, the grid will not be drawn in its entirety.
- GameObject linePrefab

  The prefab used for the line.
- Color gridColour

  The colour of the instantiated lines.
- DrawMethod drawMethod = DrawMethod.Quick

  The selected draw method.

Additional Inherited Members

5.60.1 Detailed Description

Updates and displays warping grid. This is a conversion of an XNA project found here: `https://gamedevelopment.tutsplus.com/tutorials/make-a-neon-vector-shooter-in-xna-the-warping-`

5.60.2 Member Enumeration Documentation

5.60.2.1 DrawMethod

```
enum WarpGrid.Grid.DrawMethod [strong]
```

Smooth method creates smoother curves but is more expensive. The DrawMethod::Quick is recommended for mobile devices.

5.60.3 Member Function Documentation

5.60.3.1 ApplyDirectedForce()

```
void WarpGrid.Grid.ApplyDirectedForce (
             Vector2 force,
             Vector2 position,
             float radius )
```

Applies a directed force at position.

Parameters

force	Force.
position	Position.
radius	Radius.

5.60.3.2 ApplyExplosiveForce()

```
void WarpGrid.Grid.ApplyExplosiveForce (
             float force,
             Vector2 position,
             float radius )
```

Applies an explosive force at position.

Parameters

force	Force.
position	Position.
radius	Radius.

5.60.3.3 ApplyImplosiveForce()

```
void WarpGrid.Grid.ApplyImplosiveForce (
             float force,
             Vector2 position,
             float radius )
```

Applies an implosive force at position.

Parameters

force	Force.
position	Position.
radius	Radius.

5.60.3.4 CreateGrid()

```
void WarpGrid.Grid.CreateGrid ( )
```

Creates the grid. Grid is built offscreen and then moved onto the screen to prevent instantiated unpositioned lines from showing onscreen.

5.60.3.5 DisableGrid()

```
void WarpGrid.Grid.DisableGrid ( )
```

Destroys grid if created.

5.60.4 Member Data Documentation

5.60.4.1 drawMethod

```
DrawMethod WarpGrid.Grid.drawMethod = DrawMethod.Quick
```

The selected draw method.

5.60.4.2 gridColour

```
Color WarpGrid.Grid.gridColour
```

The colour of the instantiated lines.

5.60.4.3 linePrefab

```
GameObject WarpGrid.Grid.linePrefab
```

The prefab used for the line.

5.60.4.4 maxInstantiatedLines

```
int WarpGrid.Grid.maxInstantiatedLines = 1520
```

Limits the maximum number of lines in the grid. If this number is too low, the grid will not be drawn in its entirety.

5.60.4.5 maxLineWidth

```
float WarpGrid.Grid.maxLineWidth = 0.03f
```

The width of the max line. These lines are used in the outter squares.

5.60.4.6 minLineWidth

```
float WarpGrid.Grid.minLineWidth = 0.01f
```

The minimum width of the line. This width is used in the inner squares.

5.60.4.7 size

```
Rect WarpGrid.Grid.size
```

The grid size. The grid is computationally expensive so make sure you create the smallest grid possible.

5.60.4.8 spacing

```
Vector2 WarpGrid.Grid.spacing
```

The spacing between each line.

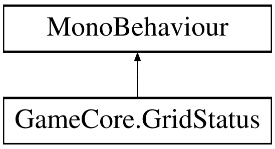
The documentation for this class was generated from the following file:

- UG/Scripts/Grid.cs

5.61 GameCore.GridStatus Class Reference

Stores persistent status of grid. Data is stored in PlayerPrefs. When a user disables/enables the grid, it is stored and loaded next time they play. As object is persistent, the grid status is carried from main menu scene to game scene.

Inheritance diagram for GameCore.GridStatus:



Public Member Functions

- void SetGridEnabled (bool ignored)

  Sets the grid status based on toggle status.

Public Attributes

- Toggle toggle

  The toggle used to enable/disable grid.

5.61.1 Detailed Description

Stores persistent status of grid. Data is stored in PlayerPrefs. When a user disables/enables the grid, it is stored and loaded next time they play. As object is persistent, the grid status is carried from main menu scene to game scene.

5.61.2 Member Function Documentation

5.61.2.1 SetGridEnabled()

```
void GameCore.GridStatus.SetGridEnabled (
            bool ignored )
```

Sets the grid status based on toggle status.

Parameters

ignored	Included to link to Unity toggle. Not used. Toggle status is queried directly from toggle.

5.61.3 Member Data Documentation

5.61.3.1 toggle

`Toggle GameCore.GridStatus.toggle`

The toggle used to enable/disable grid.

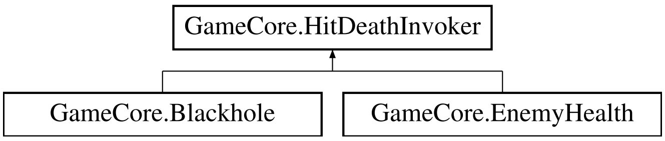
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Grid/GridStatus.cs

5.62 GameCore.HitDeathInvoker Interface Reference

Provides contract to providing a hook for an entities onDeath and onHit events.

Inheritance diagram for GameCore.HitDeathInvoker:



Properties

- Action onDeath `[get, set]`

    Gets or sets an action to perform on death.
- Action onHit `[get, set]`

    Gets or sets an action to perform on hit.

5.62.1 Detailed Description

Provides contract to providing a hook for an entities onDeath and onHit events.

5.62.2 Property Documentation

5.62.2.1 onDeath

```
Action GameCore.HitDeathInvoker.onDeath  [get], [set]
```

Gets or sets an action to perform on death.

The on death action.

5.62.2.2 onHit

```
Action GameCore.HitDeathInvoker.onHit  [get], [set]
```

Gets or sets an action to perform on hit.

The on hit action.

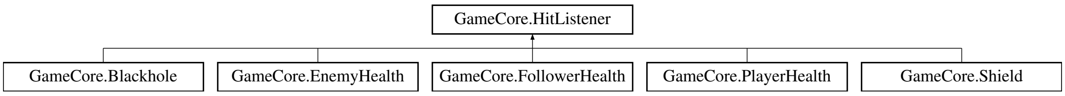
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyHealth.cs

5.63 GameCore.HitListener Interface Reference

Contract for any entity that can take damage or react to damage.

Inheritance diagram for GameCore.HitListener:



Public Member Functions

- void OnHit (int damage)

  Raises the hit event.

5.63.1 Detailed Description

Contract for any entity that can take damage or react to damage.

5.63.2 Member Function Documentation

5.63.2.1 OnHit()

```
void GameCore.HitListener.OnHit (
             int damage )
```

Raises the hit event.

Parameters

damage	Damage taken.

Implemented in GameCore.EnemyHealth, GameCore.Blackhole, GameCore.PlayerHealth, GameCore.FollowerHealth, and GameCore.Shield.

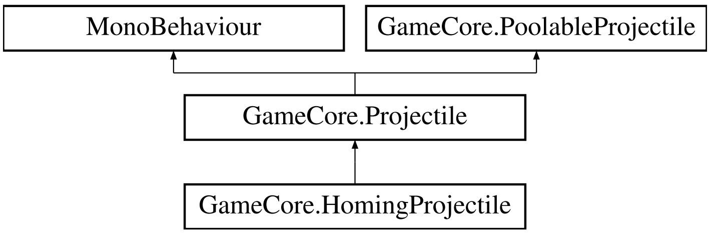
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyHealth.cs

5.64 GameCore.HomingProjectile Class Reference

Enables a projectile to change heading based on players current location.

Inheritance diagram for GameCore.HomingProjectile:



Public Attributes

- float turnSpeed = 20f

  The speed at which the projectile can turn.
- float delayToTurn = 0.8f

  The delay before projectile starts homing towards players location.
- int numOfParticlesToSpawnWhenTimeUp = 10

  The number of particles to spawn when projectile is removed from game.

Protected Member Functions

- override void **Awake** ()
- override void **Update** ()

Additional Inherited Members

5.64.1 Detailed Description

Enables a projectile to change heading based on players current location.

5.64.2 Member Data Documentation

5.64.2.1 delayToTurn

```
float GameCore.HomingProjectile.delayToTurn = 0.8f
```

The delay before projectile starts homing towards players location.

5.64.2.2 numOfParticlesToSpawnWhenTimeUp

```
int GameCore.HomingProjectile.numOfParticlesToSpawnWhenTimeUp = 10
```

The number of particles to spawn when projectile is removed from game.

5.64.2.3 turnSpeed

```
float GameCore.HomingProjectile.turnSpeed = 20f
```

The speed at which the projectile can turn.

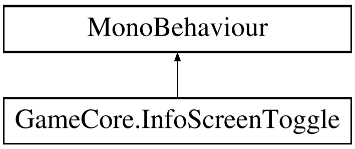
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Projectiles/HomingProjectile.cs

5.65 GameCore.InfoScreenToggle Class Reference

Shows info screen and handles ui requests from that screen.

Inheritance diagram for GameCore.InfoScreenToggle:



Public Member Functions

- void Toggle ()

  If info screen is open, it is closed and vise versa.

Public Attributes

- Sprite openInfoImage

  The button sprite to use when the info screen is closed.
- Sprite closeInfoImage

  The button sprite to use when the info screen is open.
- GameObject infoScreen

  The parent object of the info screen.
- GameObject [ ] objectsToHide

  The objects to hide before showing the info screen.

5.65.1 Detailed Description

Shows info screen and handles ui requests from that screen.

5.65.2 Member Function Documentation

5.65.2.1 Toggle()

```
void GameCore.InfoScreenToggle.Toggle ( )
```

If info screen is open, it is closed and vise versa.

5.65.3 Member Data Documentation

5.65.3.1 closeInfoImage

```
Sprite GameCore.InfoScreenToggle.closeInfoImage
```

The button sprite to use when the info screen is open.

5.65.3.2 infoScreen

```
GameObject GameCore.InfoScreenToggle.infoScreen
```

The parent object of the info screen.

5.65.3.3 objectsToHide

```
GameObject [] GameCore.InfoScreenToggle.objectsToHide
```

The objects to hide before showing the info screen.

5.65.3.4 openInfoImage

```
Sprite GameCore.InfoScreenToggle.openInfoImage
```

The button sprite to use when the info screen is closed.

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

- Pew Pew/Scripts/UI/InfoScreenToggle.cs

5.66 WarpGrid.Interpolate Class Reference

Public Types

- enum EaseType {
Linear, **EaseInQuad**, **EaseOutQuad**, **EaseInOutQuad**,
EaseInCubic, **EaseOutCubic**, **EaseInOutCubic**, **EaseInQuart**,
EaseOutQuart, **EaseInOutQuart**, **EaseInQuint**, **EaseOutQuint**,
EaseInOutQuint, **EaseInSine**, **EaseOutSine**, **EaseInOutSine**,
EaseInExpo, **EaseOutExpo**, **EaseInOutExpo**, **EaseInCirc**,
EaseOutCirc, **EaseInOutCirc** }

Public Member Functions

- delegate Vector3 **ToVector3**< **T** > (T v)
- delegate float **Function** (float a, float b, float c, float d)

Static Public Member Functions

- static IEnumerator NewEase (Function ease, Vector3 start, Vector3 end, float duration)
- static IEnumerator NewEase (Function ease, Vector3 start, Vector3 end, int slices)
- static Function Ease (EaseType type)
- static IEnumerable< Vector3 > NewBezier (Function ease, Transform[ ] nodes, float duration)
- static IEnumerable< Vector3 > NewBezier (Function ease, Transform[ ] nodes, int slices)
- static IEnumerable< Vector3 > NewBezier (Function ease, Vector3[ ] points, float duration)
- static IEnumerable< Vector3 > NewBezier (Function ease, Vector3[ ] points, int slices)
- static IEnumerable< Vector3 > NewCatmullRom (Transform[ ] nodes, int slices, bool loop)
- static IEnumerable< Vector3 > NewCatmullRom (Vector3[ ] points, int slices, bool loop)
- static Vector3 CatmullRom (Vector3 previous, Vector3 start, Vector3 end, Vector3 next, float elapsedTime, float duration)

5.66.1 Member Enumeration Documentation

5.66.1.1 EaseType

```
enum WarpGrid.Interpolate.EaseType  [strong]
```

Different methods of easing interpolation.

5.66.2 Member Function Documentation

5.66.2.1 CatmullRom()

```
static Vector3 WarpGrid.Interpolate.CatmullRom (
             Vector3 previous,
             Vector3 start,
             Vector3 end,
             Vector3 next,
             float elapsedTime,
             float duration )  [static]
```

A Vector3 Catmull-Rom spline. Catmull-Rom splines are similar to bezier splines but have the useful property that the generated curve will go through each of the control points.

NOTE: The NewCatmullRom() functions are an easier to use alternative to this raw Catmull-Rom implementation.

Parameters

previous	the point just before the start point or the start point itself if no previous point is available
start	generated when elapsedTime == 0
end	generated when elapsedTime >= duration
next	the point just after the end point or the end point itself if no next point is available

5.66.2.2 Ease()

```
static Function WarpGrid.Interpolate.Ease (
             EaseType type )  [static]
```

Returns the static method that implements the given easing type for scalars. Use this method to easily switch between easing interpolation types.

All easing methods clamp elapsedTime so that it is always <= duration.

var ease = Interpolate.Ease(EaseType.EaseInQuad); i = ease(start, distance, elapsedTime, duration);

5.66.2.3 NewBezier() [1/4]

```
static IEnumerable<Vector3> WarpGrid.Interpolate.NewBezier (
            Function ease,
            Transform [] nodes,
            float duration )  [static]
```

Returns sequence generator from the first node to the last node over duration time using the points in-between the first and last node as control points of a bezier curve used to generate the interpolated points in the sequence. If there are no control points (ie. only two nodes, first and last) then this behaves exactly the same as NewEase(). In other words a zero-degree bezier spline curve is just the easing method. The sequence is generated as it is accessed using the Time.deltaTime to calculate the portion of duration that has elapsed.

5.66.2.4 NewBezier() [2/4]

```
static IEnumerable<Vector3> WarpGrid.Interpolate.NewBezier (
            Function ease,
            Transform [] nodes,
            int slices )  [static]
```

Instead of interpolating based on time, generate n interpolated points (slices) between the first and last node.

5.66.2.5 NewBezier() [3/4]

```
static IEnumerable<Vector3> WarpGrid.Interpolate.NewBezier (
            Function ease,
            Vector3 [] points,
            float duration )  [static]
```

A Vector3[] variation of the Transform[] NewBezier() function. Same functionality but using Vector3s to define bezier curve.

5.66.2.6 NewBezier() [4/4]

```
static IEnumerable<Vector3> WarpGrid.Interpolate.NewBezier (
            Function ease,
            Vector3 [] points,
            int slices )  [static]
```

A Vector3[] variation of the Transform[] NewBezier() function. Same functionality but using Vector3s to define bezier curve.

5.66.2.7 NewCatmullRom() [1/2]

```
static IEnumerable<Vector3> WarpGrid.Interpolate.NewCatmullRom (
            Transform [] nodes,
            int slices,
            bool loop )  [static]
```

Returns sequence generator from the first node, through each control point, and to the last node. N points are generated between each node (slices) using Catmull-Rom.

5.66.2.8 NewCatmullRom() [2/2]

```
static IEnumerable<Vector3> WarpGrid.Interpolate.NewCatmullRom (
            Vector3 [ ] points,
            int slices,
            bool loop )  [static]
```

A Vector3[] variation of the Transform[] NewCatmullRom() function. Same functionality but using Vector3s to define curve.

5.66.2.9 NewEase() [1/2]

```
static IEnumerator WarpGrid.Interpolate.NewEase (
            Function ease,
            Vector3 start,
            Vector3 end,
            float duration )  [static]
```

Returns sequence generator from start to end over duration using the given easing function. The sequence is generated as it is accessed using the Time.deltaTime to calculate the portion of duration that has elapsed.

5.66.2.10 NewEase() [2/2]

```
static IEnumerator WarpGrid.Interpolate.NewEase (
            Function ease,
            Vector3 start,
            Vector3 end,
            int slices )  [static]
```

Instead of easing based on time, generate n interpolated points (slices) between the start and end positions.

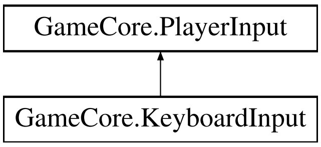
The documentation for this class was generated from the following file:

- UG/Scripts/Interpolate.cs

5.67 GameCore.KeyboardInput Class Reference

Provides method to control player based on keyboard input.

Inheritance diagram for GameCore.KeyboardInput:



Public Member Functions

- KeyboardInput (float moveSpeed)

    Initializes a new instance of the KeyboardInput class.
- Vector2 GetVelocity ()

    Gets the velocity. The players next move.
- float GetMovementSpeed ()

    Gets the players movement speed.
- void SetMovementSpeed (float amount)

    Sets the players movement speed.

5.67.1 Detailed Description

Provides method to control player based on keyboard input.

5.67.2 Constructor & Destructor Documentation

5.67.2.1 KeyboardInput()

```
GameCore.KeyboardInput.KeyboardInput (
              float moveSpeed )
```

Initializes a new instance of the KeyboardInput class.

Parameters

moveSpeed	Move speed.

5.67.3 Member Function Documentation

5.67.3.1 GetMovementSpeed()

```
float GameCore.KeyboardInput.GetMovementSpeed ( )
```

Gets the players movement speed.

Returns

The movement speed.

Implements GameCore.PlayerInput.

5.67.3.2 GetVelocity()

```
Vector2 GameCore.KeyboardInput.GetVelocity ( )
```

Gets the velocity. The players next move.

Returns

The velocity.

Implements GameCore.PlayerInput.

5.67.3.3 SetMovementSpeed()

```
void GameCore.KeyboardInput.SetMovementSpeed (
             float amount )
```

Sets the players movement speed.

Parameters

amount	Move speed.

Implements GameCore.PlayerInput.

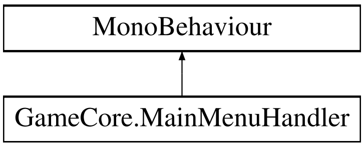
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/Movement/PlayerInput.cs

5.68 GameCore.MainMenuHandler Class Reference

Shows main menu screen and hanles UI requests from scene.

Inheritance diagram for GameCore.MainMenuHandler:



Public Member Functions

- void Play ()

  Button event. Loads the game scene.

Public Attributes

- Text highscoreText

  The text used to diaplay the current highest round.
- float touchGridRadius = 2f

  The radius of the touch effect on the grid.
- float touchGridForce = 4f

  The force if the touch effect on the grid.
- AudioClip audioOnGridTouch

  The audio to play on grid touch.

5.68.1 Detailed Description

Shows main menu screen and hanles UI requests from scene.

5.68.2 Member Function Documentation

5.68.2.1 Play()

```
void GameCore.MainMenuHandler.Play ( )
```

Button event. Loads the game scene.

5.68.3 Member Data Documentation

5.68.3.1 audioOnGridTouch

```
AudioClip GameCore.MainMenuHandler.audioOnGridTouch
```

The audio to play on grid touch.

5.68.3.2 highscoreText

```
Text GameCore.MainMenuHandler.highscoreText
```

The text used to diaplay the current highest round.

5.68.3.3 touchGridForce

```
float GameCore.MainMenuHandler.touchGridForce = 4f
```

The force if the touch effect on the grid.

5.68.3.4 touchGridRadius

```
float GameCore.MainMenuHandler.touchGridRadius = 2f
```

The radius of the touch effect on the grid.

