

Namespace WaveZ.Client

Classes

[Game](#)

Represents the main game class responsible for managing game states and screens.

[NetworkClient](#)

The network client class is responsible for handling the connection to the server and processing incoming packets.

[Resources](#)

Get various types of resources by consistent identifiers.

Enums

[GameState](#)

Implementing enum GameState to switch and work with them

Class Game

Namespace: [WaveZ.Client](#)

Assembly: client.dll

Represents the main game class responsible for managing game states and screens.

```
public class Game
```

Inheritance

[object](#) ← Game

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Game()

Initializes a new instance of the [Game](#) class.

```
public Game()
```

Methods

Draw()

Draws the current screen.

```
public void Draw()
```

SwitchState(GameState)

Switches the current game state and updates the screen accordingly.

```
public void SwitchState(GameState newState)
```

Parameters

newState [GameState](#)

The new game state to switch to.

Update()

Updates the current screen.

```
public void Update()
```

Enum GameState

Namespace: [WaveZ.Client](#)

Assembly: client.dll

Implementing enum GameState to switch and work with them

```
public enum GameState
```

Fields

Exiting = 3

HowTo = 1

InGame = 2

MainMenu = 0

Class NetworkClient

Namespace: [WaveZ.Client](#)

Assembly: client.dll

The network client class is responsible for handling the connection to the server and processing incoming packets.

```
public static class NetworkClient
```

Inheritance

[object](#) ← NetworkClient

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

PacketProcessor

```
public static readonly NetPacketProcessor PacketProcessor
```

Field Value

NetPacketProcessor

ServerAddress

```
public static string ServerAddress
```

Field Value

[string](#)

Methods

Connect(string, int)

Connects to the server with the specified username and skin.

```
public static void Connect(string username, int selectedSkin)
```

Parameters

`username` [string](#)

The username of the player.

`selectedSkin` [int](#)

The selected skin of the player.

Initialize()

Initializes the network client and sets up the event listeners.

```
public static void Initialize()
```

SendPacket<T>(T)

Sends a packet to the server.

```
public static void SendPacket<T>(T packet) where T : class, new()
```

Parameters

`packet` [T](#)

The packet to send.

Type Parameters

T

The type of the packet to send.

Stop()

Stops the network client.

```
public static void Stop()
```

Update()

Updates the network client.

```
public static void Update()
```

Events

Connected

```
public static event Action? Connected
```

Event Type

[Action↗](#)

Class Resources

Namespace: [WaveZ.Client](#)

Assembly: client.dll

Get various types of resources by consistent identifiers.

```
public static class Resources
```

Inheritance

[object](#) ← Resources

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

LoadFont(string)

Load fonts located in 'Resources/Fonts'.

```
public static Font LoadFont(string path)
```

Parameters

path [string](#)

path of the font separated by '.', including extension, relative to 'Resources/Fonts'.

Returns

Font

Exceptions

[ArgumentException](#)

No font was found at **path**

LoadSound(string)

Load sounds located in 'Resources/Sounds'.

```
public static Sound LoadSound(string path)
```

Parameters

path [string](#)

path of the sound separated by '.', including extension, relative to 'Resources/Sounds'.

Returns

Sound

Exceptions

[ArgumentException](#)

No sound was found at **path**

LoadString(string)

Load text content of resource located in 'Resources'.

```
public static string LoadString(string path)
```

Parameters

path [string](#)

path of the resource separated by '.', including extension, relative to 'Resources'.

Returns

[string](#)

Exceptions

[ArgumentException](#)

No resource was found at `path`

LoadTexture(string)

Load texture located in 'Resources/Textures'.

```
public static Texture2D LoadTexture(string path)
```

Parameters

`path` [string](#)

path of the texture separated by '.', including extension, relative to 'Resources/Textures'.

Returns

Texture2D

Exceptions

[ArgumentException](#)

No texture was found at `path`

UnloadAll()

```
public static void UnloadAll()
```

Namespace WaveZ.Client.Gameplay

Classes

[GameObject](#)

Keeps track of all instances of its type. Is used by other system such as Drawables.