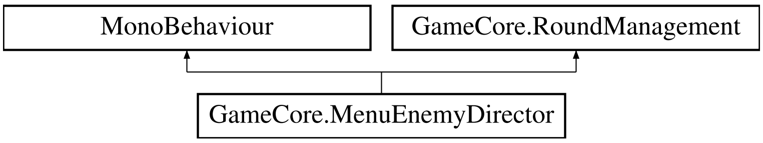
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/UI/MainMenuHandler.cs

5.69 GameCore.MenuEnemyDirector Class Reference

Directs enemies as part of the main menu scene.

Inheritance diagram for GameCore.MenuEnemyDirector:



Public Member Functions

- void OnChallengeRoundOver (int enemiesKilled, int maxEnemies)

  Included to meet the contract outlined in RoundManagement. However it is not used during the menu scene.
- void OnRoundOver ()

  Starts a new round.

Public Attributes

- GameObject [ ] menuRounds

  Possible enemies that can be spawned during the main menu scene. One is selected at random.

5.69.1 Detailed Description

Directs enemies as part of the main menu scene.

5.69.2 Member Function Documentation

5.69.2.1 OnChallengeRoundOver()

```
void GameCore.MenuEnemyDirector.OnChallengeRoundOver (
             int enemiesKilled,
             int maxEnemies )
```

Included to meet the contract outlined in RoundManagement. However it is not used during the menu scene.

Parameters

enemiesKilled	Ignored.
maxEnemies	Ignored.

Implements GameCore.RoundManagement.

5.69.2.2 OnRoundOver()

```
void GameCore.MenuEnemyDirector.OnRoundOver ( )
```

Starts a new round.

Implements GameCore.RoundManagement.

5.69.3 Member Data Documentation

5.69.3.1 menuRounds

```
GameObject [ ] GameCore.MenuEnemyDirector.menuRounds
```

Possible enemies that can be spawned during the main menu scene. One is selected at random.

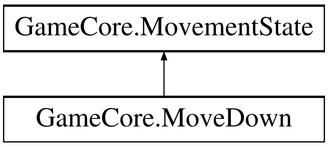
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/MenuEnemyDirector.cs

5.70 GameCore.MoveDown Class Reference

Move down movement state.

Inheritance diagram for GameCore.MoveDown:



Public Member Functions

- MoveDown (Vector2 targetOffset)

  Initializes a new instance of the MoveDown class.
- void Enter (Transform owner)

  Enter the specified state.
- Vector2 NextMove ()

  Returns the next movement.
- bool CompletedMove (Transform owner)

  Returns true if object has completed the movement state.

5.70.1 Detailed Description

Move down movement state.

5.70.2 Constructor & Destructor Documentation

5.70.2.1 MoveDown()

```
GameCore.MoveDown.MoveDown (
              Vector2 targetOffset )
```

Initializes a new instance of the MoveDown class.

Parameters

targetOffset	Target offset.

5.70.3 Member Function Documentation

5.70.3.1 CompletedMove()

```
bool GameCore.MoveDown.CompletedMove (
              Transform owner )
```

Returns true if object has completed the movement state.

Returns

true

```
false
```

Parameters

owner	Owner.

Implements GameCore.MovementState.

5.70.3.2 Enter()

```
void GameCore.MoveDown.Enter (
             Transform owner )
```

Enter the specified state.

Parameters

owner	Owner of state.

Implements GameCore.MovementState.

5.70.3.3 NextMove()

```
Vector2 GameCore.MoveDown.NextMove ( )
```

Returns the next movement.

Returns

The move to perform.

Implements GameCore.MovementState.

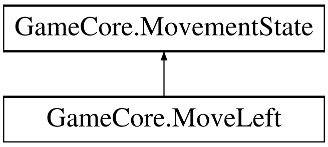
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/MovementState.cs

5.71 GameCore.MoveLeft Class Reference

Move left movement state.

Inheritance diagram for GameCore.MoveLeft:



Public Member Functions

- MoveLeft (ScreenBounds bounds)

    Initializes a new instance of the MoveLeft class.
- void Enter (Transform owner)

    Enter the specified state.
- Vector2 NextMove ()

    Returns the next movement.
- bool CompletedMove (Transform owner)

    Returns true if object has completed the movement state.

5.71.1 Detailed Description

Move left movement state.

5.71.2 Constructor & Destructor Documentation

5.71.2.1 MoveLeft()

```
GameCore.MoveLeft.MoveLeft (
             ScreenBounds bounds )
```

Initializes a new instance of the MoveLeft class.

Parameters

bounds	Bounds.

5.71.3 Member Function Documentation

5.71.3.1 CompletedMove()

```
bool GameCore.MoveLeft.CompletedMove (
             Transform owner )
```

Returns true if object has completed the movement state.

Returns

true

```
false
```

Parameters

owner	Owner.

Implements GameCore.MovementState.

5.71.3.2 Enter()

```
void GameCore.MoveLeft.Enter (
             Transform owner )
```

Enter the specified state.

Parameters

owner	Owner of state.

Implements GameCore.MovementState.

5.71.3.3 NextMove()

```
Vector2 GameCore.MoveLeft.NextMove ( )
```

Returns the next movement.

Returns

The move to perform.

Implements GameCore.MovementState.

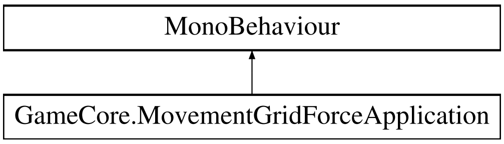
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/MovementState.cs

5.72 GameCore.MovementGridForceApplication Class Reference

Applies a directional force to the background grid based on owners direction and velocity.

Inheritance diagram for GameCore.MovementGridForceApplication:



Public Attributes

- float radius = 1f

  The force radius.
- float forceMultiplier = 1f

  The multiplier to apply to velocity.

5.72.1 Detailed Description

Applies a directional force to the background grid based on owners direction and velocity.

5.72.2 Member Data Documentation

5.72.2.1 forceMultiplier

```
float GameCore.MovementGridForceApplication.forceMultiplier = 1f
```

The multiplier to apply to velocity.

5.72.2.2 radius

```
float GameCore.MovementGridForceApplication.radius = 1f
```

The force radius.

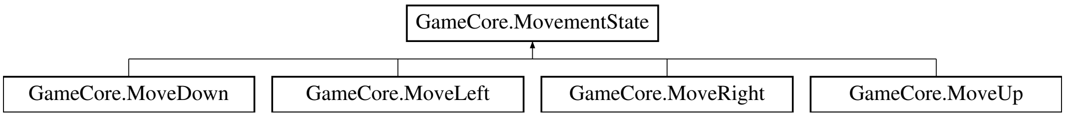
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/MovementGridForceApplication.cs

5.73 GameCore.MovementState Interface Reference

Contract for a directional movement state.

Inheritance diagram for GameCore.MovementState:



Public Member Functions

- void Enter (Transform owner)

  Enter the specified state.
- Vector2 NextMove ()

  Returns the next movement.
- bool CompletedMove (Transform owner)

  Returns true if object has completed the movement state.

5.73.1 Detailed Description

Contract for a directional movement state.

5.73.2 Member Function Documentation

5.73.2.1 CompletedMove()

```
bool GameCore.MovementState.CompletedMove (
            Transform owner )
```

Returns true if object has completed the movement state.

Returns

`true`, if move was completeded, `false` otherwise.

Parameters

owner	Owner.

Implemented in GameCore.MoveDown, GameCore.MoveUp, GameCore.MoveLeft, and GameCore.MoveRight.

5.73.2.2 Enter()

```
void GameCore.MovementState.Enter (
            Transform owner )
```

Enter the specified state.

Parameters

owner	Owner of state.

Implemented in GameCore.MoveDown, GameCore.MoveUp, GameCore.MoveLeft, and GameCore.MoveRight.

5.73.2.3 NextMove()

```
Vector2 GameCore.MovementState.NextMove ( )
```

Returns the next movement.

Returns

The move to perform.

Implemented in GameCore.MoveDown, GameCore.MoveUp, GameCore.MoveLeft, and GameCore.MoveRight.

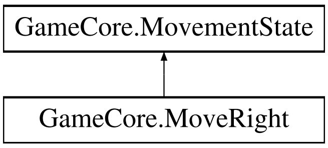
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/MovementState.cs

5.74 GameCore.MoveRight Class Reference

Move right movement state.

Inheritance diagram for GameCore.MoveRight:



Public Member Functions

- MoveRight (ScreenBounds bounds)

  Initializes a new instance of the MoveRight class.
- void Enter (Transform owner)

  Enter the specified state.
- Vector2 NextMove ()

  Returns the next movement.
- bool CompletedMove (Transform owner)

  Returns true if object has completed the movement state.

5.74.1 Detailed Description

Move right movement state.

5.74.2 Constructor & Destructor Documentation

5.74.2.1 MoveRight()

```
GameCore.MoveRight.MoveRight (
            ScreenBounds bounds )
```

Initializes a new instance of the MoveRight class.

Parameters

bounds	Bounds.

5.74.3 Member Function Documentation

5.74.3.1 CompletedMove()

```
bool GameCore.MoveRight.CompletedMove (
             Transform owner )
```

Returns true if object has completed the movement state.

Returns

true

`false`

Parameters

owner	Owner.

Implements GameCore.MovementState.

5.74.3.2 Enter()

```
void GameCore.MoveRight.Enter (
             Transform owner )
```

Enter the specified state.

Parameters

owner	Owner of state.

Implements GameCore.MovementState.

5.74.3.3 NextMove()

```
Vector2 GameCore.MoveRight.NextMove ( )
```

Returns the next movement.

Returns

The move to perform.

Implements GameCore.MovementState.

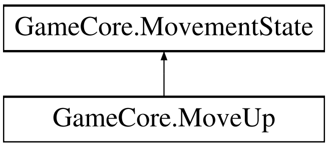
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/MovementState.cs

5.75 GameCore.MoveUp Class Reference

Move up movement state.

Inheritance diagram for GameCore.MoveUp:



Public Member Functions

- void Enter (Transform owner)

  Enter the specified state.
- Vector2 NextMove ()

  Returns the next movement.
- bool CompletedMove (Transform owner)

  Returns true if object has completed the movement state.

5.75.1 Detailed Description

Move up movement state.

5.75.2 Member Function Documentation

5.75.2.1 CompletedMove()

```
bool GameCore.MoveUp.CompletedMove (
            Transform owner )
```

Returns true if object has completed the movement state.

Returns

true

`false`

Parameters

owner	Owner.

Implements GameCore.MovementState.

5.75.2.2 Enter()

```
void GameCore.MoveUp.Enter (
             Transform owner )
```

Enter the specified state.

Parameters

owner	Owner of state.

Implements GameCore.MovementState.

5.75.2.3 NextMove()

```
Vector2 GameCore.MoveUp.NextMove ( )
```

Returns the next movement.

Returns

The move to perform.

Implements GameCore.MovementState.

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

- Pew Pew/Scripts/Enemies/Movement/MovementState.cs

5.76 GameCore.ObjectPool< T > Class Template Reference

Generic object pool.

Public Member Functions

- ObjectPool (GameObject prefab, int numToPool)

  Initializes a new instance of the ObjectPool'1 class.
- ObjectPool (GameObject prefab, int numToPool, Transform owner)

  Initializes a new instance of the ObjectPool'1 class.
- T GetObject ()

  Returns object from pool. If no object is found, null is returned.
- void PoolObject (T obj)

  Disables object and adds to pool.
- List< T > GetActive ()

  Returns list of active objects. These objects have are currently in use.

5.76.1 Detailed Description

Generic object pool.

Type Constraints

T* : *MonoBehaviour

5.76.2 Constructor & Destructor Documentation

5.76.2.1 ObjectPool() [1/2]

```
GameCore.ObjectPool< T >.ObjectPool (
              GameObject prefab,
              int numToPool )
```

Initializes a new instance of the ObjectPool'1 class.

Parameters

prefab	Prefab.
numToPool	Number to pool.

5.76.2.2 ObjectPool() [2/2]

```
GameCore.ObjectPool< T >.ObjectPool (
              GameObject prefab,
              int numToPool,
              Transform owner )
```

Initializes a new instance of the ObjectPool'1 class.

Parameters

prefab	Prefab.
numToPool	Number to pool.
owner	Owner.

5.76.3 Member Function Documentation

5.76.3.1 GetActive()

```
List<T> GameCore.ObjectPool< T >.GetActive ( )
```

Returns list of active objects. These objects have are currently in use.

Returns

The active.

5.76.3.2 GetObject()

```
T GameCore.ObjectPool< T >.GetObject ( )
```

Returns object from pool. If no object is found, null is returned.

Returns

The object.

5.76.3.3 PoolObject()

```
void GameCore.ObjectPool< T >.PoolObject (
            T obj )
```

Disables object and adds to pool.

Parameters

obj	Object to pool.

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

- Pew Pew/Scripts/Utilities/ObjectPool.cs

5.77 PE2D.ParticleBuilder Struct Reference

Holds the particle state. Passed to the ParticleFactory to build particles.

Public Attributes

- Vector2 velocity

  Initial velocity of particle.
- WrapAroundType wrapAroundType

  Screen constraint type.
- float lengthMultiplier

  The particles scale is multipled by this.
- float velocityDampModifier

  The percentage amount that a particles velocity remains each timestep.
- bool ignoreEffectors

  If enables, the particle built with this state will ignore effectors.
- bool **canBeCollectedByPlayer**
- float minLengthClamp

  Clamp the minimum length of a particles sprite.
- float maxLengthClamp

  Clamp the maximum length of a particles sprite.
- bool removeWhenVelocityReachesThreshold

  Will remove a particle if velocity reaches a threshold.
- float customVelocityThreshold

  The velocity at which a particle will be removed, only used if removeWhenVelocityReachesThreshold = true.
- bool removeWhenAlphaReachesThreshold

  Will remove the particle when its alpha reaches a specified threshold.
- float customAlphaThreshold

  The particles sprites alpha threshold at which a particle will be removed, only used if removeWhenAlphaReaches←Threshold = true.
- bool **ignoreInitialEffectorTime**

5.77.1 Detailed Description

Holds the particle state. Passed to the ParticleFactory to build particles.

5.77.2 Member Data Documentation

5.77.2.1 customAlphaThreshold

```
float PE2D.ParticleBuilder.customAlphaThreshold
```

The particles sprites alpha threshold at which a particle will be removed, only used if removeWhenAlphaReachesThreshold = true.

5.77.2.2 customVelocityThreshold

```
float PE2D.ParticleBuilder.customVelocityThreshold
```

The velocity at which a particle will be removed, only used if removeWhenVelocityReachesThreshold = true.

5.77.2.3 ignoreEffectors

```
bool PE2D.ParticleBuilder.ignoreEffectors
```

If enables, the particle built with this state will ignore effectors.

5.77.2.4 lengthMultiplier

```
float PE2D.ParticleBuilder.lengthMultiplier
```

The particles scale is multipled by this.

5.77.2.5 maxLengthClamp

```
float PE2D.ParticleBuilder.maxLengthClamp
```

Clamp the maximum length of a particles sprite.

5.77.2.6 minLengthClamp

```
float PE2D.ParticleBuilder.minLengthClamp
```

Clamp the minimum length of a particles sprite.

5.77.2.7 removeWhenAlphaReachesThreshold

```
bool PE2D.ParticleBuilder.removeWhenAlphaReachesThreshold
```

Will remove the particle when its alpha reaches a specified threshold.

5.77.2.8 removeWhenVelocityReachesThreshold

```
bool PE2D.ParticleBuilder.removeWhenVelocityReachesThreshold
```

Will remove a particle if velocity reaches a threshold.

5.77.2.9 velocity

```
Vector2 PE2D.ParticleBuilder.velocity
```

Initial velocity of particle.

5.77.2.10 velocityDampModifier

```
float PE2D.ParticleBuilder.velocityDampModifier
```

The percentage amount that a particles velocity remains each timestep.

5.77.2.11 wrapAroundType

```
WrapAroundType PE2D.ParticleBuilder.wrapAroundType
```

Screen constraint type.

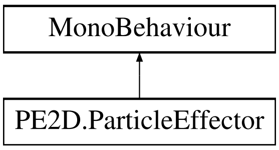
The documentation for this struct was generated from the following file:

- pe2d/Particles/ParticleBuilder.cs

5.78 PE2D.ParticleEffector Class Reference

Add to a gameobject to effect a particles movement.

Inheritance diagram for PE2D.ParticleEffector:



Public Attributes

- EffectorType **effectorType**
- float **distance**
- float **rotateDistance**
- float **force**
- bool **effectProjectiles** = true

5.78.1 Detailed Description

Add to a gameobject to effect a particles movement.

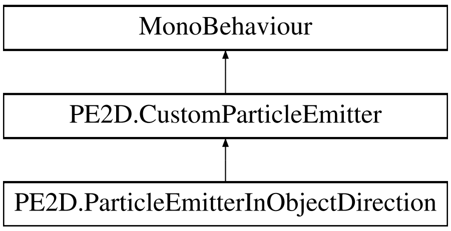
The documentation for this class was generated from the following file:

- pe2d/Particles/ParticleEffector.cs

5.79 PE2D.ParticleEmitterInObjectDirection Class Reference

Emits particles based on objects rotation.

Inheritance diagram for PE2D.ParticleEmitterInObjectDirection:


MonoBehaviour
PE2D.CustomParticleEmitter
PE2D.ParticleEmitterInObjectDirection

Protected Member Functions

- override void **ReleaseParticle** ()

Additional Inherited Members

5.79.1 Detailed Description

Emits particles based on objects rotation.

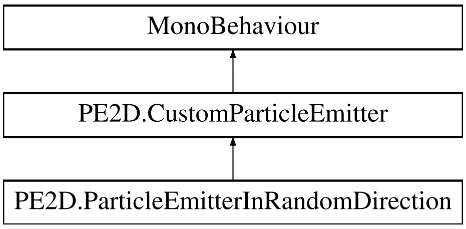
The documentation for this class was generated from the following file:

- pe2d/Particles/Emitters/ParticleEmitterInObjectDirection.cs

5.80 PE2D.ParticleEmitterInRandomDirection Class Reference

Emits particles from objects position in a random direction.

Inheritance diagram for PE2D.ParticleEmitterInRandomDirection:


MonoBehaviour
PE2D.CustomParticleEmitter
PE2D.ParticleEmitterInRandomDirection

Protected Member Functions

- override void **ReleaseParticle** ()

Additional Inherited Members

5.80.1 Detailed Description

Emits particles from objects position in a random direction.

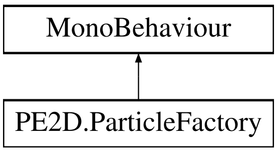
The documentation for this class was generated from the following file:

- pe2d/Particles/Emitters/ParticleEmitterInRandomDirection.cs

5.81 PE2D.ParticleFactory Class Reference

Creates and maintain an object pool of particles.

Inheritance diagram for PE2D.ParticleFactory:



Public Member Functions

- void CreateParticle (Vector2 position, Color colour, float duration, Vector2 initialScale, ParticleBuilder state)

  Creates a particle at position with the specified state.
- void RemoveAllActiveParticles ()

  Sets all enabled particles to be removed in the next time step.
- bool **ParticlesRemaining** ()
- List< CustomParticle > **GetLiveParticles** ()

Public Attributes

- GameObject particlePrefab

  Particle prefab.
- int maxParticleCount

  The max particle count. This number of particles is created at runtime and placed in a finite pool.

Properties

- static ParticleFactory instance `[get]`

  Gets the instance of this class. Can be called from any script. Only one instance of a particle factory can exist in one scene.

5.81.1 Detailed Description

Creates and maintain an object pool of particles.

5.81.2 Member Function Documentation

5.81.2.1 CreateParticle()

```
void PE2D.ParticleFactory.CreateParticle (
            Vector2 position,
            Color colour,
            float duration,
            Vector2 initialScale,
            ParticleBuilder state )
```

Creates a particle at position with the specified state.

Parameters

position	Initial position of particle.
tint	The initial colour of particle.
duration	The maximum duration of particle.
scale	Initial scale of particle.
state	THe particle state.

5.81.2.2 RemoveAllActiveParticles()

```
void PE2D.ParticleFactory.RemoveAllActiveParticles ( )
```

Sets all enabled particles to be removed in the next time step.

5.81.3 Member Data Documentation

5.81.3.1 maxParticleCount

```
int PE2D.ParticleFactory.maxParticleCount
```

The max particle count. This number of particles is created at runtime and placed in a finite pool.

5.81.3.2 particlePrefab

```
GameObject PE2D.ParticleFactory.particlePrefab
```

Particle prefab.

5.81.4 Property Documentation

5.81.4.1 instance

```
ParticleFactory PE2D.ParticleFactory.instance [static], [get]
```

Gets the instance of this class. Can be called from any script. Only one instance of a particle factory can exist in one scene.

The instance.

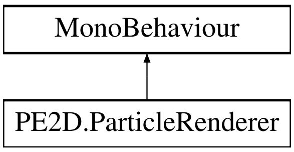
The documentation for this class was generated from the following file:

- pe2d/Particles/ParticleFactory.cs

5.82 PE2D.ParticleRenderer Class Reference

Simple renderer script for particles that disables the sprite renderer on enable and re-enables the srpite renderer after a time specified by ParticleRenderer::RENDERER_DELAY. Attach to the particle prefab to prevent occasional graphic glitches.

Inheritance diagram for PE2D.ParticleRenderer:



5.82.1 Detailed Description

Simple renderer script for particles that disables the sprite renderer on enable and re-enables the srpite renderer after a time specified by ParticleRenderer::RENDERER_DELAY. Attach to the particle prefab to prevent occasional graphic glitches.

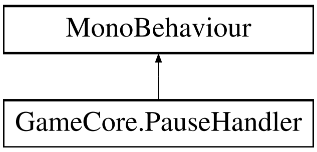
The documentation for this class was generated from the following file:

- pe2d/Particles/ParticleRenderer.cs

5.83 GameCore.PauseHandler Class Reference

Shows pause screen and hanles UI events from that scene.

Inheritance diagram for GameCore.PauseHandler:



Public Member Functions

- void EnableButton (float seconds)

  Enables pause button.
- void DisableButton ()

  Disables the pause button.
- void Pause ()

  Button event. Pauses the game. Pauses round entities and updates UI.
- void Restart ()

  Button event. Reloads game scene.
- void Resume ()

  Button event. Resumes game.

Public Attributes

- Text currentRoundText

  The text used to display the current round.
- Text highestRoundText

  The text used to display the highest round.
- Text pointsText

  The texts used to display the plauers current point total.
- Score scoreHandler

  The score handler, used to obtain the current point total.
- Button pauseButton

  The pause button.
- GameObject [ ] objectsToHide

  The objects to hide before showing scene.
- GameObject pauseMenu

  The parent GameObject of all pause menu UI items.
- bool isPaused

  The current pause status.

5.83.1 Detailed Description

Shows pause screen and hanles UI events from that scene.

5.83.2 Member Function Documentation

5.83.2.1 DisableButton()

```
void GameCore.PauseHandler.DisableButton ( )
```

Disables the pause button.

5.83.2.2 EnableButton()

```
void GameCore.PauseHandler.EnableButton (
             float seconds )
```

Enables pause button.

Parameters

seconds	The number of seconds before the pause button is enabled..

5.83.2.3 Pause()

```
void GameCore.PauseHandler.Pause ( )
```

Button event. Pauses the game. Pauses round entities and updates UI.

5.83.2.4 Restart()

```
void GameCore.PauseHandler.Restart ( )
```

Button event. Reloads game scene.

5.83.2.5 Resume()

```
void GameCore.PauseHandler.Resume ( )
```

Button event. Resumes game.

5.83.3 Member Data Documentation

5.83.3.1 currentRoundText

```
Text GameCore.PauseHandler.currentRoundText
```

The text used to display the current round.

5.83.3.2 highestRoundText

```
Text GameCore.PauseHandler.highestRoundText
```

The text used to display the highest round.

5.83.3.3 isPaused

```
bool GameCore.PauseHandler.isPaused
```

The current pause status.

5.83.3.4 objectsToHide

```
GameObject [] GameCore.PauseHandler.objectsToHide
```

The objects to hide before showing scene.

5.83.3.5 pauseButton

```
Button GameCore.PauseHandler.pauseButton
```

The pause button.

5.83.3.6 pauseMenu

```
GameObject GameCore.PauseHandler.pauseMenu
```

The parent GameObject of all pause menu UI items.

5.83.3.7 pointsText

```
Text GameCore.PauseHandler.pointsText
```

The texts used to display the plauers current point total.

5.83.3.8 scoreHandler

```
Score GameCore.PauseHandler.scoreHandler
```

The score handler, used to obtain the current point total.

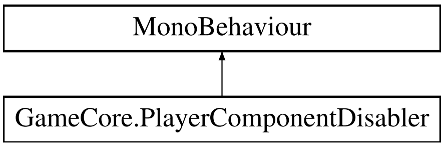
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/UI/PauseHandler.cs

5.84 GameCore.PlayerComponentDisabler Class Reference

Disables specified components when player is killed. Enables components when player is spawned.

Inheritance diagram for GameCore.PlayerComponentDisabler:



Public Attributes

- GameObject [ ] components

    The components to enable/disable.

5.84.1 Detailed Description

Disables specified components when player is killed. Enables components when player is spawned.

5.84.2 Member Data Documentation

5.84.2.1 components

```
GameObject [ ] GameCore.PlayerComponentDisabler.components
```

The components to enable/disable.

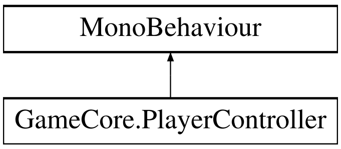
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/PlayerComponentDisabler.cs

5.85 GameCore.PlayerController Class Reference

Updates player position based on input.

Inheritance diagram for GameCore.PlayerController:



Public Member Functions

- void PauseMovement ()

  Pauses the movement.
- void ResumeMovement ()

  Resumes the movement.
- void IncrementSpeed (float increment=1f)

  Increments the players speed. Called when Ship Speed is purchased in store.
- void IncrementSpeedForSeconds (float increment, float time)

  Increments the players speed for seconds.

Public Attributes

- float desktopMovementSpeed = 10f

  The movment speed when not playing on mobile.
- float mobileMovementSpeed = 10f

  The movement speed when playing on mobile.

5.85.1 Detailed Description

Updates player position based on input.

5.85.2 Member Function Documentation

5.85.2.1 IncrementSpeed()

```
void GameCore.PlayerController.IncrementSpeed (
            float increment = 1f )
```

Increments the players speed. Called when Ship Speed is purchased in store.

Parameters

increment	Increment.

5.85.2.2 IncrementSpeedForSeconds()

```
void GameCore.PlayerController.IncrementSpeedForSeconds (
            float increment,
            float time )
```

Increments the players speed for seconds.

Parameters

increment	Amount to increase speed.
time	Time in seconds players speed is increased.

5.85.2.3 PauseMovement()

```
void GameCore.PlayerController.PauseMovement ( )
```

Pauses the movement.

5.85.2.4 ResumeMovement()

```
void GameCore.PlayerController.ResumeMovement ( )
```

Resumes the movement.

5.85.3 Member Data Documentation

5.85.3.1 desktopMovementSpeed

```
float GameCore.PlayerController.desktopMovementSpeed = 10f
```

The movment speed when not playing on mobile.

5.85.3.2 mobileMovementSpeed

```
float GameCore.PlayerController.mobileMovementSpeed = 10f
```

The movement speed when playing on mobile.

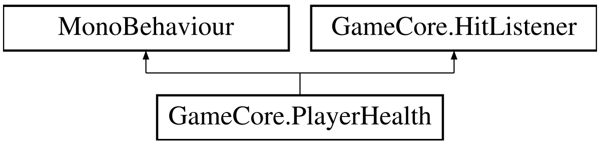
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/Movement/PlayerController.cs

5.86 GameCore.PlayerHealth Class Reference

Handles player health, applying damage, losing lives, and respawning.

Inheritance diagram for GameCore.PlayerHealth:



Public Member Functions

- void IncrementLives ()

  Increments the number of players lives.
- void OnHit (int damage)

  Removes a players life.

Public Attributes

- SpriteRenderer [ ] spriteRenderers

  The sprite renderers to enable/disable on death/respawn.
- int initialLives = 3

  The initial lives. Additional lives can be purchased by the player in the store.
- Color particleColourOnDeath

  The particle colour on death.
- Action OnDeath

  The on death action. Called when player loses a life.
- Action OnSpawn

  The on spawn action. Called when player spawns.
- AudioClip audioOnPlayerDeath

  The audio to player on player death.
- float secondsToRespawn = 2f

  The number of seconds it takes for the player to respawn.

5.86.1 Detailed Description

Handles player health, applying damage, losing lives, and respawning.

5.86.2 Member Function Documentation

5.86.2.1 IncrementLives()

```
void GameCore.PlayerHealth.IncrementLives ( )
```

Increments the number of players lives.

5.86.2.2 OnHit()

```
void GameCore.PlayerHealth.OnHit (
             int damage )
```

Removes a players life.

Parameters

damage	Damage taken.

Implements GameCore.HitListener.

5.86.3 Member Data Documentation

5.86.3.1 audioOnPlayerDeath

```
AudioClip GameCore.PlayerHealth.audioOnPlayerDeath
```

The audio to player on player death.

5.86.3.2 initialLives

```
int GameCore.PlayerHealth.initialLives = 3
```

The initial lives. Additional lives can be purchased by the player in the store.

5.86.3.3 OnDeath

```
Action GameCore.PlayerHealth.OnDeath
```

The on death action. Called when player loses a life.

5.86.3.4 OnSpawn

```
Action GameCore.PlayerHealth.OnSpawn
```

The on spawn action. Called when player spawns.

5.86.3.5 particleColourOnDeath

```
Color GameCore.PlayerHealth.particleColourOnDeath
```

The particle colour on death.

5.86.3.6 secondsToRespawn

```
float GameCore.PlayerHealth.secondsToRespawn = 2f
```

The number of seconds it takes for the player to respawn.

5.86.3.7 spriteRenderers

```
SpriteRenderer [] GameCore.PlayerHealth.spriteRenderers
```

The sprite renderers to enable/disable on death/respawn.

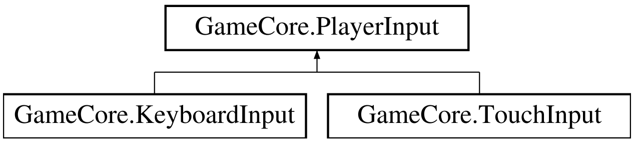
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/PlayerHealth.cs

5.87 GameCore.PlayerInput Interface Reference

Contract for getting players next move.

Inheritance diagram for GameCore.PlayerInput:



Public Member Functions

- float GetMovementSpeed ()

  Gets the players movement speed.
- void SetMovementSpeed (float amount)

  Sets the players movement speed.
- Vector2 GetVelocity ()

  Gets the velocity. The players next move.

5.87.1 Detailed Description

Contract for getting players next move.

5.87.2 Member Function Documentation

5.87.2.1 GetMovementSpeed()

```
float GameCore.PlayerInput.GetMovementSpeed ( )
```

Gets the players movement speed.

Returns

The movement speed.

Implemented in GameCore.TouchInput, and GameCore.KeyboardInput.

5.87.2.2 GetVelocity()

```
Vector2 GameCore.PlayerInput.GetVelocity ( )
```

Gets the velocity. The players next move.

Returns

The velocity.

Implemented in GameCore.TouchInput, and GameCore.KeyboardInput.

5.87.2.3 SetMovementSpeed()

```
void GameCore.PlayerInput.SetMovementSpeed (
             float amount )
```

Sets the players movement speed.

Parameters

amount	Move speed.

Implemented in GameCore.TouchInput, and GameCore.KeyboardInput.

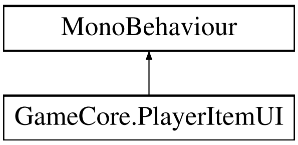
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Player/Movement/PlayerInput.cs

5.88 GameCore.PlayerItemUI Class Reference

Shows number of lives and bombs on the in-game UI the player currently has.

Inheritance diagram for GameCore.PlayerItemUI:



Public Member Functions

- void SetItemCount (int count)

  Updates the count in the UI.

Public Attributes

- Text livesText

  The text object used to display the number of lives/bombs.

5.88.1 Detailed Description

Shows number of lives and bombs on the in-game UI the player currently has.

5.88.2 Member Function Documentation

5.88.2.1 SetItemCount()

```
void GameCore.PlayerItemUI.SetItemCount (
         int count )
```

Updates the count in the UI.

Parameters

count	New count.

5.88.3 Member Data Documentation

5.88.3.1 livesText

```
Text GameCore.PlayerItemUI.livesText
```

The text object used to display the number of lives/bombs.

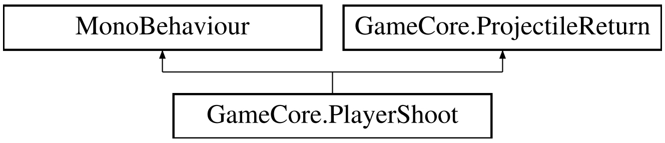
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/PlayerItemUI.cs

5.89 GameCore.PlayerShoot Class Reference

Provides shoot functionality for the player. Projectiles are retrieved from a pool. Also provides burst functionality.

Inheritance diagram for GameCore.PlayerShoot:



Public Member Functions

- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void BeginShooting ()

  Begin shooting.
- void PoolProjectile (Projectile projectile)

  Adds the projectile to a pool for later use.
- void DecrementSecBetweenShotsForSeconds (float decrement, float seconds)

  Decrements the seconds between shots. The effect lasts for the number of seconds passed.
- void DecrementSecBetweenShots (float decrement)

  Decrements the seconds between shots permantly.
- void IncrementDamage (int increment=1)

  Increases the damage of the projectiles.
- void IncrementShotBurst (int increment=1)

  Increment the number of shots in a burst.
- void DoubleShootingForSeconds (float seconds)

  Adds temporary powerup, enabling player to shoot two parallel projectiles.

Public Attributes

- int damage = 1

  The projectile damage.
- GameObject bulletPrefab

  The bullet prefab to spawn.
- float secsBetweenShot = 0.2f

  The seconds between projectile release.
- AudioClip audioOnShoot

  The audio to play on shoot.
- int bulletsPerBurst = 1

  The number of bullets per burst. This amount is increased through shop purchases.
- float secDelayBetweenBulletsInBurst = 0.2f

  The second delay between bullets in a burst.
- int numToPool = 12

  The number of projectiles to create at the beginning of the game.

5.89.1 Detailed Description

Provides shoot functionality for the player. Projectiles are retrieved from a pool. Also provides burst functionality.

5.89.2 Member Function Documentation

5.89.2.1 BeginShooting()

```
void GameCore.PlayerShoot.BeginShooting ( )
```

Begin shooting.

5.89.2.2 DecrementSecBetweenShots()

```
void GameCore.PlayerShoot.DecrementSecBetweenShots (
             float decrement )
```

Decrements the seconds between shots permantly.

Parameters

decrement	The amount to decrement the time between shots.

5.89.2.3 DecrementSecBetweenShotsForSeconds()

```
void GameCore.PlayerShoot.DecrementSecBetweenShotsForSeconds (
             float decrement,
             float seconds )
```

Decrements the seconds between shots. The effect lasts for the number of seconds passed.

Parameters

decrement	The amount to decrement the time between shots.
seconds	The amount of time the decrement lasts.

5.89.2.4 DoubleShootingForSeconds()

```
void GameCore.PlayerShoot.DoubleShootingForSeconds (
             float seconds )
```

Adds temporary powerup, enabling player to shoot two parallel projectiles.

Parameters

seconds	The seconds the powerup lasts.

5.89.2.5 IncrementDamage()

```
void GameCore.PlayerShoot.IncrementDamage (
             int increment = 1 )
```

Increases the damage of the projectiles.

Parameters

increment	The amount to increment damage.

5.89.2.6 IncrementShotBurst()

```
void GameCore.PlayerShoot.IncrementShotBurst (
             int increment = 1 )
```

Increment the number of shots in a burst.

Parameters

increment	The number of bullets to add to each burst.

5.89.2.7 Pause()

```
void GameCore.PlayerShoot.Pause ( )
```

Pause this instance.

5.89.2.8 PoolProjectile()

```
void GameCore.PlayerShoot.PoolProjectile (
            Projectile projectile )
```

Adds the projectile to a pool for later use.

Parameters

p	The projectile to pool.
projectile	Projectile.

Implements GameCore.ProjectileReturn.

5.89.2.9 Resume()

```
void GameCore.PlayerShoot.Resume ( )
```

Resume this instance.

5.89.3 Member Data Documentation

5.89.3.1 audioOnShoot

```
AudioClip GameCore.PlayerShoot.audioOnShoot
```

The audio to play on shoot.

5.89.3.2 bulletPrefab

```
GameObject GameCore.PlayerShoot.bulletPrefab
```

The bullet prefab to spawn.

5.89.3.3 bulletsPerBurst

```
int GameCore.PlayerShoot.bulletsPerBurst = 1
```

The number of bullets per burst. This amount is increased through shop purchases.

5.89.3.4 damage

```
int GameCore.PlayerShoot.damage = 1
```

The projectile damage.

5.89.3.5 numToPool

```
int GameCore.PlayerShoot.numToPool = 12
```

The number of projectiles to create at the beginning of the game.

5.89.3.6 secDelayBetweenBulletsInBurst

```
float GameCore.PlayerShoot.secDelayBetweenBulletsInBurst = 0.2f
```

The second delay between bullets in a burst.

5.89.3.7 secsBetweenShot

```
float GameCore.PlayerShoot.secsBetweenShot = 0.2f
```

The seconds between projectile release.

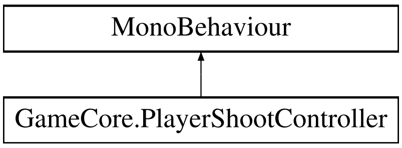
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/PlayerShoot.cs

5.90 GameCore.PlayerShootController Class Reference

Controls all player weapons. Enables the pausing and resuming of shooting i.e. between rounds, or when the player dies/respawns.

Inheritance diagram for GameCore.PlayerShootController:



Public Member Functions

- void BeginShooting ()

  Begins the shooting of each weapon.
- void PauseAll ()

  Pauses all.
- void ResumeAll ()

  Resumes all.

5.90.1 Detailed Description

Controls all player weapons. Enables the pausing and resuming of shooting i.e. between rounds, or when the player dies/respawns.

5.90.2 Member Function Documentation

5.90.2.1 BeginShooting()

```
void GameCore.PlayerShootController.BeginShooting ( )
```

Begins the shooting of each weapon.

5.90.2.2 PauseAll()

```
void GameCore.PlayerShootController.PauseAll ( )
```

Pauses all.

5.90.2.3 ResumeAll()

```
void GameCore.PlayerShootController.ResumeAll ( )
```

Resumes all.

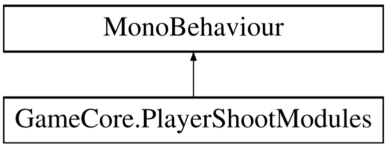
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/PlayerShootController.cs

5.91 GameCore.PlayerShootModules Class Reference

Provides functionality to add new shoot modules to player (when purchased through the store).