[GameplayScreen](#)

Represents the main gameplay/battlefield area.

Interfaces

[IDrawable](#)

Enables implementing classes to be drawn. Gets called in the Gameloop.

[ISolid](#)

Represents objects with collision

[IUpdatable](#)

IUpdateable for state updates based on deltaTime

Enums

[GameplayState](#)

Gamestates, that get switched during the gameplay.

[WaveState](#)

Different states of the zombiewave

Class GameObject

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

Keeps track of all instances of its type. Is used by other system such as Drawables.

```
public static class GameObject
```

Inheritance

[object](#) ← GameObject

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

GameObjects

```
public static IReadOnlyCollection<object> GameObjects { get; }
```

Property Value

[IReadOnlyCollection](#) <[object](#)>

Methods

Destroy(object)

```
public static void Destroy(object obj)
```

Parameters

obj [object](#)

Instantiate<T>(T)

```
public static T Instantiate<T>(T obj) where T : class
```

Parameters

obj T

Returns

T

Type Parameters

T

Class GameplayScreen

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

Represents the main gameplay/battlefield area.

```
public class GameplayScreen : IScreen
```

Inheritance

[object](#) ← GameplayScreen

Implements

[IScreen](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

GameplayScreen(Player?)

```
public GameplayScreen(Player? player = null)
```

Parameters

player [Player](#)

Methods

Draw()

```
public void Draw()
```

Unload()

```
public void Unload()
```

Update()

```
public void Update()
```

Enum GameState

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

Gamestates, that get switched during the gameplay.

```
public enum GameState
```

Fields

Buying = 2

Dead = 3

Pause = 1

Playing = 0

Interface IDrawable

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

Enables implementing classes to be drawn. Gets called in the Gameloop.

```
public interface IDrawable
```

Properties

Z

```
int z { get; }
```

Property Value

[int](#)

Methods

Draw()

```
void Draw()
```

DrawAll()

```
public static void DrawAll()
```

Interface ISolid

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

Represents objects with collision

```
public interface ISolid
```

Properties

Position

```
Vector2 Position { get; }
```

Property Value

[Vector2](#) ↗

Methods

GetCollisionBox()

```
Rectangle GetCollisionBox()
```

Returns

Rectangle

Interface IUpdatable

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

IUpdateable for state updates based on deltaTime

```
public interface IUpdatable
```

Methods

Update(float)

```
void Update(float deltaTime)
```

Parameters

deltaTime [float](#)

UpdateAll(float)

```
public static void UpdateAll(float deltaTime)
```

Parameters

deltaTime [float](#)

Enum WaveState

Namespace: [WaveZ.Client.Gameplay](#)

Assembly: client.dll

Different states of the zombiewave

```
public enum WaveState
```

Fields

NotStarted = 2

Rest = 1

Wave = 0

Namespace WaveZ.Client.Gameplay.Entities

Classes

[CameraManager](#)

Manages the current camera used in the game.

Class CameraManager

Namespace: [WaveZ.Client.Gameplay.Entities](#)

Assembly: client.dll

Manages the current camera used in the game.

```
public static class CameraManager
```

Inheritance

[object](#) ← CameraManager

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

CurrentCamera

Gets or sets the current camera used in the game.

```
public static Camera2D CurrentCamera { get; set; }
```

Property Value

Camera2D

Namespace WaveZ.Client.Gameplay.Entities. Enemies

Classes

[BossProjectile](#)

Represents a projectile fired by a BossZombie. Implements IDrawable and IUpdatable interfaces.

[BossZombie](#)

Represents a BossZombie, a more powerful version of a regular zombie with enhanced attributes.

[ExplosionZombie](#)

Represents an advanced type of zombie that can trigger an explosion animation. Inherits from the base Zombie class and includes features such as alternate textures, opacity effects, and explosion animations.

[SwordsBoss](#)

Represents a boss zombie that uses rotating swords.