Inheritance diagram for GameCore.PlayerShootModules:



Public Member Functions

- bool IsActionable ()

  Determines whether a new module can be enabled.
- int GetNumberOfActionableModules ()

  Gets the number of modules that can be activated.
- void EnableNewModule ()

  Enables a new module.

Public Attributes

- GameObject [ ] shootModules

  The shoot modules the can be enabled in game.

5.91.1 Detailed Description

Provides functionality to add new shoot modules to player (when purchased through the store).

5.91.2 Member Function Documentation

5.91.2.1 EnableNewModule()

```
void GameCore.PlayerShootModules.EnableNewModule ( )
```

Enables a new module.

5.91.2.2 GetNumberOfActionableModules()

```
int GameCore.PlayerShootModules.GetNumberOfActionableModules ( )
```

Gets the number of modules that can be activated.

Returns

The number of modules that can be activated.

5.91.2.3 IsActionable()

```
bool GameCore.PlayerShootModules.IsActionable ( )
```

Determines whether a new module can be enabled.

Returns

`true` if this instance is actionable; otherwise, `false`.

5.91.3 Member Data Documentation

5.91.3.1 shootModules

```
GameObject [] GameCore.PlayerShootModules.shootModules
```

The shoot modules the can be enabled in game.

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

- Pew Pew/Scripts/Player/PlayerShootModules.cs

5.92 WarpGrid.PointMass Class Reference

A moveable point on the grid.

Public Member Functions

- PointMass (Vector2 position, float invMass)

  Initializes a new instance of the PointMass class.
- void ApplyForce (Vector2 force)

  Applies a force to the point.
- void IncreaseDamping (float factor)

  Dampens the effect of force application.
- void Update ()

  Update this instance. Updates velocity and position of point/

Public Attributes

- Vector2 Position

  Current point position.
- Vector2 Velocity

  Current point velocity.
- float InverseMass

  The inverse mass of the point (lower numbers result in a point with a higher mass).

5.92.1 Detailed Description

A moveable point on the grid.

5.92.2 Constructor & Destructor Documentation

5.92.2.1 PointMass()

```
WarpGrid.PointMass.PointMass (
             Vector2 position,
             float invMass )
```

Initializes a new instance of the PointMass class.

Parameters

position	Position.
invMass	Inv mass.

5.92.3 Member Function Documentation

5.92.3.1 ApplyForce()

```
void WarpGrid.PointMass.ApplyForce (
             Vector2 force )
```

Applies a force to the point.

Parameters

force	Force.

5.92.3.2 IncreaseDamping()

```
void WarpGrid.PointMass.IncreaseDamping (
             float factor )
```

Dampens the effect of force application.

Parameters

factor	Factor.

5.92.3.3 Update()

```
void WarpGrid.PointMass.Update ( )
```

Update this instance. Updates velocity and position of point/

5.92.4 Member Data Documentation

5.92.4.1 InverseMass

```
float WarpGrid.PointMass.InverseMass
```

The inverse mass of the point (lower numbers result in a point with a higher mass).

5.92.4.2 Position

```
Vector2 WarpGrid.PointMass.Position
```

Current point position.

5.92.4.3 Velocity

```
Vector2 WarpGrid.PointMass.Velocity
```

Current point velocity.

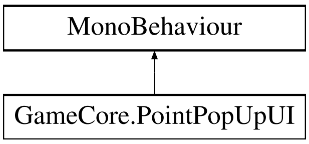
The documentation for this class was generated from the following file:

- UG/Scripts/PointMass.cs

5.93 GameCore.PointPopUpUI Class Reference

Shows pop up text when player collects a particle and updates players score.

Inheritance diagram for GameCore.PointPopUpUI:



Public Member Functions

- void ShowAtPosition (Vector3 position)

  Shows point text at position.
- void ShowTextAtPosition (object text, Vector2 position)

  Shows the specified text at position.

Public Attributes

- GameObject pointsTextPrefab

  The points text prefab.
- AudioClip audioOnPoint

  The audio to play on point.

5.93.1 Detailed Description

Shows pop up text when player collects a particle and updates players score.

5.93.2 Member Function Documentation

5.93.2.1 ShowAtPosition()

```
void GameCore.PointPopUpUI.ShowAtPosition (
             Vector3 position )
```

Shows point text at position.

Parameters

position	Position to show point text.

5.93.2.2 ShowTextAtPosition()

```
void GameCore.PointPopUpUI.ShowTextAtPosition (
             object text,
             Vector2 position )
```

Shows the specified text at position.

Parameters

text	Text to display.
position	Position to show text.

5.93.3 Member Data Documentation

5.93.3.1 audioOnPoint

```
AudioClip GameCore.PointPopUpUI.audioOnPoint
```

The audio to play on point.

5.93.3.2 pointsTextPrefab

```
GameObject GameCore.PointPopUpUI.pointsTextPrefab
```

The points text prefab.

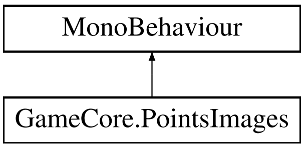
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Score/PointPopUpUI.cs

5.94 GameCore.PointsImages Class Reference

Handles enabling and disabling of points images (used to signify how many instances of an item have been purchased).

Inheritance diagram for GameCore.PointsImages:



Public Member Functions

- int GetNumberEnabled ()

  Gets the number of enabled points images.
- void EnableNextPointImage ()

  Enables the next point image.
- void DisableImages (int numToDisable)

  Disables the number of images. First checks if that many images are enabled. If not an error is output.

Public Attributes

- Image [ ] images

  The points images to enable/disable.

5.94.1 Detailed Description

Handles enabling and disabling of points images (used to signify how many instances of an item have been purchased).

5.94.2 Member Function Documentation

5.94.2.1 DisableImages()

```
void GameCore.PointsImages.DisableImages (
            int numToDisable )
```

Disables the number of images. First checks if that many images are enabled. If not an error is output.

Parameters

numToDisable	Number to disable.

5.94.2.2 EnableNextPointImage()

```
void GameCore.PointsImages.EnableNextPointImage ( )
```

Enables the next point image.

5.94.2.3 GetNumberEnabled()

```
int GameCore.PointsImages.GetNumberEnabled ( )
```

Gets the number of enabled points images.

Returns

The number enabled.

5.94.3 Member Data Documentation

5.94.3.1 images

```
Image [] GameCore.PointsImages.images
```

The points images to enable/disable.

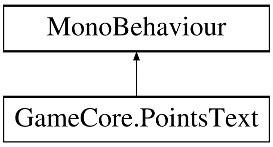
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/PointsImages.cs

5.95 GameCore.PointsText Class Reference

Attached to each points text. Handles text movement and fade out.

Inheritance diagram for GameCore.PointsText:



Public Member Functions

- void Show (Vector3 position)

  Show the point text at the specified position.
- void SetScore (int score)

  Sets text to string value of score.
- void SetText (object text)

  Sets the text.

Public Attributes

- float moveSpeed = 1f

  The upwards movement speed.

5.95.1 Detailed Description

Attached to each points text. Handles text movement and fade out.

5.95.2 Member Function Documentation

5.95.2.1 SetScore()

```
void GameCore.PointsText.SetScore (
             int score )
```

Sets text to string value of score.

Parameters

score	Score to display.

5.95.2.2 SetText()

```
void GameCore.PointsText.SetText (
             object text )
```

Sets the text.

Parameters

text	Text to display.

5.95.2.3 Show()

```
void GameCore.PointsText.Show (
             Vector3 position )
```

Show the point text at the specified position.

Parameters

position	Position to show points text.

5.95.3 Member Data Documentation

5.95.3.1 moveSpeed

```
float GameCore.PointsText.moveSpeed = 1f
```

The upwards movement speed.

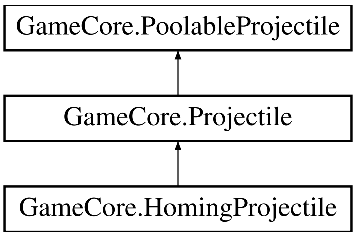
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Score/PointsText.cs

5.96 GameCore.PoolableProjectile Interface Reference

Contract for any projectile that can be returned to a pool.

Inheritance diagram for GameCore.PoolableProjectile:



Public Member Functions

- void ReturnProjectile ()

  Returns the projectile to an object pool.

Properties

- int damage `[get]`

  Gets the damage of projectile.

5.96.1 Detailed Description

Contract for any projectile that can be returned to a pool.

5.96.2 Member Function Documentation

5.96.2.1 ReturnProjectile()

```
void GameCore.PoolableProjectile.ReturnProjectile ( )
```

Returns the projectile to an object pool.

Implemented in GameCore.Projectile.

5.96.3 Property Documentation

5.96.3.1 damage

```
int GameCore.PoolableProjectile.damage  [get]
```

Gets the damage of projectile.

The damage.

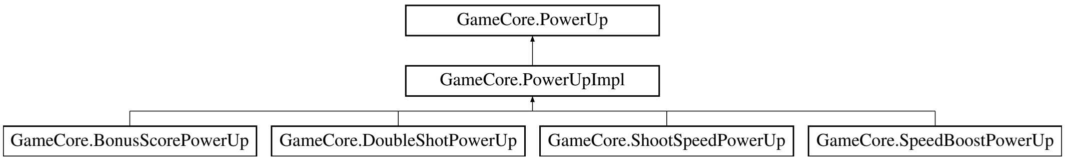
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Projectiles/Projectile.cs

5.97 GameCore.PowerUp Interface Reference

Contract for all in-game powerups.

Inheritance diagram for GameCore.PowerUp:


GameCore.PowerUp
GameCore.PowerUpImpl
GameCore.BonusScorePowerUp
GameCore.DoubleShotPowerUp
GameCore.ShootSpeedPowerUp
GameCore.SpeedBoostPowerUp

Public Member Functions

- void Perform (Transform player)

  Perform the specified powerup action.

5.97.1 Detailed Description

Contract for all in-game powerups.

5.97.2 Member Function Documentation

5.97.2.1 Perform()

```
void GameCore.PowerUp.Perform (
             Transform player )
```

Perform the specified powerup action.

Parameters

player	Player tranform.

Implemented in GameCore.PowerUpImpl, GameCore.ShootSpeedPowerUp, GameCore.SpeedBoostPowerUp, GameCore.DoubleShotPowerUp, and GameCore.BonusScorePowerUp.

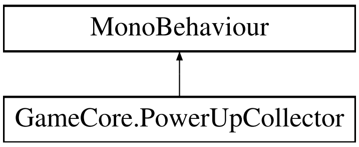
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Powerups/PowerUp.cs

5.98 GameCore.PowerUpCollector Class Reference

Functionality for collecting and activating powerups.

Inheritance diagram for GameCore.PowerUpCollector:



5.98.1 Detailed Description

Functionality for collecting and activating powerups.

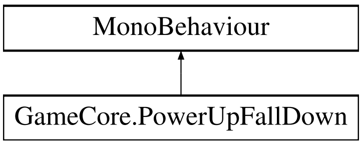
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/PowerUpCollector.cs

5.99 GameCore.PowerUpFallDown Class Reference

Attached to each powerup. Enables powerups to fall into a position where they can be picked up by the player.

Inheritance diagram for GameCore.PowerUpFallDown:



Public Attributes

- float movementSpeed = 1f

  The negative y movement speed.
- float minY = -2.9f

  The target y position.

5.99.1 Detailed Description

Attached to each powerup. Enables powerups to fall into a position where they can be picked up by the player.

5.99.2 Member Data Documentation

5.99.2.1 minY

```
float GameCore.PowerUpFallDown.minY = -2.9f
```

The target y position.

5.99.2.2 movementSpeed

```
float GameCore.PowerUpFallDown.movementSpeed = 1f
```

The negative y movement speed.

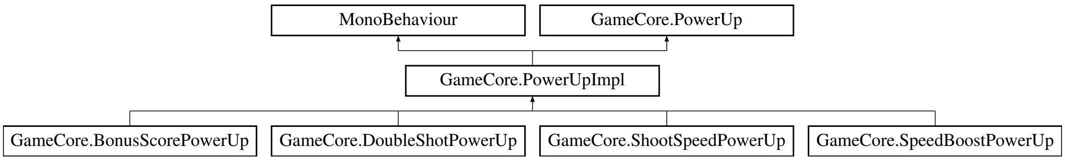
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/PowerUpFallDown.cs

5.100 GameCore.PowerUpImpl Class Reference

The abstract base class for all powerups. Provides access to UI text system (to show powerup name) and any common fields.

Inheritance diagram for GameCore.PowerUpImpl:


MonoBehaviour
GameCore.PowerUp
GameCore.PowerUpImpl
GameCore.BonusScorePowerUp
GameCore.DoubleShotPowerUp
GameCore.ShootSpeedPowerUp
GameCore.SpeedBoostPowerUp

Public Member Functions

- abstract void Perform (Transform player)

  Perform the specified powerup action. Implement in concrete base classes.

Public Attributes

- float maxTimeAlive = 2f

  The maximum time the powerup can be on the floor before either dissapearing or bring picked up by the player.
- float flashTime = 1.5f

  The seconds before the powerup starts flashing. Used to indicate to the player that the powerup will shortly be removed from the game unless they pick it up.
- float timeBetweenFlashes = 0.07f

  The time between flashes.
- Color particleColour

  The colour of particles spawned when powerup is picked up.
- int numOfParticlesToSpawn = 10

  The number of particles to spawn when player is picked up.

Protected Member Functions

- virtual void **Start** ()
- void **ShowMessage** (object msg)

Static Protected Attributes

- static PowerUpParticleExplosion **PARTICLE_EXPLOSION**

5.100.1 Detailed Description

The abstract base class for all powerups. Provides access to UI text system (to show powerup name) and any common fields.

5.100.2 Member Function Documentation

5.100.2.1 Perform()

```
abstract void GameCore.PowerUpImpl.Perform (
             Transform player )  [pure virtual]
```

Perform the specified powerup action. Implement in concrete base classes.

Parameters

player	Player tranform.

Implements GameCore.PowerUp.

Implemented in GameCore.ShootSpeedPowerUp, GameCore.SpeedBoostPowerUp, GameCore.DoubleShotPowerUp, and GameCore.BonusScorePowerUp.

5.100.3 Member Data Documentation

5.100.3.1 flashTime

```
float GameCore.PowerUpImpl.flashTime = 1.5f
```

The seconds before the powerup starts flashing. Used to indicate to the player that the powerup will shortly be removed from the game unless they pick it up.

5.100.3.2 maxTimeAlive

```
float GameCore.PowerUpImpl.maxTimeAlive = 2f
```

The maximum time the powerup can be on the floor before either dissapearing or bring picked up by the player.

5.100.3.3 numOfParticlesToSpawn

```
int GameCore.PowerUpImpl.numOfParticlesToSpawn = 10
```

The number of particles to spawn when player is picked up.

5.100.3.4 particleColour

```
Color GameCore.PowerUpImpl.particleColour
```

The colour of particles spawned when powerup is picked up.

5.100.3.5 timeBetweenFlashes

```
float GameCore.PowerUpImpl.timeBetweenFlashes = 0.07f
```

The time between flashes.

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

- Pew Pew/Scripts/Powerups/PowerUp.cs

5.101 GameCore.PowerUpParticleExplosion Class Reference

Spawns particle explosion on particle pick up.

Public Member Functions

- void Spawn (Vector2 position, int numOfParticles, Color particleColour)
  Spawn the specified particle explosion at position, with numOfParticles and of particleColour.

5.101.1 Detailed Description

Spawns particle explosion on particle pick up.

5.101.2 Member Function Documentation

5.101.2.1 Spawn()

```
void GameCore.PowerUpParticleExplosion.Spawn (
             Vector2 position,
             int numOfParticles,
             Color particleColour )
```

Spawn the specified particle explosion at position, with numOfParticles and of particleColour.

Parameters

position	The position of explosion.
numOfParticles	The number of particles.
particleColour	The particle colour.

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

- Pew Pew/Scripts/Powerups/PowerUp.cs

5.102 GameCore.PowerUpSpawn Class Reference

Data class for powerup spawns.

Public Attributes

- GameObject powerUpPrefab

  The prefab to spawn.
- float weight

  The relative chance to spawn this powerup.

5.102.1 Detailed Description

Data class for powerup spawns.

5.102.2 Member Data Documentation

5.102.2.1 powerUpPrefab

```
GameObject GameCore.PowerUpSpawn.powerUpPrefab
```

The prefab to spawn.

5.102.2.2 weight

```
float GameCore.PowerUpSpawn.weight
```

The relative chance to spawn this powerup.

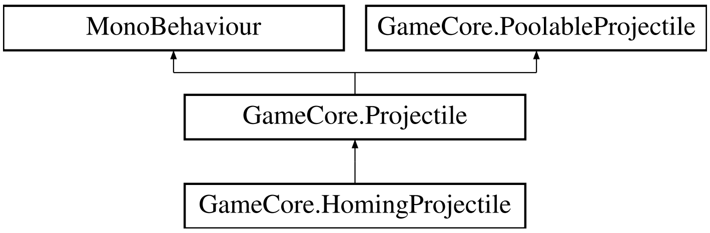
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/DropPowerUpOnDeath.cs

5.103 GameCore.Projectile Class Reference

The standard projectile. Is poolable and effected by blackhole and repel GameObjects.

Inheritance diagram for GameCore.Projectile:



Public Member Functions

- void Initialise (ProjectileReturn owner, Vector2 dir, int damage)

  Initialise the specified projectile with: owner, direction and damage. Applies force to move the projectile in the desired direction.
- void ReturnProjectile ()

  Returns the projectile to an object pool.
- void Pause ()

  Pause this instance. Stores velocity at pause (used in resume).
- void Resume ()

  Resume this instance. Restores velocity saved when paused.

Public Attributes

- float moveForce = 100f

  The force applied to the Rigidbody attached to the projectile.
- float timeAlive = 2f

  The maximum time the projectile is alive before being returned to a pool.
- float effectorMultiplier = 200f

  Effectors force is multipled by this.

Protected Member Functions

- virtual void **Awake** ()
- virtual void **Update** ()

Protected Attributes

- Rigidbody2D **m_Rigidbody2D**
- float **m_CurrentTimeAlive** = 0f
- bool **m_Paused** = false

Properties

- int damage `[get]`

  Gets the damage of projectile.

5.103.1 Detailed Description

The standard projectile. Is poolable and effected by blackhole and repel GameObjects.

5.103.2 Member Function Documentation

5.103.2.1 Initialise()

```
void GameCore.Projectile.Initialise (
             ProjectileReturn owner,
             Vector2 dir,
             int damage )
```

Initialise the specified projectile with: owner, direction and damage. Applies force to move the projectile in the desired direction.

Parameters

owner	The object pool.
dir	Movement direction.
damage	Damage.

5.103.2.2 Pause()

```
void GameCore.Projectile.Pause ( )
```

Pause this instance. Stores velocity at pause (used in resume).

5.103.2.3 Resume()

```
void GameCore.Projectile.Resume ( )
```

Resume this instance. Restores velocity saved when paused.

5.103.2.4 ReturnProjectile()

```
void GameCore.Projectile.ReturnProjectile ( )
```

Returns the projectile to an object pool.

Implements GameCore.PoolableProjectile.

5.103.3 Member Data Documentation

5.103.3.1 effectorMultiplier

```
float GameCore.Projectile.effectorMultiplier = 200f
```

Effectors force is multipled by this.

5.103.3.2 moveForce

```
float GameCore.Projectile.moveForce = 100f
```

The force applied to the Rigidbody attached to the projectile.

5.103.3.3 timeAlive

```
float GameCore.Projectile.timeAlive = 2f
```

The maximum time the projectile is alive before being returned to a pool.

5.103.4 Property Documentation

5.103.4.1 damage

```
int GameCore.Projectile.damage  [get]
```

Gets the damage of projectile.

The damage.

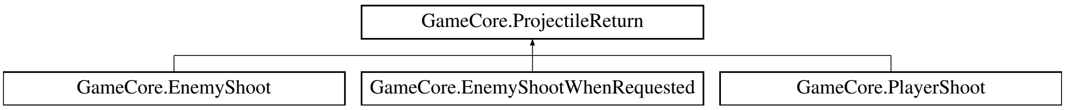
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Projectiles/Projectile.cs

5.104 GameCore.ProjectileReturn Interface Reference

A contract for any entity that can pool a projectile.

Inheritance diagram for GameCore.ProjectileReturn:



Public Member Functions

- void PoolProjectile (Projectile p)

  Adds the projectile to a pool.

5.104.1 Detailed Description

A contract for any entity that can pool a projectile.

5.104.2 Member Function Documentation

5.104.2.1 PoolProjectile()

```
void GameCore.ProjectileReturn.PoolProjectile (
             Projectile p )
```

Adds the projectile to a pool.

Parameters

p	The projectile to pool.

Implemented in GameCore.EnemyShoot, GameCore.PlayerShoot, and GameCore.EnemyShootWhenRequested.

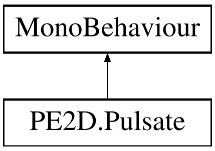
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Player/PlayerShoot.cs

5.105 PE2D.Pulsate Class Reference

Simple script used to pulse an objects size. Used in the demo scene for the effectors.

Inheritance diagram for PE2D.Pulsate:



5.105.1 Detailed Description

Simple script used to pulse an objects size. Used in the demo scene for the effectors.

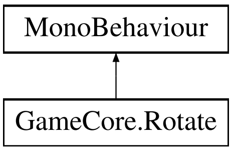
The documentation for this class was generated from the following file:

- pe2d/Pulsate.cs

5.106 GameCore.Rotate Class Reference

Rotates GameObject on z axis.

Inheritance diagram for GameCore.Rotate:



Public Member Functions

- void Activate ()

  Begins rotation. If waitToActivate is false, rotation is started in Start method.

Public Attributes

- float rotateSpeed = 80f

  The target revolution in degrees per second (e.g. 360 = 1 full rotation per second).
- bool randomSign = false

  The object has a 50% chance to rotate either left or right.
- bool waitToActivate = false

  The rotation will not begin until Activate is called.

5.106.1 Detailed Description

Rotates GameObject on z axis.

5.106.2 Member Function Documentation

5.106.2.1 Activate()

```
void GameCore.Rotate.Activate ( )
```

Begins rotation. If waitToActivate is false, rotation is started in Start method.

5.106.3 Member Data Documentation

5.106.3.1 randomSign

```
bool GameCore.Rotate.randomSign = false
```

The object has a 50% chance to rotate either left or right.

5.106.3.2 rotateSpeed

```
float GameCore.Rotate.rotateSpeed = 80f
```

The target revolution in degrees per second (e.g. 360 = 1 full rotation per second).

5.106.3.3 waitToActivate

```
bool GameCore.Rotate.waitToActivate = false
```

The rotation will not begin until Activate is called.

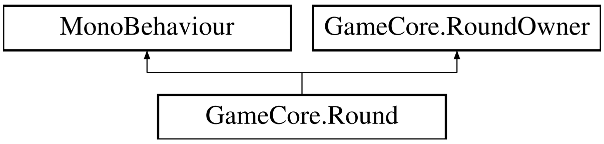
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Utilities/Rotate.cs

5.107 GameCore.Round Class Reference

Responsible for round progression: starting and signifying to the GameManager that the round is complete.

Inheritance diagram for GameCore.Round:



Public Types

- enum RoundType { **Wave**, **Challenge**, **Boss** }

  Round types.

Public Member Functions

- void StartRound ()

  Starts the round. Calls each enemies respective begin methods. Delays starting shooting for 1 second.
- void RemoveEnemyFromRound (RoundEnemy enemy)

  Registers enemy removed from round.
- void EnemyEscapedRound (RoundEnemy enemy)

  Registers enemy escaped round.

Public Attributes

- Action onEnemyRemoved

  Invoked every time an enemy is removed.
- RoundType roundType = RoundType.Wave

  The type of the round.
- List< EnemyHealth > enemies = new List<EnemyHealth>()

  All non-stationary enemies within the round. Important: enemies that are destroyed are not removed from this list. Null checks are required.

Properties

- int maxEnemies `[get]`

  Gets the number of enemies first spawned.
- int enemiesRemaining `[get]`

  Gets the number of enemies remaining.

5.107.1 Detailed Description

Responsible for round progression: starting and signifying to the GameManager that the round is complete.

5.107.2 Member Enumeration Documentation

5.107.2.1 RoundType

```
enum GameCore.Round.RoundType  [strong]
```

Round types.

5.107.3 Member Function Documentation

5.107.3.1 EnemyEscapedRound()

```
void GameCore.Round.EnemyEscapedRound (
             RoundEnemy enemy )
```

Registers enemy escaped round.

Parameters

enemy	Enemy.

Implements GameCore.RoundOwner.

5.107.3.2 RemoveEnemyFromRound()

```
void GameCore.Round.RemoveEnemyFromRound (
             RoundEnemy enemy )
```

Registers enemy removed from round.

Parameters

enemy	Enemy.

Implements GameCore.RoundOwner.

5.107.3.3 StartRound()

```
void GameCore.Round.StartRound ( )
```

Starts the round. Calls each enemies respective begin methods. Delays starting shooting for 1 second.

5.107.4 Member Data Documentation

5.107.4.1 enemies

```
List<EnemyHealth> GameCore.Round.enemies = new List<EnemyHealth>()
```

All non-stationary enemies within the round. Important: enemies that are destroyed are not removed from this list. Null checks are required.

5.107.4.2 onEnemyRemoved

```
Action GameCore.Round.onEnemyRemoved
```

Invoked every time an enemy is removed.

5.107.4.3 roundType

```
RoundType GameCore.Round.roundType = RoundType.Wave
```

The type of the round.

5.107.5 Property Documentation

5.107.5.1 enemiesRemaining

```
int GameCore.Round.enemiesRemaining [get]
```

Gets the number of enemies remaining.

The enemies remaining.

5.107.5.2 maxEnemies

```
int GameCore.Round.maxEnemies  [get]
```

Gets the number of enemies first spawned.

The max enemies.

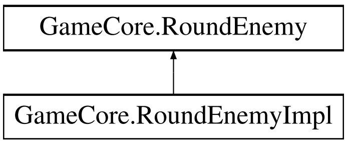
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/Round.cs

5.108 GameCore.RoundEnemy Interface Reference

Contract for an enemy that is part of a pround.

Inheritance diagram for GameCore.RoundEnemy:



Public Member Functions

- void RegisterRoundOwner (RoundOwner round)

  Registers the round owner. Used to let round owner know when it has been removed from the round.
- void EscapedWave ()

  Signifies that the enemy has escaped the round.

Properties

- Transform myTransform `[get]`

  Gets entities transform.

5.108.1 Detailed Description

Contract for an enemy that is part of a pround.

5.108.2 Member Function Documentation

5.108.2.1 EscapedWave()

```
void GameCore.RoundEnemy.EscapedWave ( )
```

Signifies that the enemy has escaped the round.

Implemented in GameCore.RoundEnemyImpl.

5.108.2.2 RegisterRoundOwner()

```
void GameCore.RoundEnemy.RegisterRoundOwner (
             RoundOwner round )
```

Registers the round owner. Used to let round owner know when it has been removed from the round.

Parameters

round	Round.

Implemented in GameCore.RoundEnemyImpl.

5.108.3 Property Documentation

5.108.3.1 myTransform

```
Transform GameCore.RoundEnemy.myTransform  [get]
```

Gets entities transform.

Transform.

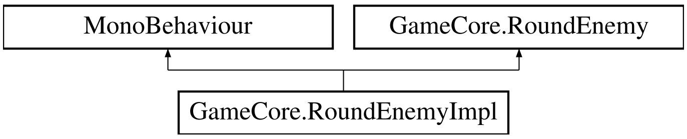
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Round/RoundEnemyImpl.cs

5.109 GameCore.RoundEnemyImpl Class Reference

Round enemy implementation. Informs roundowner when entity has been killed or has escaped round.

Inheritance diagram for GameCore.RoundEnemyImpl:



Public Member Functions

- void RegisterRoundOwner (RoundOwner round)

  Registers the round owner. Used to let round owner know when it has been removed from the round.
- void EscapedWave ()

  Signifies that the enemy has escaped the round.

Properties

- Transform myTransform `[get]`

  Gets entities transform.

5.109.1 Detailed Description

Round enemy implementation. Informs roundowner when entity has been killed or has escaped round.

5.109.2 Member Function Documentation

5.109.2.1 EscapedWave()

```
void GameCore.RoundEnemyImpl.EscapedWave ( )
```

Signifies that the enemy has escaped the round.

Implements GameCore.RoundEnemy.

5.109.2.2 RegisterRoundOwner()

```
void GameCore.RoundEnemyImpl.RegisterRoundOwner (
            RoundOwner round )
```

Registers the round owner. Used to let round owner know when it has been removed from the round.

Parameters

round	Round.

Implements GameCore.RoundEnemy.

5.109.3 Property Documentation

5.109.3.1 myTransform

```
Transform GameCore.RoundEnemyImpl.myTransform  [get]
```

Gets entities transform.

Transform.

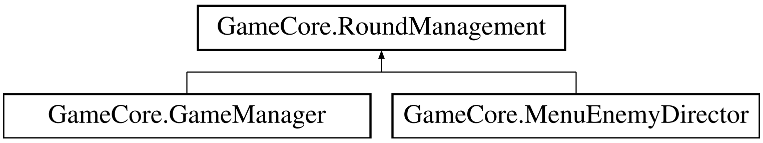
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/RoundEnemyImpl.cs

5.110 GameCore.RoundManagement Interface Reference

Contract for any class that can perform actions when a round or challenge round finishes.

Inheritance diagram for GameCore.RoundManagement:



Public Member Functions

- void **OnChallengeRoundOver** (int enemiesKilled, int maxEnemies)
- void **OnRoundOver** ()

5.110.1 Detailed Description

Contract for any class that can perform actions when a round or challenge round finishes.

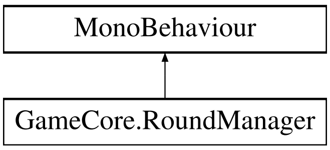
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/GameManager.cs

5.111 GameCore.RoundManager Class Reference

Starts current round and provides functionality to begin next round.

Inheritance diagram for GameCore.RoundManager:



Public Member Functions

- void Begin ()

  Begin this instance.
- void BeginNextRound ()

  Begins the next round.

Public Attributes

- RoundText roundText

  The text shown when a new round starts.
- GameObject [ ] roundPrefabs

  A list of possible round prefabs.

Properties

- int currentRound `[get]`

  Gets the current round.

5.111.1 Detailed Description

Starts current round and provides functionality to begin next round.

5.111.2 Member Function Documentation

5.111.2.1 Begin()

```
void GameCore.RoundManager.Begin ( )
```

Begin this instance.

5.111.2.2 BeginNextRound()

```
void GameCore.RoundManager.BeginNextRound ( )
```

Begins the next round.

5.111.3 Member Data Documentation

5.111.3.1 roundPrefabs

```
GameObject [] GameCore.RoundManager.roundPrefabs
```

A list of possible round prefabs.

5.111.3.2 roundText

```
RoundText GameCore.RoundManager.roundText
```

The text shown when a new round starts.

5.111.4 Property Documentation

5.111.4.1 currentRound

```
int GameCore.RoundManager.currentRound  [get]
```

Gets the current round.

The current round.

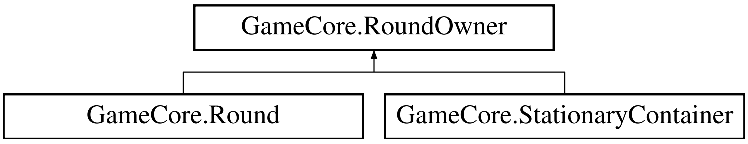
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/RoundManager.cs

5.112 GameCore.RoundOwner Interface Reference

Contract for any entity responsible for tracking enemies in a round.

Inheritance diagram for GameCore.RoundOwner:



Public Member Functions

- void RemoveEnemyFromRound (RoundEnemy enemy)

  Registers enemy removed from round.
- void EnemyEscapedRound (RoundEnemy enemy)

  Registers enemy escaped round.

5.112.1 Detailed Description

Contract for any entity responsible for tracking enemies in a round.

5.112.2 Member Function Documentation

5.112.2.1 EnemyEscapedRound()

```
void GameCore.RoundOwner.EnemyEscapedRound (
            RoundEnemy enemy )
```

Registers enemy escaped round.

Parameters

enemy	Enemy.

Implemented in GameCore.Round, and GameCore.StationaryContainer.

5.112.2.2 RemoveEnemyFromRound()

```
void GameCore.RoundOwner.RemoveEnemyFromRound (
            RoundEnemy enemy )
```

Registers enemy removed from round.

Parameters

enemy	Enemy.

Implemented in GameCore.Round, and GameCore.StationaryContainer.

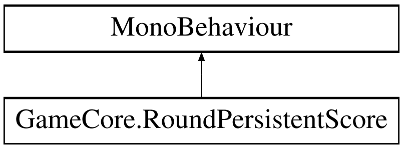
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Round/Round.cs

5.113 GameCore.RoundPersistentScore Class Reference

Persistently stores and retrieves the highest round the player has reached. Data is stored in PlayerPrefs.

Inheritance diagram for GameCore.RoundPersistentScore:



Public Member Functions

- void SetRound (int round)

  Sets round player has reached.

Properties

- int highestRound `[get]`

  Gets the highest round achieved by the player.

5.113.1 Detailed Description

Persistently stores and retrieves the highest round the player has reached. Data is stored in PlayerPrefs.

5.113.2 Member Function Documentation

5.113.2.1 SetRound()

```
void GameCore.RoundPersistentScore.SetRound (
             int round )
```

Sets round player has reached.

Parameters

round	Round player has reached.

5.113.3 Property Documentation

5.113.3.1 highestRound

```
int GameCore.RoundPersistentScore.highestRound  [get]
```

Gets the highest round achieved by the player.

The highest round.

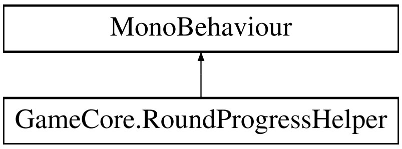
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Score/RoundPersistentScore.cs

5.114 GameCore.RoundProgressHelper Class Reference

Ensures that if an enemy is offscreen for too long a period it is removed from the round.

Inheritance diagram for GameCore.RoundProgressHelper:



Public Attributes

- GameManager gameManager

    The game manager. Used to retrieve current round enemies.

5.114.1 Detailed Description

Ensures that if an enemy is offscreen for too long a period it is removed from the round.

5.114.2 Member Data Documentation

5.114.2.1 gameManager

`GameManager GameCore.RoundProgressHelper.gameManager`

The game manager. Used to retrieve current round enemies.

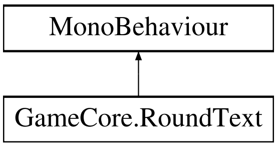
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/RoundProgressHelper.cs

5.115 GameCore.RoundText Class Reference

Updates onscreen text to signify a round start or end.

Inheritance diagram for GameCore.RoundText:



Public Member Functions

- void SetRoundNumber (int roundNumber)

  Sets the round number. Does not show text if not already onscreen.
- void SetWaveCompleteText ()

  Sets text to RoundText::
- void SetChallengeWaveStartText ()

  Sets text to RoundText::CHALLENGE_WAVE_START_TEXT
- void SetBossWaveStartText ()

  Sets text to RoundText::BOSS_WAVE_START_TEXT
- void SetBossCompleteText ()

  Sets text to RoundText::BOSS_WAVE_COMPLETE_TEXT
- void SetChallengeWaveCompleteText ()

  Sets text to RoundText::CHALLENGE_WAVE_COMPLETE_TEXT
- void SetRoundsCompleteText ()

  Sets text to RoundText::ALL_ROUNDS_COMPLETE_TEXT
- void SetGameOver ()

  Sets text to RoundText::GAME_OVER_TEXT
- void CalculatePercentage (float enemiesKilled, float maxEnemies)

  Calculates and shows the total challenge enemies killed as a percentage.
- void ShowForSeconds (float showSeconds, float fadeOutSeconds, Action callbackOnFadeOut=null)

  Shows text onscreen for a number of seconds.
- void WaitForChallengePercentageToBeCalculated (float fadeOutTimeSeconds, Action onComplete)

  Waits for challenge percentage calculation to finish.

Public Attributes

- Image background

  The text background. Phased in and out with text.

5.115.1 Detailed Description

Updates onscreen text to signify a round start or end.

5.115.2 Member Function Documentation

5.115.2.1 CalculatePercentage()

```
void GameCore.RoundText.CalculatePercentage (
             float enemiesKilled,
             float maxEnemies )
```

Calculates and shows the total challenge enemies killed as a percentage.

Parameters

enemiesKilled	Enemies killed.
maxEnemies	Max enemies.

5.115.2.2 SetBossCompleteText()

```
void GameCore.RoundText.SetBossCompleteText ( )
```

Sets text to RoundText::BOSS_WAVE_COMPLETE_TEXT

5.115.2.3 SetBossWaveStartText()

```
void GameCore.RoundText.SetBossWaveStartText ( )
```

Sets text to RoundText::BOSS_WAVE_START_TEXT

5.115.2.4 SetChallengeWaveCompleteText()

```
void GameCore.RoundText.SetChallengeWaveCompleteText ( )
```

Sets text to RoundText::CHALLENGE_WAVE_COMPLETE_TEXT

5.115.2.5 SetChallengeWaveStartText()

```
void GameCore.RoundText.SetChallengeWaveStartText ( )
```

Sets text to RoundText::CHALLENGE_WAVE_START_TEXT

5.115.2.6 SetGameOver()

```
void GameCore.RoundText.SetGameOver ( )
```

Sets text to RoundText::GAME_OVER_TEXT

5.115.2.7 SetRoundNumber()

```
void GameCore.RoundText.SetRoundNumber (
             int roundNumber )
```

Sets the round number. Does not show text if not already onscreen.

Parameters

roundNumber	Round number.

5.115.2.8 SetRoundsCompleteText()

```
void GameCore.RoundText.SetRoundsCompleteText ( )
```

Sets text to RoundText::ALL_ROUNDS_COMPLETE_TEXT

5.115.2.9 SetWaveCompleteText()

```
void GameCore.RoundText.SetWaveCompleteText ( )
```

Sets text to RoundText::

5.115.2.10 ShowForSeconds()

```
void GameCore.RoundText.ShowForSeconds (
             float showSeconds,
             float fadeOutSeconds,
             Action callbackOnFadeOut = null )
```

Shows text onscreen for a number of seconds.

Parameters

showSeconds	Number of seconds to show text.
fadeOutSeconds	Time to fade out text.
callbackOnFadeOut	Callback on fade out.

5.115.2.11 WaitForChallengePercentageToBeCalculated()

```
void GameCore.RoundText.WaitForChallengePercentageToBeCalculated (
            float fadeOutTimeSeconds,
            Action onComplete )
```

Waits for challenge percentage calculation to finish.

Parameters

fadeOutTimeSeconds	Fade out time seconds.
onComplete	On complete.

5.115.3 Member Data Documentation

5.115.3.1 background

```
Image GameCore.RoundText.background
```

The text background. Phased in and out with text.