[Zombie](#)

Represents a basic zombie entity in the game. Provides functionality for handling zombie updates, rendering, and network interactions.

Class BossProjectile

Namespace: [WaveZ.Client.Gameplay.Entities.Enemies](#)

Assembly: client.dll

Represents a projectile fired by a BossZombie. Implements IDrawable and IUpdatable interfaces.

```
public class BossProjectile : IDrawable, IUpdatable
```

Inheritance

[object](#) ← BossProjectile

Implements

[IDrawable](#), [IUpdatable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Direction

```
public Vector2 Direction { get; set; }
```

Property Value

[Vector2](#)

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Speed

```
public float Speed { get; }
```

Property Value

[float](#)

Z

```
public int Z { get; }
```

Property Value

[int](#)

Methods

AddProjectile(int, Vector2, Vector2)

```
public static void AddProjectile(int id, Vector2 position, Vector2 direction)
```

Parameters

`id int`

`position Vector2`

`direction Vector2`

Draw()

```
public void Draw()
```

RemoveProjectile(int)

```
public static void RemoveProjectile(int id)
```

Parameters

`id int`

Update(float)

```
public void Update(float deltaTime)
```

Parameters

`deltaTime float`

UpdateProjectile(int, Vector2)

```
public static void UpdateProjectile(int id, Vector2 position)
```

Parameters

`id int`

position [Vector2](#)

Class BossZombie

Namespace: [WaveZ.Client.Gameplay.Entities.Enemies](#)

Assembly: client.dll

Represents a BossZombie, a more powerful version of a regular zombie with enhanced attributes.

```
public class BossZombie : Zombie, IDrawable, ISolid, IUpdatable
```

Inheritance

[object](#) ← [Zombie](#) ← BossZombie

Implements

[IDrawable](#), [ISolid](#), [IUpdatable](#)

Inherited Members

[Zombie.Instances](#), [Zombie.position](#), [Zombie.rotation](#), [Zombie.radius](#), [Zombie.attackRadius](#),
[Zombie.zombieValue](#), [Zombie.zombieTexture](#), [Zombie.playerPosition](#), [Zombie.Z](#), [Zombie.id](#),
[Zombie.health](#), [Zombie.speed](#), [Zombie.MaxHealth](#), [Zombie.lastShotBy](#), [Zombie.SetupSubscriptions\(\)](#),
[Zombie.Update\(float\)](#), [Zombie.GetZombiesInRange\(float\)](#), [Zombie.Position](#), [Zombie.GetCollisionBox\(\)](#),
[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

BossZombie(string)

```
public BossZombie(string texturePath)
```

Parameters

texturePath [string](#)

Methods

Draw()

```
public override void Draw()
```

DrawHealthBar()

```
protected override void DrawHealthBar()
```

Class ExplosionZombie

Namespace: [WaveZ.Client.Gameplay.Entities.Enemies](#)

Assembly: client.dll

Represents an advanced type of zombie that can trigger an explosion animation. Inherits from the base Zombie class and includes features such as alternate textures, opacity effects, and explosion animations.

```
public class ExplosionZombie : Zombie, IDrawable, ISolid, IUpdatable
```

Inheritance

[object](#) ← [Zombie](#) ← [ExplosionZombie](#)

Implements

[IDrawable](#), [ISolid](#), [IUpdatable](#)

Inherited Members

[Zombie.Instances](#), [Zombie.position](#), [Zombie.rotation](#), [Zombie.radius](#), [Zombie.attackRadius](#),
[Zombie.zombieValue](#), [Zombie.zombieTexture](#), [Zombie.playerPosition](#), [Zombie.Z](#), [Zombie.id](#),
[Zombie.health](#), [Zombie.speed](#), [Zombie.MaxHealth](#), [Zombie.lastShotBy](#), [Zombie.SetupSubscriptions\(\)](#),
[Zombie.DrawHealthBar\(\)](#), [Zombie.GetZombiesInRange\(float\)](#), [Zombie.Position](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

ExplosionZombie(string)

Constructor to initialize an ExplosionZombie with a texture path.

```
public ExplosionZombie(string texturePath)
```

Parameters

texturePath [string](#)

The path to the texture of the explosion zombie.