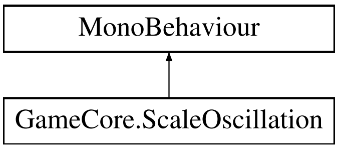
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/RoundText.cs

5.116 GameCore.ScaleOscillation Class Reference

Lerps between min and max scale over time.

Inheritance diagram for GameCore.ScaleOscillation:



Public Attributes

- float minScale = 0.7f

  The minimum scale.
- float maxScale = 1.3f

  The maximum scale.
- float scaleSpeed = 5f

  The scale speed.
- float scaleDecreaseOnHit

  The amount to decrease scale on hit.

5.116.1 Detailed Description

Lerps between min and max scale over time.

5.116.2 Member Data Documentation

5.116.2.1 maxScale

```
float GameCore.ScaleOscillation.maxScale = 1.3f
```

The maximum scale.

5.116.2.2 minScale

```
float GameCore.ScaleOscillation.minScale = 0.7f
```

The minimum scale.

5.116.2.3 scaleDecreaseOnHit

```
float GameCore.ScaleOscillation.scaleDecreaseOnHit
```

The amount to decrease scale on hit.

5.116.2.4 scaleSpeed

```
float GameCore.ScaleOscillation.scaleSpeed = 5f
```

The scale speed.

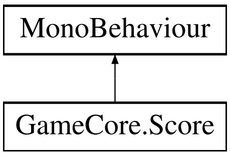
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Effectors/ScaleOscillation.cs

5.117 GameCore.Score Class Reference

Handles players score (points) for a specific run. Has functionality to add (when points are collected) and remove (when player purchases items at the shop) points. Updates the score UI incrementally.

Inheritance diagram for GameCore.Score:



Public Member Functions

- void AddScore (int score)

  Adds to players score.
- void RemoveScore (int score)

  Removes from players score.

Properties

- int score `[get, set]`

  Gets or sets the players current score.

5.117.1 Detailed Description

Handles players score (points) for a specific run. Has functionality to add (when points are collected) and remove (when player purchases items at the shop) points. Updates the score UI incrementally.

5.117.2 Member Function Documentation

5.117.2.1 AddScore()

```
void GameCore.Score.AddScore (
             int score )
```

Adds to players score.

Parameters

score	Score.

5.117.2.2 RemoveScore()

```
void GameCore.Score.RemoveScore (
             int score )
```

Removes from players score.

Parameters

score	Score.

5.117.3 Property Documentation

5.117.3.1 score

```
int GameCore.Score.score  [get], [set]
```

Gets or sets the players current score.

The score.

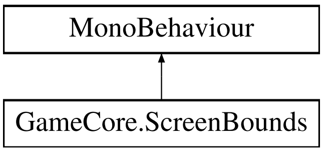
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Score/Score.cs

5.118 GameCore.ScreenBounds Class Reference

Holds data about the screen bounds.

Inheritance diagram for GameCore.ScreenBounds:



Public Member Functions

- Vector2 GetHorizontalBounds ()

  Gets the horizontal bounds. X = left, Y = right.
- Vector2 GetHorizontalViewportBounds ()

  Gets the horizontal viewport bounds. X = left, Y = right.
- Vector2 GetVerticalViewportBounds ()

  Gets the vertical viewport bounds. X = bottom, Y = top.
- bool IsWithinBounds (Vector2 viewportPos)

  Determines whether the specified viewportPos is within the screen bounds.

Public Attributes

- float lowerVerticalBounds = 0.2f

  The lower vertical bounds. Enemies below this will cause damage.

5.118.1 Detailed Description

Holds data about the screen bounds.

5.118.2 Member Function Documentation

5.118.2.1 GetHorizontalBounds()

```
Vector2 GameCore.ScreenBounds.GetHorizontalBounds ( )
```

Gets the horizontal bounds. X = left, Y = right.

Returns

The horizontal bounds.

5.118.2.2 GetHorizontalViewportBounds()

```
Vector2 GameCore.ScreenBounds.GetHorizontalViewportBounds ( )
```

Gets the horizontal viewport bounds. X = left, Y = right.

Returns

The horizontal viewport bounds.

5.118.2.3 GetVerticalViewportBounds()

```
Vector2 GameCore.ScreenBounds.GetVerticalViewportBounds ( )
```

Gets the vertical viewport bounds. X = bottom, Y = top.

Returns

The vertical viewport bounds.

5.118.2.4 IsWithinBounds()

```
bool GameCore.ScreenBounds.IsWithinBounds (
             Vector2 viewportPos )
```

Determines whether the specified viewportPos is within the screen bounds.

Returns

`true` if the specified viewportPos is within bounds; otherwise, `false`.

Parameters

viewportPos	Viewport position.

5.118.3 Member Data Documentation

5.118.3.1 lowerVerticalBounds

```
float GameCore.ScreenBounds.lowerVerticalBounds = 0.2f
```

The lower vertical bounds. Enemies below this will cause damage.

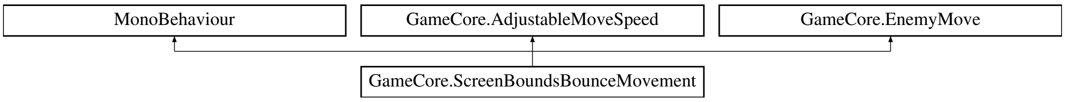
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/ScreenBounds.cs

5.119 GameCore.ScreenBoundsBounceMovement Class Reference

Controls enemies that bouce around the screen.

Inheritance diagram for GameCore.ScreenBoundsBounceMovement:


MonoBehaviour
GameCore.AdjustableMoveSpeed
GameCore.EnemyMove
GameCore.ScreenBoundsBounceMovement

Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in.
- void Pause ()

  Pause this instance if not continueMovementOnPlayerDeath.
- void Resume ()

  Resume this instance.
- void **IncrementSpeed** ()

Public Attributes

- LayerMask hitMask

  The layermask for the screen bounds.
- float moveSpeed = 2f

  The movement speed.
- float moveSpeedIncrement = 1

  The amount to increment the movement speed near round end.
- bool continueMovementOnPlayerDeath = false

  Sets whether this instance should continue moving while player is respawning.

5.119.1 Detailed Description

Controls enemies that bouce around the screen.

5.119.2 Member Function Documentation

5.119.2.1 Begin()

```
void GameCore.ScreenBoundsBounceMovement.Begin ( )
```

Begin this instance. Starts fade in.

Implements GameCore.EnemyMove.

5.119.2.2 Pause()

```
void GameCore.ScreenBoundsBounceMovement.Pause ( )
```

Pause this instance if not continueMovementOnPlayerDeath.

Implements GameCore.EnemyMove.

5.119.2.3 Resume()

```
void GameCore.ScreenBoundsBounceMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.119.3 Member Data Documentation

5.119.3.1 continueMovementOnPlayerDeath

```
bool GameCore.ScreenBoundsBounceMovement.continueMovementOnPlayerDeath = false
```

Sets whether this instance should continue moving while player is respawning.

5.119.3.2 hitMask

```
LayerMask GameCore.ScreenBoundsBounceMovement.hitMask
```

The layermask for the screen bounds.

5.119.3.3 moveSpeed

```
float GameCore.ScreenBoundsBounceMovement.moveSpeed = 2f
```

The movement speed.

5.119.3.4 moveSpeedIncrement

```
float GameCore.ScreenBoundsBounceMovement.moveSpeedIncrement = 1
```

The amount to increment the movement speed near round end.

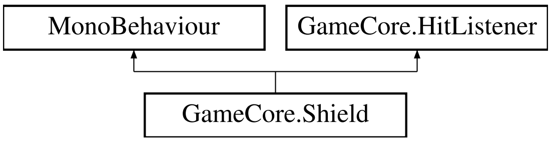
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/ScreenBoundsBounceMovement.cs

5.120 GameCore.Shield Class Reference

Provides shield functionality for player (when purchased through the in-game store).

Inheritance diagram for GameCore.Shield:



Public Member Functions

- void OnHit (int damage)

  Removes the shield from the game. Spawns particle explosion.

Public Attributes

- int damage = 10

  The amount of damage given when an enemy hits the shield.
- float orbitSpeed = 40f

  The speed at which the shield orbits the player.
- Color colorExplosion

  The color of the particles released when the shield is destroyed.
- int numOfParticlesOnDeath = 80

  The number of particles released on death.

5.120.1 Detailed Description

Provides shield functionality for player (when purchased through the in-game store).

5.120.2 Member Function Documentation

5.120.2.1 OnHit()

```
void GameCore.Shield.OnHit (
             int damage )
```

Removes the shield from the game. Spawns particle explosion.

Parameters

damage	Damage taken. As the shield can only take one hit, this is ignored.

Implements GameCore.HitListener.

5.120.3 Member Data Documentation

5.120.3.1 colorExplosion

```
Color GameCore.Shield.colorExplosion
```

The color of the particles released when the shield is destroyed.

5.120.3.2 damage

```
int GameCore.Shield.damage = 10
```

The amount of damage given when an enemy hits the shield.

5.120.3.3 numOfParticlesOnDeath

```
int GameCore.Shield.numOfParticlesOnDeath = 80
```

The number of particles released on death.

5.120.3.4 orbitSpeed

```
float GameCore.Shield.orbitSpeed = 40f
```

The speed at which the shield orbits the player.

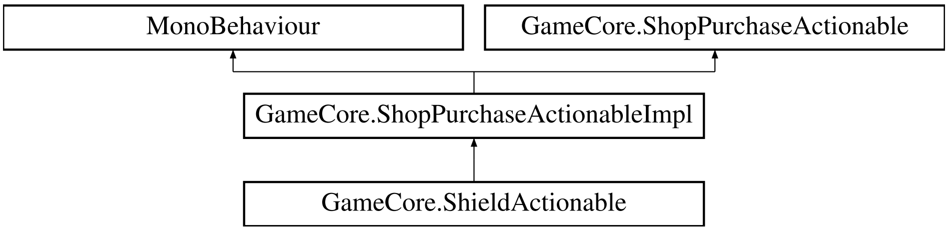
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/Shield.cs

5.121 GameCore.ShieldActionable Class Reference

Provides a player with a shield (or additional shield) when purchased. A player can have up to four active shields. When a shield is destroyed the player can purchase the item again (at an increased cost).

Inheritance diagram for GameCore.ShieldActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ShieldActionable

Public Member Functions

- override bool IsActionable ()

  Determines whether this instance is actionable.
- override void DoAction ()

  Performs the action.

Public Attributes

- GameObject shield

  The shield to enable.

Additional Inherited Members

5.121.1 Detailed Description

Provides a player with a shield (or additional shield) when purchased. A player can have up to four active shields. When a shield is destroyed the player can purchase the item again (at an increased cost).

5.121.2 Member Function Documentation

5.121.2.1 DoAction()

```
override void GameCore.ShieldActionable.DoAction ( )  [virtual]
```

Performs the action.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.121.2.2 IsActionable()

```
override bool GameCore.ShieldActionable.IsActionable ( ) [virtual]
```

Determines whether this instance is actionable.

Returns

true if shield not already active.

```
false
```

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.121.3 Member Data Documentation

5.121.3.1 shield

```
GameObject GameCore.ShieldActionable.shield
```

The shield to enable.

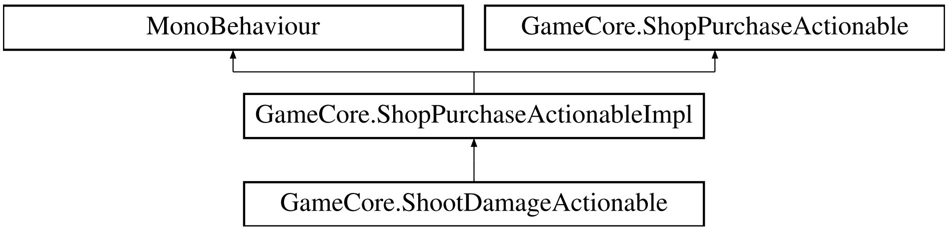
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ShieldActionable.cs

5.122 GameCore.ShootDamageActionable Class Reference

Increases the damage of players projectiles by one when purchased.

Inheritance diagram for GameCore.ShootDamageActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ShootDamageActionable

Public Member Functions

- override void DoAction ()
  Increments player damage by calling PlayerShot::IncrementDamage.

Public Attributes

- PlayerShoot playerShoot

  The player shoot instance.

Additional Inherited Members

5.122.1 Detailed Description

Increases the damage of players projectiles by one when purchased.

5.122.2 Member Function Documentation

5.122.2.1 DoAction()

```
override void GameCore.ShootDamageActionable.DoAction ( )  [virtual]
```

Increments player damage by calling PlayerShot::IncrementDamage.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.122.3 Member Data Documentation

5.122.3.1 playerShoot

```
PlayerShoot GameCore.ShootDamageActionable.playerShoot
```

The player shoot instance.

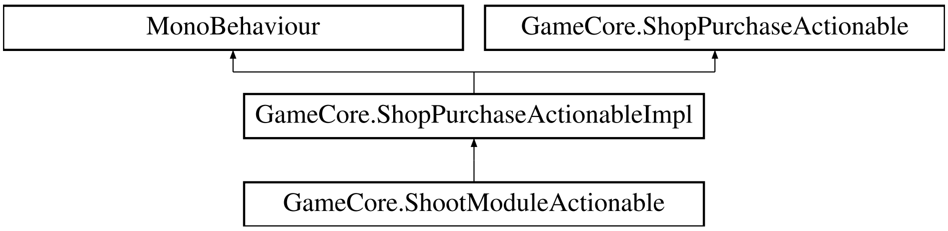
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ShootDamageActionable.cs

5.123 GameCore.ShootModuleActionable Class Reference

Adds new shoot modules, followers, or shields when purchased.

Inheritance diagram for GameCore.ShootModuleActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ShootModuleActionable

Public Member Functions

- override bool IsActionable ()

  Determines whether this instance is actionable.
- override void DoAction ()

  Enables new module.
- override void CheckActionable ()

  Checks if this instance is actionable/purchasable. Returns true if not all instances have been purchased.

Public Attributes

- string objectName

  Tag name of object that contains the PlayerShootModules to be actioned.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

5.123.1 Detailed Description

Adds new shoot modules, followers, or shields when purchased.

5.123.2 Member Function Documentation

5.123.2.1 CheckActionable()

```
override void GameCore.ShootModuleActionable.CheckActionable ( )  [virtual]
```

Checks if this instance is actionable/purchasable. Returns true if not all instances have been purchased.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.123.2.2 DoAction()

```
override void GameCore.ShootModuleActionable.DoAction ( )  [virtual]
```

Enables new module.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.123.2.3 IsActionable()

```
override bool GameCore.ShootModuleActionable.IsActionable ( ) [virtual]
```

Determines whether this instance is actionable.

Returns

true if PlayerShootModules::IsActionable

```
false
```

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.123.3 Member Data Documentation

5.123.3.1 objectName

```
string GameCore.ShootModuleActionable.objectName
```

Tag name of object that contains the PlayerShootModules to be actioned.

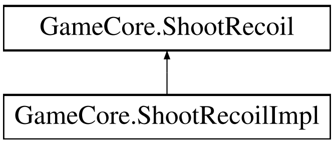
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ShootModuleActionable.cs

5.124 GameCore.ShootRecoil Interface Reference

Contract for any GameObject that can provide recoil.

Inheritance diagram for GameCore.ShootRecoil:



Public Member Functions

- void Execute ()

  Execute recoil.

5.124.1 Detailed Description

Contract for any GameObject that can provide recoil.

5.124.2 Member Function Documentation

5.124.2.1 Execute()

```
void GameCore.ShootRecoil.Execute ( )
```

Execute recoil.

Implemented in GameCore.ShootRecoilImpl.

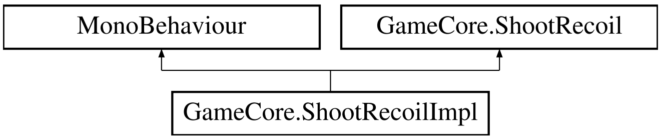
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Player/ShootRecoilImpl.cs

5.125 GameCore.ShootRecoilImpl Class Reference

Provides weapon recoil functionality. Where the gun is temporarily moved back by the force of a shot.

Inheritance diagram for GameCore.ShootRecoilImpl:



Public Member Functions

- void Execute ()

  Execute recoil.

Public Attributes

- Transform weapon

  The weapon.
- float Recoil = 1f

  Displacement force.
- float returnSpeed = 1f

  The speed at which the weapon returns to its original position.

5.125.1 Detailed Description

Provides weapon recoil functionality. Where the gun is temporarily moved back by the force of a shot.

5.125.2 Member Function Documentation

5.125.2.1 Execute()

```
void GameCore.ShootRecoilImpl.Execute ( )
```

Execute recoil.

Implements GameCore.ShootRecoil.

5.125.3 Member Data Documentation

5.125.3.1 Recoil

```
float GameCore.ShootRecoilImpl.Recoil = 1f
```

Displacement force.

5.125.3.2 returnSpeed

```
float GameCore.ShootRecoilImpl.returnSpeed = 1f
```

The speed at which the weapon returns to its original position.

5.125.3.3 weapon

```
Transform GameCore.ShootRecoilImpl.weapon
```

The weapon.

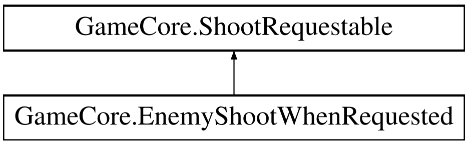
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/ShootRecoilImpl.cs

5.126 GameCore.ShootRequestable Interface Reference

Contract for any entity that can shoot projectiles.

Inheritance diagram for GameCore.ShootRequestable:



Public Member Functions

- void **RequestShoot** ()

5.126.1 Detailed Description

Contract for any entity that can shoot projectiles.

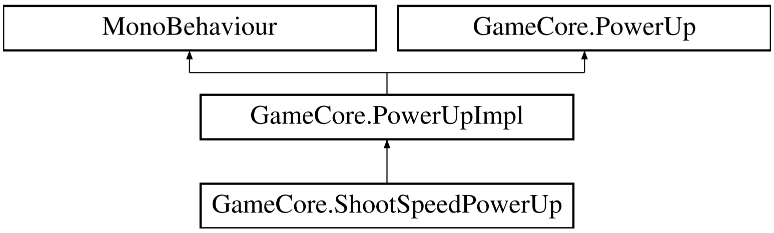
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/EnemyShootWhenRequested.cs

5.127 GameCore.ShootSpeedPowerUp Class Reference

Increases players shooting speed temporaily when picked up.

Inheritance diagram for GameCore.ShootSpeedPowerUp:



Public Member Functions

- override void Perform (Transform player)

  Perform the specified powerup action. Finds main player shoot module and invokes PlayerShoot::DecrementSec↵ BetweenShotsForSeconds.

Public Attributes

- float secsBetweenShotDecrement = 0.4f

  The seconds decrement between players shots.
- float secSpeedIncrease = 3f

  The amount of time shot speed is increased.

Additional Inherited Members

5.127.1 Detailed Description

Increases players shooting speed temporaily when picked up.

5.127.2 Member Function Documentation

5.127.2.1 Perform()

```
override void GameCore.ShootSpeedPowerUp.Perform (
             Transform player )  [virtual]
```

Perform the specified powerup action. Finds main player shoot module and invokes PlayerShoot::DecrementSecBetweenShotsForSeconds.

Parameters

player	Player tranform.

Implements GameCore.PowerUpImpl.

5.127.3 Member Data Documentation

5.127.3.1 secsBetweenShotDecrement

```
float GameCore.ShootSpeedPowerUp.secsBetweenShotDecrement = 0.4f
```

The seconds decrement between players shots.

5.127.3.2 secSpeedIncrease

```
float GameCore.ShootSpeedPowerUp.secSpeedIncrease = 3f
```

The amount of time shot speed is increased.

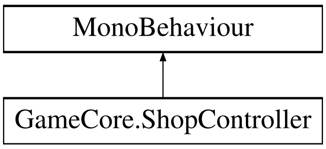
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Powerups/ShootSpeedPowerUp.cs

5.128 GameCore.ShopController Class Reference

Responsible for opening and closing shop, and updating whether items can be purchased.

Inheritance diagram for GameCore.ShopController:



Public Member Functions

- void OpenShop ()

  Opens the shop and updates purchases.
- void CloseShop ()

  Closes the shop and plays audioclip.

Public Attributes

- AudioClip audioOnShopClose

  The audio to play on shop close.
- Button closeShopButton

  THe button responsible for closing the shop. This button is disabled whilst the shop opening animation plays.
- ButtonAnimationController animationController

  The animation controller responsible for animating the shop buttons.

5.128.1 Detailed Description

Responsible for opening and closing shop, and updating whether items can be purchased.

5.128.2 Member Function Documentation

5.128.2.1 CloseShop()

```
void GameCore.ShopController.CloseShop ( )
```

Closes the shop and plays audioclip.

5.128.2.2 OpenShop()

```
void GameCore.ShopController.OpenShop ( )
```

Opens the shop and updates purchases.

5.128.3 Member Data Documentation

5.128.3.1 animationController

```
ButtonAnimationController GameCore.ShopController.animationController
```

The animation controller responsible for animating the shop buttons.

5.128.3.2 audioOnShopClose

```
AudioClip GameCore.ShopController.audioOnShopClose
```

The audio to play on shop close.

5.128.3.3 closeShopButton

```
Button GameCore.ShopController.closeShopButton
```

THe button responsible for closing the shop. This button is disabled whilst the shop opening animation plays.

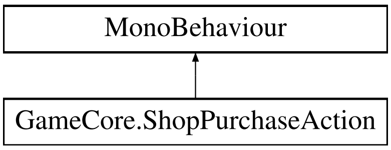
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ShopController.cs

5.129 GameCore.ShopPurchaseAction Class Reference

Actions any shop purchase requests.

Inheritance diagram for GameCore.ShopPurchaseAction:



Public Member Functions

- void DoAction (ShopPurchaseActionableImpl action)

  Performs action if actionable.

Public Attributes

- Action OnPuchase

  Invoked when a shop purchase action is performed.
- AudioClip audioOnPurchase

  The audio clip to play on purchase action.

5.129.1 Detailed Description

Actions any shop purchase requests.

5.129.2 Member Function Documentation

5.129.2.1 DoAction()

```
void GameCore.ShopPurchaseAction.DoAction (
             ShopPurchaseActionableImpl action )
```

Performs action if actionable.

Parameters

action	Action to perform.

5.129.3 Member Data Documentation

5.129.3.1 audioOnPurchase

```
AudioClip GameCore.ShopPurchaseAction.audioOnPurchase
```

The audio clip to play on purchase action.

5.129.3.2 OnPuchase

```
Action GameCore.ShopPurchaseAction.OnPuchase
```

Invoked when a shop purchase action is performed.

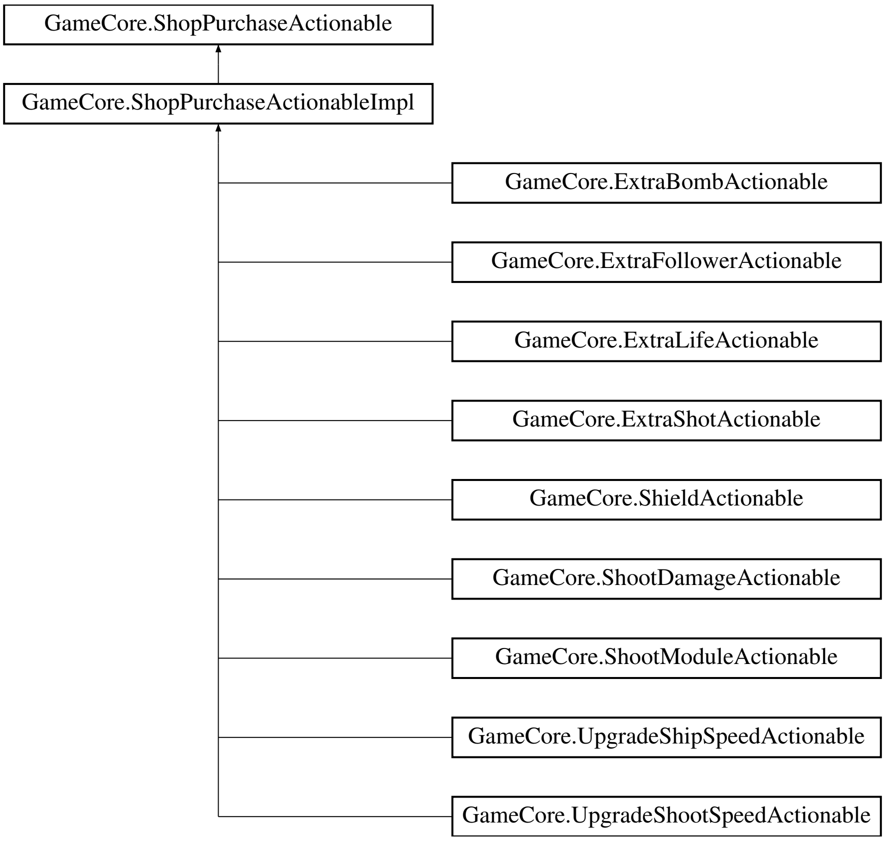
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ShopPurchaseAction.cs

5.130 GameCore.ShopPurchaseActionable Interface Reference

Contract for any item that can be purchased in the shop. Provides methods for performing action, checking if all actions have been performed, and checking if action can be performed.

Inheritance diagram for GameCore.ShopPurchaseActionable:


GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ExtraBombActionable
GameCore.ExtraFollowerActionable
GameCore.ExtraLifeActionable
GameCore.ExtraShotActionable
GameCore.ShieldActionable
GameCore.ShootDamageActionable
GameCore.ShootModuleActionable
GameCore.UpgradeShipSpeedActionable
GameCore.UpgradeShootSpeedActionable

Public Member Functions

- bool IsActionable ()

  Determines whether this instance is actionable.
- void DoAction ()

  Performs the action.
- void CheckActionable ()

  Checks if this instance is actionable/purchasable.
- void CheckComplete ()

  Checks if all items have been purchased.

5.130.1 Detailed Description

Contract for any item that can be purchased in the shop. Provides methods for performing action, checking if all actions have been performed, and checking if action can be performed.

5.130.2 Member Function Documentation

5.130.2.1 CheckActionable()

```
void GameCore.ShopPurchaseActionable.CheckActionable ( )
```

Checks if this instance is actionable/purchasable.

Implemented in GameCore.ShopPurchaseActionableImpl, and GameCore.ShootModuleActionable.

5.130.2.2 CheckComplete()

```
void GameCore.ShopPurchaseActionable.CheckComplete ( )
```

Checks if all items have been purchased.

Implemented in GameCore.ShopPurchaseActionableImpl.

5.130.2.3 DoAction()

```
void GameCore.ShopPurchaseActionable.DoAction ( )
```

Performs the action.

Implemented in GameCore.ShopPurchaseActionableImpl, GameCore.ShootModuleActionable, GameCore.ShieldActionable, GameCore.UpgradeShipSpeedActionable, GameCore.UpgradeShootSpeedActionable, GameCore.ExtraBombActionable, GameCore.ExtraLifeActionable, GameCore.ShootDamageActionable, and GameCore.ExtraShotActionable.

5.130.2.4 IsActionable()

```
bool GameCore.ShopPurchaseActionable.IsActionable ( )
```

Determines whether this instance is actionable.

Returns

`true` if this instance is actionable; otherwise, `false`.

Implemented in GameCore.ShopPurchaseActionableImpl, GameCore.ShootModuleActionable, and GameCore.ShieldActionable.

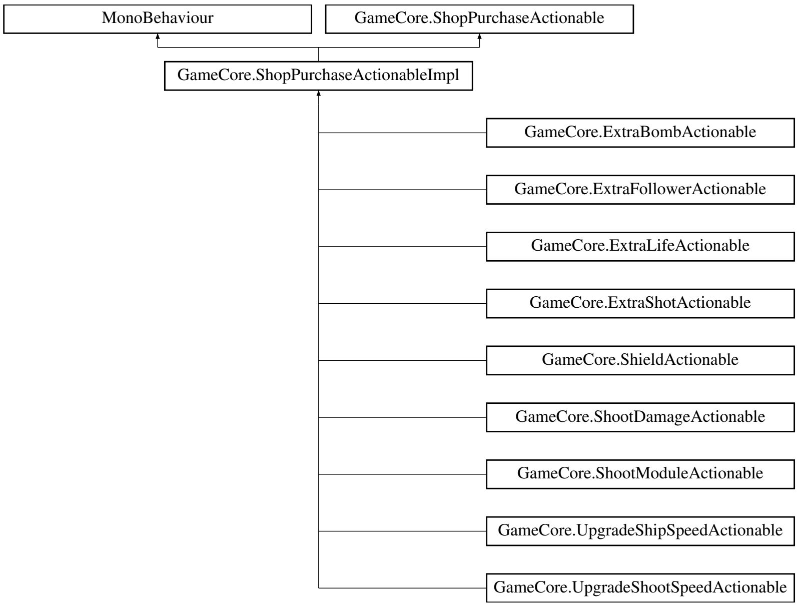
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Shop/ShopPurchasableActionable.cs

5.131 GameCore.ShopPurchaseActionableImpl Class Reference

Base class for any shop purchase items. Provides access to and manipulation of all common shop purchase features, including: PointsImages, foreground overlay (enabled when item not currently purchasable), the text that displays the item cost, and the cost value.

Inheritance diagram for GameCore.ShopPurchaseActionableImpl:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.ExtraBombActionable
GameCore.ExtraFollowerActionable
GameCore.ExtraLifeActionable
GameCore.ExtraShotActionable
GameCore.ShieldActionable
GameCore.ShootDamageActionable
GameCore.ShootModuleActionable
GameCore.UpgradeShipSpeedActionable
GameCore.UpgradeShootSpeedActionable

Public Member Functions

- virtual void CheckActionable ()

  Checks if this instance is actionable/purchasable. If not actionable, an image is overlayed.
- void CheckComplete ()

  Checks if all items have been purchased. If true, the cost text is set to '-'
- virtual bool IsActionable ()

  Determines whether this instance is actionable. True if the player can afford to purchase and it is still has instances remaining.
- virtual void DoAction ()

  Performs the action.

Public Attributes

- GameObject foreground

  Image used to overlay on a shop item when it can't be purchased.
- PointsImages pointsImages

  The images used to show how many instances of this item have been purchased.
- Text pointsText

  The text showing the cost of the item.
- int cost = 10

  The cost of purchasing the first instance of this item. Each subsequent purchase doubles in cost.

Protected Member Functions

- virtual void **Awake** ()

Protected Attributes

- int **m_MaxUses**

5.131.1 Detailed Description

Base class for any shop purchase items. Provides access to and manipulation of all common shop purchase features, including: PointsImages, foreground overlay (enabled when item not currently purchasable), the text that displays the item cost, and the cost value.

5.131.2 Member Function Documentation

5.131.2.1 CheckActionable()

```
virtual void GameCore.ShopPurchaseActionableImpl.CheckActionable ( )  [virtual]
```

Checks if this instance is actionable/purchasable. If not actionable, an image is overlayed.

Implements GameCore.ShopPurchaseActionable.

Reimplemented in GameCore.ShootModuleActionable.

5.131.2.2 CheckComplete()

```
void GameCore.ShopPurchaseActionableImpl.CheckComplete ( )
```

Checks if all items have been purchased. If true, the cost text is set to '-'

Implements GameCore.ShopPurchaseActionable.

5.131.2.3 DoAction()

```
virtual void GameCore.ShopPurchaseActionableImpl.DoAction ( )  [virtual]
```

Performs the action.

Implements GameCore.ShopPurchaseActionable.

Reimplemented in GameCore.ShootModuleActionable, GameCore.ShieldActionable, GameCore.UpgradeShipSpeedActionable, GameCore.UpgradeShootSpeedActionable, GameCore.ExtraBombActionable, GameCore.ExtraLifeActionable, GameCore.ShootDamageActionable, and GameCore.ExtraShotActionable.

5.131.2.4 IsActionable()

```
virtual bool GameCore.ShopPurchaseActionableImpl.IsActionable ( )  [virtual]
```

Determines whether this instance is actionable. True if the player can afford to purchase and it is still has instances remaining.

Returns

true

```
false
```

Implements GameCore.ShopPurchaseActionable.

Reimplemented in GameCore.ShootModuleActionable, and GameCore.ShieldActionable.

5.131.3 Member Data Documentation

5.131.3.1 cost

```
int GameCore.ShopPurchaseActionableImpl.cost = 10
```

The cost of purchasing the first instance of this item. Each subsequent purchase doubles in cost.

5.131.3.2 foreground

```
GameObject GameCore.ShopPurchaseActionableImpl.foreground
```

Image used to overlay on a shop item when it can't be purchased.

5.131.3.3 pointsImages

```
PointsImages GameCore.ShopPurchaseActionableImpl.pointsImages
```

The images used to show how many instances of this item have been purchased.

5.131.3.4 pointsText

```
Text GameCore.ShopPurchaseActionableImpl.pointsText
```

The text showing the cost of the item.

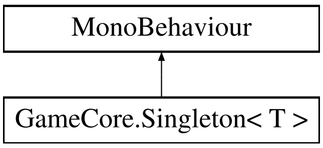
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/ShopPurchasableActionable.cs

5.132 GameCore.Singleton< T > Class Template Reference

Generic singleton base class.

Inheritance diagram for GameCore.Singleton< T >:



Properties

- static T Instance `[get]`

  returns instance of T.
- static bool HasInstance `[get]`

  returns true if instance has not been destroyed.
- static bool IsDestroyed `[get]`

  returns true if instance is not null.

5.132.1 Detailed Description

Generic singleton base class.

Template Parameters

T	

Type Constraints

T* : *MonoBehaviour

5.132.2 Property Documentation

5.132.2.1 HasInstance

```
bool GameCore.Singleton< T >.HasInstance  [static], [get]
```

returns true if instance has not been destroyed.

5.132.2.2 Instance

```
T GameCore.Singleton< T >.Instance  [static], [get]
```

returns instance of T.

5.132.2.3 IsDestroyed

```
bool GameCore.Singleton< T >.IsDestroyed  [static], [get]
```

returns true if instance is not null.

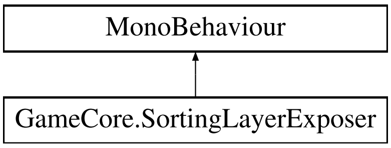
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Utilities/Singleton.cs

5.133 GameCore.SortingLayerExposer Class Reference

Exposes sorting layer of MeshRenderer.

Inheritance diagram for GameCore.SortingLayerExposer:



Public Attributes

- string sortingLayerName = "Default"

  The name of the layer to set.
- int sortingOrder = 0

  The sorting order to set.

5.133.1 Detailed Description

Exposes sorting layer of MeshRenderer.

5.133.2 Member Data Documentation

5.133.2.1 sortingLayerName

```
string GameCore.SortingLayerExposer.sortingLayerName = "Default"
```

The name of the layer to set.

5.133.2.2 sortingOrder

```
int GameCore.SortingLayerExposer.sortingOrder = 0
```

The sorting order to set.

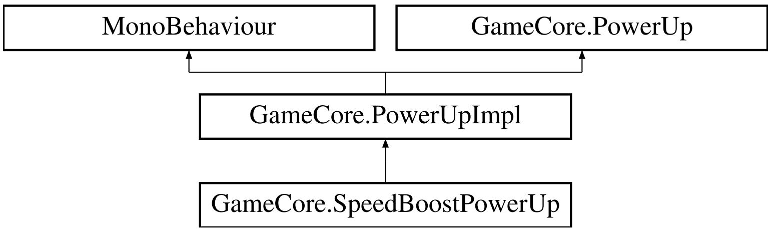
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Utilities/SortingLayerExposer.cs

5.134 GameCore.SpeedBoostPowerUp Class Reference

Increases players movement speed temporarily.

Inheritance diagram for GameCore.SpeedBoostPowerUp:



Public Member Functions

- override void Perform (Transform player)

  Perform the specified powerup action. Invokes PlayerController::IncrementSpeedForSeconds

Public Attributes

- float speedIncrease = 5f

  The amount to increase movement speed.
- float secSpeedIncrease = 3f

  How long the players movement speed is increased.

Additional Inherited Members

5.134.1 Detailed Description

Increases players movement speed temporarily.

5.134.2 Member Function Documentation

5.134.2.1 Perform()

```
override void GameCore.SpeedBoostPowerUp.Perform (
             Transform player )  [virtual]
```

Perform the specified powerup action. Invokes PlayerController::IncrementSpeedForSeconds

Parameters

player	Player tranform.

Implements GameCore.PowerUpImpl.

5.134.3 Member Data Documentation

5.134.3.1 secSpeedIncrease

```
float GameCore.SpeedBoostPowerUp.secSpeedIncrease = 3f
```

How long the players movement speed is increased.

5.134.3.2 speedIncrease

```
float GameCore.SpeedBoostPowerUp.speedIncrease = 5f
```

The amount to increase movement speed.

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

- Pew Pew/Scripts/Powerups/SpeedBoostPowerUp.cs

5.135 WarpGrid.Spring Struct Reference

Connects two PointMass on a grid.

Public Member Functions

- Spring (PointMass end1, PointMass end2, float stiffness, float damping)

  Initializes a new instance of the Spring struct.
- void Update ()

  Applies a pulling force to each attached point.

Public Attributes

- PointMass End1

  PointMass 1.
- PointMass End2

  PointMass 2.
- float TargetLength

  The points will move to be within this range of each other.
- float Stiffness

  Signifies how easy it is for the springs to be pulled apart.
- float Damping

  Provides a dampening effect on the movement of the connected point masses.

5.135.1 Detailed Description

Connects two PointMass on a grid.

5.135.2 Constructor & Destructor Documentation

5.135.2.1 Spring()

```
WarpGrid.Spring.Spring (
            PointMass end1,
            PointMass end2,
            float stiffness,
            float damping )
```

Initializes a new instance of the Spring struct.

Parameters

end1	First point.
end2	Second point.
stiffness	Stiffness.
damping	Damping.

5.135.3 Member Function Documentation

5.135.3.1 Update()

```
void WarpGrid.Spring.Update ( )
```

Applies a pulling force to each attached point.

5.135.4 Member Data Documentation

5.135.4.1 Damping

```
float WarpGrid.Spring.Damping
```

Provides a dampening effect on the movement of the connected point masses.

5.135.4.2 End1

```
PointMass WarpGrid.Spring.End1
```

PointMass 1.

5.135.4.3 End2

```
PointMass WarpGrid.Spring.End2
```

PointMass 2.

5.135.4.4 Stiffness

```
float WarpGrid.Spring.Stiffness
```

Signifies how easy it is for the springs to be pulled apart.

5.135.4.5 TargetLength