ExplosionZombie(string, string)

```
public ExplosionZombie(string texturePath, string alternateTexturePath)
```

Parameters

texturePath [string ↗](#)

alternateTexturePath [string ↗](#)

Methods

Draw()

```
public override void Draw()
```

GetCollisionBox()

```
public override Rectangle GetCollisionBox()
```

Returns

Rectangle

TriggerExplosion()

```
public void TriggerExplosion()
```

Update(float)

```
public override void Update(float deltaTime)
```

Parameters

deltaTime [float](#)

Class SwordsBoss

Namespace: [WaveZ.Client.Gameplay.Entities.Enemies](#)

Assembly: client.dll

Represents a boss zombie that uses rotating swords.

```
public class SwordsBoss : Zombie, IDrawable, ISolid, IUpdatable
```

Inheritance

[object](#) ← [Zombie](#) ← SwordsBoss

Implements

[IDrawable](#), [ISolid](#), [IUpdatable](#)

Inherited Members

[Zombie.Instances](#), [Zombie.position](#), [Zombie.rotation](#), [Zombie.radius](#), [Zombie.attackRadius](#),
[Zombie.zombieValue](#), [Zombie.zombieTexture](#), [Zombie.playerPosition](#), [Zombie.Z](#), [Zombie.id](#),
[Zombie.health](#), [Zombie.speed](#), [Zombie.MaxHealth](#), [Zombie.lastShotBy](#), [Zombie.SetupSubscriptions\(\)](#),
[Zombie.DrawHealthBar\(\)](#), [Zombie.Update\(float\)](#), [Zombie.GetZombiesInRange\(float\)](#), [Zombie.Position](#),
[Zombie.GetCollisionBox\(\)](#), [object.Equals\(object\)](#), [object.Equals\(object, object\)](#),
[object.GetHashCode\(\)](#), [object.GetType\(\)](#), [object.MemberwiseClone\(\)](#),
[object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

SwordsBoss(string)

```
public SwordsBoss(string texturePath)
```

Parameters

texturePath [string](#)

Methods

Draw()

```
public override void Draw()
```

UpdateRotation(float, bool)

```
public void UpdateRotation(float newRotation, bool extended)
```

Parameters

newRotation [float](#)

extended [bool](#)

Class Zombie

Namespace: [WaveZ.Client.Gameplay.Entities.Enemies](#)

Assembly: client.dll

Represents a basic zombie entity in the game. Provides functionality for handling zombie updates, rendering, and network interactions.

```
public class Zombie : IDrawable, ISolid, IUpdatable
```

Inheritance

[object](#) ← Zombie

Implements

[IDrawable](#), [ISolid](#), [IUpdatable](#)

Derived

[BossZombie](#), [ExplosionZombie](#), [SwordsBoss](#)

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

Zombie(string)

```
public Zombie(string texturePath)
```

Parameters

texturePath [string](#)

Fields

Instances

```
protected static readonly Dictionary<int, Zombie> Instances
```

Field Value

[Dictionary](#)<[int](#), [Zombie](#)>

_playerPosition

```
protected readonly Vector2 _playerPosition
```

Field Value

[Vector2](#)

_zombieTexture

```
protected readonly Texture2D _zombieTexture
```

Field Value

Texture2D

attackRadius

```
protected float attackRadius
```

Field Value

[float](#)

health

```
public int health
```

Field Value

[int](#)

id

```
public int id
```

Field Value

[int](#)

lastShotBy

```
public Player lastShotBy
```

Field Value

[Player](#)

position

```
protected Vector2 position
```

Field Value

[Vector2](#)

radius

```
protected float radius
```

Field Value

[float](#)

rotation

```
protected float rotation
```

Field Value

[float](#)

speed

```
public float speed
```

Field Value

[float](#)

zombieValue

```
protected int zombieValue
```

Field Value

[int](#)

Properties

MaxHealth

```
public int MaxHealth { get; protected set; }
```

Property Value

[int ↗](#)

Position

```
public Vector2 Position { get; }
```

Property Value

[Vector2 ↗](#)

Z

```
public int Z { get; }
```

Property Value

[int ↗](#)

Methods

Draw()

```
public virtual void Draw()
```

DrawHealthBar()

```
protected virtual void DrawHealthBar()
```

GetCollisionBox()

```
public virtual Rectangle GetCollisionBox()
```

Returns

Rectangle

GetZombiesInRange(float)

```
public List<Zombie> GetZombiesInRange(float distanceThreshold)
```

Parameters

`distanceThreshold` float ↗

Returns

[List ↗](#) <[Zombie](#)>

SetupSubscriptions()

Sets up the subscriptions to network packets related to zombies.

```
public static void SetupSubscriptions()
```

Update(float)

```
public virtual void Update(float deltaTime)
```

Parameters

`deltaTime` float ↗

Namespace WaveZ.Client.Gameplay.Entities. Map

Classes

[Barrel](#)

Represents a barrel in the game that can be drawn, updated, and detected for collisions. The barrel can trigger an explosion animation and send network packets when it explodes or is destroyed.

[Wall](#)

Represents a wall in the game. This wall is drawable and solid, meaning it can be used for collision detection.

Class Barrel

Namespace: [WaveZ.Client.Gameplay.Entities.Map](#)

Assembly: client.dll

Represents a barrel in the game that can be drawn, updated, and detected for collisions. The barrel can trigger an explosion animation and send network packets when it explodes or is destroyed.

```
public class Barrel : IDrawable, ISolid, IUpdatable
```

Inheritance

[object](#) ← Barrel

Implements

[IDrawable](#), [ISolid](#), [IUpdatable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Barrel()

```
public Barrel()
```

Fields

id

```
public int id
```

Field Value

[int](#)

Properties

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Z

```
public int Z { get; }
```

Property Value

[int](#)

Methods

Draw()

```
public void Draw()
```

GetCollisionBox()

```
public Rectangle GetCollisionBox()
```

Returns

Rectangle

TriggerExplosion()

```
public void TriggerExpllosion()
```

Update(float)

```
public void Update(float deltaTime)
```

Parameters

deltaTime [float](#)

Class Wall

Namespace: [WaveZ.Client.Gameplay.Entities.Map](#)

Assembly: client.dll

Represents a wall in the game. This wall is drawable and solid, meaning it can be used for collision detection.

```
public class Wall : IDrawable, ISolid
```

Inheritance

[object](#) ← Wall

Implements

[IDrawable](#), [ISolid](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Z

```
public int Z { get; }
```

Property Value

[int ↗](#)

Methods

Draw()

```
public void Draw()
```

GetCollisionBox()

```
public Rectangle GetCollisionBox()
```

Returns

Rectangle

Namespace WaveZ.Client.Gameplay.Entities.

Player

Classes

[Bullet](#)

Represents a bullet fired by the player.

[Gun](#)

Represents a gun used by the player in the game.

[Player](#)

Represents the mainplayer in the game.

[PlayerTextures](#)

Manages the ALL textures for different player characters and their weapons.

[RemoteBullet](#)

Represents a remote bullet in the game, which can be drawn on the screen.

[RemoteGun](#)

Represents a remote gun in the game, which can be drawn on the screen.

[Weapon](#)

Implementing all functionalitys of the weapons

Enums

[ShootingMode](#)

Represents the different shooting modes available for a player.