```
float WarpGrid.Spring.TargetLength
```

The points will move to be within this range of each other.

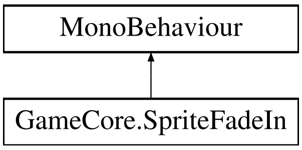
The documentation for this struct was generated from the following file:

- UG/Scripts/Spring.cs

5.136 GameCore.SpriteFadeIn Class Reference

Lerps a sprites alpha from 0 to 1 over a set time defined by GameManager::ROUND_BEGIN_TIME.

Inheritance diagram for GameCore.SpriteFadeIn:



Public Member Functions

- void StartFadeIn (float maxAlpha=1f)
  Starts the fade in.

Properties

- bool finished [get]
  Gets a value indicating whether this SpriteFadeIn has finished lerping alpha.

5.136.1 Detailed Description

Lerps a sprites alpha from 0 to 1 over a set time defined by GameManager::ROUND_BEGIN_TIME.

5.136.2 Member Function Documentation

5.136.2.1 StartFadeIn()

```
void GameCore.SpriteFadeIn.StartFadeIn (
            float maxAlpha = 1f )
```

Starts the fade in.

Parameters

maxAlpha	The maximum alpha.

5.136.3 Property Documentation

5.136.3.1 finished

```
bool GameCore.SpriteFadeIn.finished  [get]
```

Gets a value indicating whether this SpriteFadeIn has finished lerping alpha.

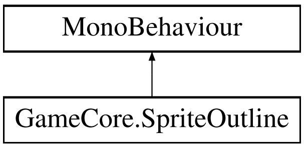
`true` if finished; otherwise, `false`.

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

- Pew Pew/Scripts/Enemies/SpriteFadeIn.cs

5.137 GameCore.SpriteOutline Class Reference

Inheritance diagram for GameCore.SpriteOutline:



Public Attributes

- Color **color** = Color.white
- int **outlineSize** = 1

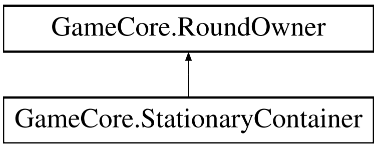
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Sprites/SpriteOutline.cs

5.138 GameCore.StationaryContainer Class Reference

Holds all stationary enemies within a round.

Inheritance diagram for GameCore.StationaryContainer:



Public Member Functions

- void AddEnemy (RoundEnemy e)

  Adds an enemy to structure.
- void RemoveEnemyFromRound (RoundEnemy enemy)

  Removes stationary enemy from round.
- void EnemyEscapedRound (RoundEnemy enemy)

  Removes stationary enemy from round.
- int GetEnemyCount ()

  The current number of stationary enemies in the round.
- List< EnemyHealth > GetAliveEnemies ()

  Gets the alive enemies in the round.

5.138.1 Detailed Description

Holds all stationary enemies within a round.

5.138.2 Member Function Documentation

5.138.2.1 AddEnemy()

```
void GameCore.StationaryContainer.AddEnemy (
             RoundEnemy e )
```

Adds an enemy to structure.

Parameters

e	Enemy to add.

5.138.2.2 EnemyEscapedRound()

```
void GameCore.StationaryContainer.EnemyEscapedRound (
            RoundEnemy enemy )
```

Removes stationary enemy from round.

Parameters

enemy	Enemy to remove.

Implements GameCore.RoundOwner.

5.138.2.3 GetAliveEnemies()

```
List<EnemyHealth> GameCore.StationaryContainer.GetAliveEnemies ( )
```

Gets the alive enemies in the round.

Returns

The alive enemies.

5.138.2.4 GetEnemyCount()

```
int GameCore.StationaryContainer.GetEnemyCount ( )
```

The current number of stationary enemies in the round.

Returns

The enemy count.

5.138.2.5 RemoveEnemyFromRound()

```
void GameCore.StationaryContainer.RemoveEnemyFromRound (
            RoundEnemy enemy )
```

Removes stationary enemy from round.

Parameters

enemy	Enemy to remove.

Implements GameCore.RoundOwner.

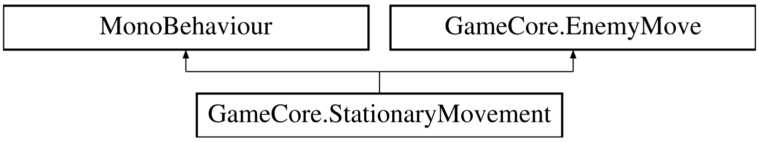
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Round/Round.cs

5.139 GameCore.StationaryMovement Class Reference

Controls stationary enemies fade in and collider enabled status.

Inheritance diagram for GameCore.StationaryMovement:



Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in. Enables collider when fade in complete.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.

5.139.1 Detailed Description

Controls stationary enemies fade in and collider enabled status.

5.139.2 Member Function Documentation

5.139.2.1 Begin()

```
void GameCore.StationaryMovement.Begin ( )
```

Begin this instance. Starts fade in. Enables collider when fade in complete.

Implements GameCore.EnemyMove.

5.139.2.2 Pause()

```
void GameCore.StationaryMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.139.2.3 Resume()

```
void GameCore.StationaryMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

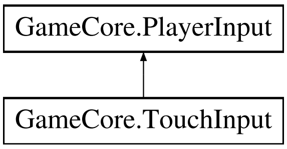
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/StationaryMovement.cs

5.140 GameCore.TouchInput Class Reference

Provides a method to control the player based on mobile touch input.

Inheritance diagram for GameCore.TouchInput:



Public Member Functions

- TouchInput (Transform player, float moveSpeed)

  Initializes a new instance of the TouchInput class.
- Vector2 GetVelocity ()

  Gets the velocity. The players next move.
- float GetMovementSpeed ()

  Gets the players movement speed.
- void SetMovementSpeed (float amount)

  Sets the players movement speed.

5.140.1 Detailed Description

Provides a method to control the player based on mobile touch input.

5.140.2 Constructor & Destructor Documentation

5.140.2.1 TouchInput()

```
GameCore.TouchInput.TouchInput (
             Transform player,
             float moveSpeed )
```

Initializes a new instance of the TouchInput class.

Parameters

player	Player.
moveSpeed	Move speed.

5.140.3 Member Function Documentation

5.140.3.1 GetMovementSpeed()

```
float GameCore.TouchInput.GetMovementSpeed ( )
```

Gets the players movement speed.

Returns

The movement speed.

Implements GameCore.PlayerInput.

5.140.3.2 GetVelocity()

```
Vector2 GameCore.TouchInput.GetVelocity ( )
```

Gets the velocity. The players next move.

Returns

The velocity.

Implements GameCore.PlayerInput.

5.140.3.3 SetMovementSpeed()

```
void GameCore.TouchInput.SetMovementSpeed (
             float amount )
```

Sets the players movement speed.

Parameters

amount	Move speed.

Implements GameCore.PlayerInput.

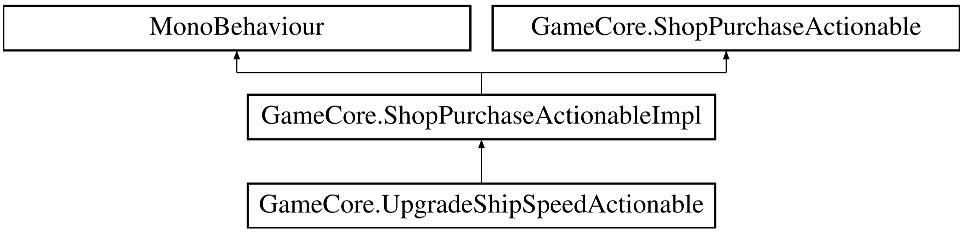
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Player/Movement/PlayerInput.cs

5.141 GameCore.UpgradeShipSpeedActionable Class Reference

Increases the players movement speed when purchased.

Inheritance diagram for GameCore.UpgradeShipSpeedActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.UpgradeShipSpeedActionable

Public Member Functions

- override void DoAction ()

    Invokes PlayerController::IncrementSpeed.

Public Attributes

- float speedIncrement

    The amount to increase players movement speed.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

5.141.1 Detailed Description

Increases the players movement speed when purchased.

5.141.2 Member Function Documentation

5.141.2.1 DoAction()

```
override void GameCore.UpgradeShipSpeedActionable.DoAction ( ) [virtual]
```

Invokes PlayerController::IncrementSpeed.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.141.3 Member Data Documentation

5.141.3.1 speedIncrement

```
float GameCore.UpgradeShipSpeedActionable.speedIncrement
```

The amount to increase players movement speed.

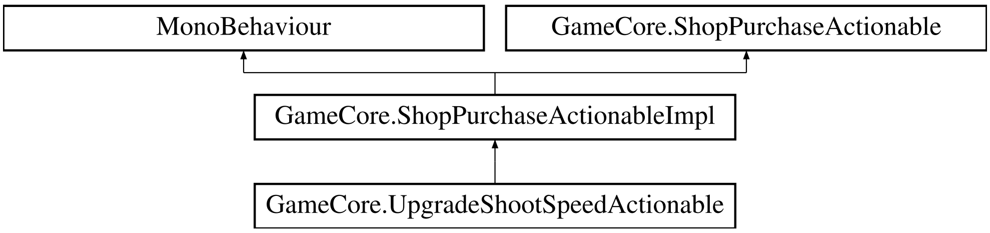
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/UpgradeShipSpeedActionable.cs

5.142 GameCore.UpgradeShootSpeedActionable Class Reference

Decreases time between shots for player when purchased.

Inheritance diagram for GameCore.UpgradeShootSpeedActionable:


MonoBehaviour
GameCore.ShopPurchaseActionable
GameCore.ShopPurchaseActionableImpl
GameCore.UpgradeShootSpeedActionable

Public Member Functions

- override void DoAction ()

  Decrements shoot speed for all PlayerShoot modules attached to the player.

Public Attributes

- float secsBetweenShotDecrements = 0.01f

  The amount (in seconds) to decrease the time between player shots.

Protected Member Functions

- override void **Awake** ()

Additional Inherited Members

5.142.1 Detailed Description

Decreases time between shots for player when purchased.

5.142.2 Member Function Documentation

5.142.2.1 DoAction()

```
override void GameCore.UpgradeShootSpeedActionable.DoAction ( )  [virtual]
```

Decrements shoot speed for all PlayerShoot modules attached to the player.

Reimplemented from GameCore.ShopPurchaseActionableImpl.

5.142.3 Member Data Documentation

5.142.3.1 secsBetweenShotDecrements

```
float GameCore.UpgradeShootSpeedActionable.secsBetweenShotDecrements = 0.01f
```

The amount (in seconds) to decrease the time between player shots.

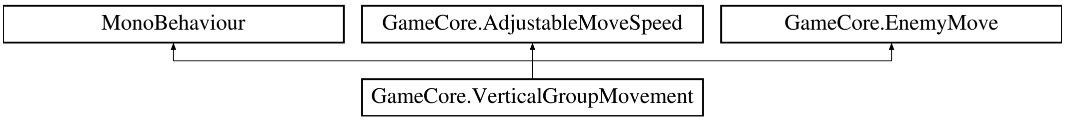
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Shop/UpgradeShootSpeedActionable.cs

5.143 GameCore.VerticalGroupMovement Class Reference

Controls groups of vertically moving enemies.

Inheritance diagram for GameCore.VerticalGroupMovement:



Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in for all child objects.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void IncrementSpeed ()

  Increments the speed near round end.

Public Attributes

- float moveSpeed = 0.04f

  The movement speed.
- float moveSpeedAdjustment = 0.01f

  The amount to increase the move speed near round end.
- MovementDirection moveDirection = MovementDirection.Right

  The movement direction.
- bool moveDown = true

  Sets whether this instance should move down when it reaches scren edge.

5.143.1 Detailed Description

Controls groups of vertically moving enemies.

5.143.2 Member Function Documentation

5.143.2.1 Begin()

```
void GameCore.VerticalGroupMovement.Begin ( )
```

Begin this instance. Starts fade in for all child objects.

Implements GameCore.EnemyMove.

5.143.2.2 IncrementSpeed()

```
void GameCore.VerticalGroupMovement.IncrementSpeed ( )
```

Increments the speed near round end.

Implements GameCore.AdjustableMoveSpeed.

5.143.2.3 Pause()

```
void GameCore.VerticalGroupMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.143.2.4 Resume()

```
void GameCore.VerticalGroupMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.143.3 Member Data Documentation

5.143.3.1 moveDirection

```
MovementDirection GameCore.VerticalGroupMovement.moveDirection = MovementDirection.Right
```

The movement direction.

5.143.3.2 moveDown

```
bool GameCore.VerticalGroupMovement.moveDown = true
```

Sets whether this instance should move down when it reaches scren edge.

5.143.3.3 moveSpeed

```
float GameCore.VerticalGroupMovement.moveSpeed = 0.04f
```

The movement speed.

5.143.3.4 moveSpeedAdjustment

```
float GameCore.VerticalGroupMovement.moveSpeedAdjustment = 0.01f
```

The amount to increase the move speed near round end.

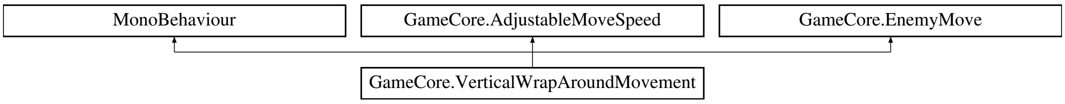
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/VerticalGroupMovement.cs

5.144 GameCore.VerticalWrapAroundMovement Class Reference

Controls enemies that move vertically and wrap around the screen.

Inheritance diagram for GameCore.VerticalWrapAroundMovement:


MonoBehaviour
GameCore.AdjustableMoveSpeed
GameCore.EnemyMove
GameCore.VerticalWrapAroundMovement

Public Member Functions

- void Begin ()

  Begin this instance. Starts fade in.
- void Pause ()

  Pause this instance.
- void Resume ()

  Resume this instance.
- void IncrementSpeed ()

  Increments the speed near round end.

Public Attributes

- MovementDirection movementDirection = MovementDirection.Left

  The movement direction.
- float moveSpeed = 10f

  The movement speed.
- float moveSpeedAdjustment = 2f

  The amount to increase movement speed near round end.
- bool oscillateY = false

  Sets whether this instance should move up and down on the y axis.
- bool removeWhenLastEnemy = false

  Sets whether this instance should be removed from the round when it has gone offscreen and is the last enemy.

5.144.1 Detailed Description

Controls enemies that move vertically and wrap around the screen.

5.144.2 Member Function Documentation

5.144.2.1 Begin()

```
void GameCore.VerticalWrapAroundMovement.Begin ( )
```

Begin this instance. Starts fade in.

Implements GameCore.EnemyMove.

5.144.2.2 IncrementSpeed()

```
void GameCore.VerticalWrapAroundMovement.IncrementSpeed ( )
```

Increments the speed near round end.

Implements GameCore.AdjustableMoveSpeed.

5.144.2.3 Pause()

```
void GameCore.VerticalWrapAroundMovement.Pause ( )
```

Pause this instance.

Implements GameCore.EnemyMove.

5.144.2.4 Resume()

```
void GameCore.VerticalWrapAroundMovement.Resume ( )
```

Resume this instance.

Implements GameCore.EnemyMove.

5.144.3 Member Data Documentation

5.144.3.1 movementDirection

```
MovementDirection GameCore.VerticalWrapAroundMovement.movementDirection = MovementDirection.↩
Left
```

The movement direction.

5.144.3.2 moveSpeed

```
float GameCore.VerticalWrapAroundMovement.moveSpeed = 10f
```

The movement speed.

5.144.3.3 moveSpeedAdjustment

```
float GameCore.VerticalWrapAroundMovement.moveSpeedAdjustment = 2f
```

The amount to increase movement speed near round end.

5.144.3.4 oscillateY

```
bool GameCore.VerticalWrapAroundMovement.oscillateY = false
```

Sets whether this instance should move up and down on the y axis.

5.144.3.5 removeWhenLastEnemy

```
bool GameCore.VerticalWrapAroundMovement.removeWhenLastEnemy = false
```

Sets whether this instance should be removed from the round when it has gone offscreen and is the last enemy.

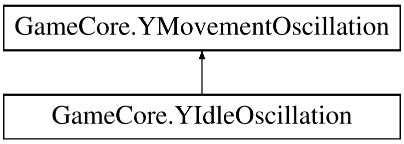
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/VerticalWrapAroundMovement.cs

5.145 GameCore.YIdleOscillation Class Reference

Y idle oscillation. No oscillation is performed.

Inheritance diagram for GameCore.YIdleOscillation:



Public Member Functions

- YIdleOscillation (Transform owner)

  Initializes a new instance of the YIdleOscillation class.
- Vector3 GetOscillation ()

  Returns owners position.

5.145.1 Detailed Description

Y idle oscillation. No oscillation is performed.

5.145.2 Constructor & Destructor Documentation

5.145.2.1 YIdleOscillation()

```
GameCore.YIdleOscillation.YIdleOscillation (
            Transform owner )
```

Initializes a new instance of the YIdleOscillation class.

Parameters

owner	Owner.

5.145.3 Member Function Documentation

5.145.3.1 GetOscillation()

```
Vector3 GameCore.YIdleOscillation.GetOscillation ( )
```

Returns owners position.

Returns

The oscillation vector.

Implements GameCore.YMovementOscillation.

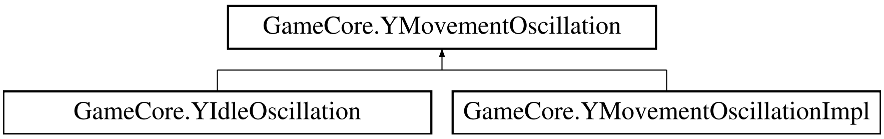
The documentation for this class was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/VerticalWrapAroundMovement.cs

5.146 GameCore.YMovementOscillation Interface Reference

Contract for perfoming Oscillation.

Inheritance diagram for GameCore.YMovementOscillation:


GameCore.YMovementOscillation
GameCore.YIdleOscillation
GameCore.YMovementOscillationImpl

Public Member Functions

- Vector3 GetOscillation ()
  Gets an oscillation vector.

5.146.1 Detailed Description

Contract for perfoming Oscillation.

5.146.2 Member Function Documentation

5.146.2.1 GetOscillation()

```
Vector3 GameCore.YMovementOscillation.GetOscillation ( )
```

Gets an oscillation vector.

Returns

The oscillation vector.

Implemented in GameCore.YIdleOscillation, and GameCore.YMovementOscillationImpl.

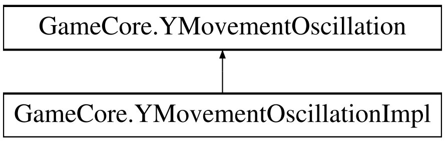
The documentation for this interface was generated from the following file:

- Pew Pew/Scripts/Enemies/Movement/VerticalWrapAroundMovement.cs

5.147 GameCore.YMovementOscillationImpl Class Reference

Implementation of Y Oscillation.

Inheritance diagram for GameCore.YMovementOscillationImpl:


GameCore.YMovementOscillation
GameCore.YMovementOscillationImpl

Public Member Functions

- YMovementOscillationImpl (Transform owner, float scale, float yOffset)

  Initializes a new instance of the YMovementOscillationImpl class.
- Vector3 GetOscillation ()

  Gets an oscillation vector.

5.147.1 Detailed Description

Implementation of Y Oscillation.

5.147.2 Constructor & Destructor Documentation

5.147.2.1 YMovementOscillationImpl()

```
GameCore.YMovementOscillationImpl.YMovementOscillationImpl (
            Transform owner,
            float scale,
            float yOffset )
```

Initializes a new instance of the YMovementOscillationImpl class.

Parameters

owner	Owner to oscillate.
scale	Scale of oscillation.
yOffset	Y offset.

5.147.3 Member Function Documentation

5.147.3.1 GetOscillation()

```
Vector3 GameCore.YMovementOscillationImpl.GetOscillation ( )
```

Gets an oscillation vector.

Returns

The oscillation vector.

Implements GameCore.YMovementOscillation.

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

- Pew Pew/Scripts/Enemies/Movement/VerticalWrapAroundMovement.cs

Index