[WeaponType](#)

Class Bullet

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Represents a bullet fired by the player.

```
public sealed class Bullet : IDrawable, IUpdatable
```

Inheritance

[object](#) ← Bullet

Implements

[IDrawable](#), [IUpdatable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Bullet(Vector2, Vector2, int, Player)

```
public Bullet(Vector2 position, Vector2 velocity, int damage, Player shooter)
```

Parameters

position [Vector2](#)

velocity [Vector2](#)

damage [int](#)

shooter [Player](#)

Fields

shooter

```
public Player shooter
```

Field Value

[Player](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int↗](#)

IsActive

```
public bool IsActive { get; }
```

Property Value

[bool↗](#)

Z

```
public int Z { get; }
```

Property Value

[int↗](#)

Methods

Draw()

```
public void Draw()
```

RemoveCollectedBullets()

Removes all collected bullets that are marked for removal.

```
public static void RemoveCollectedBullets()
```

Update(float)

```
public void Update(float deltaTime)
```

Parameters

deltaTime [float](#)

Class Gun

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Represents a gun used by the player in the game.

```
public sealed class Gun : IDrawable, IUpdatable
```

Inheritance

[object](#) ← Gun

Implements

[IDrawable](#), [IUpdatable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Gun(Vector2)

Represents a gun used by the player in the game.

```
public Gun(Vector2 position)
```

Parameters

position [Vector2](#)

Properties

BulletSpeed

```
public float BulletSpeed { get; }
```

Property Value

[float](#)

Bullets

public List<Bullet> Bullets { get; }

Property Value

[List](#)<[Bullet](#)>

Position

public Vector2 Position { get; set; }

Property Value

[Vector2](#)

Z

public int Z { get; }

Property Value

[int](#)

Methods

Draw()

public void Draw()

Update(float)

```
public void Update(float deltaTime)
```

Parameters

deltaTime [float](#)

Class Player

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Represents the mainplayer in the game.

```
public sealed class Player : IDrawable, IUpdatable
```

Inheritance

[object](#) ← Player

Implements

[IDrawable](#), [IUpdatable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Player(int, int, int, int, IPlayerTargeting?)

```
public Player(int screenWidth, int screenHeight, int width, int height, IPlayerTargeting?  
targeting = null)
```

Parameters

screenWidth [int](#)

screenHeight [int](#)

width [int](#)

height [int](#)

targeting [IPlayerTargeting](#)

Properties

CurrentSkin

```
public int CurrentSkin { get; set; }
```

Property Value

[int ↗](#)

Gun

```
public Gun Gun { get; }
```

Property Value

[Gun](#)

GunShootingMode

```
public ShootingMode GunShootingMode { get; }
```

Property Value

[ShootingMode](#)

Health

```
public float Health { get; set; }
```

Property Value

[float ↗](#)

Instance

```
public static Player? Instance { get; }
```

Property Value

[Player](#)

Money

```
public int Money { get; set; }
```

Property Value

[int↗](#)

Score

```
public int Score { get; set; }
```

Property Value

[int↗](#)

SelectedWeaponIndex

```
public int SelectedWeaponIndex { get; set; }
```

Property Value

[int↗](#)

ShootingInterval

```
public float ShootingInterval { get; }
```

Property Value

[float](#)

Speed

```
public float Speed { get; set; }
```

Property Value

[float](#)

Transform

```
public Transform2D Transform { get; set; }
```

Property Value

[Transform2D](#)

WeaponSlots

```
public List<Weapon> WeaponSlots { get; set; }
```

Property Value

[List](#) <[Weapon](#)>

Z

```
public int z { get; }
```

Property Value

[int](#)

Methods

Draw()

```
public void Draw()
```

GetCollisionBox(Vector2)

```
public Rectangle GetCollisionBox(Vector2 position)
```

Parameters

position [Vector2](#)

Returns

Rectangle

Update(float)

```
public void Update(float deltaTime)
```

Parameters

deltaTime [float](#)

UpdatePlayerTexture()

```
public void UpdatePlayerTexture()
```

Events

OnPlayerDied

```
public event Action? OnPlayerDied
```

Event Type

[Action](#) ↗

Class PlayerTextures

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Manages the ALL textures for different player characters and their weapons.

```
public class PlayerTextures
```

Inheritance

[object](#) ← PlayerTextures

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

PlayerTextures()

```
public PlayerTextures()
```

Properties

AllTextures

```
public Texture2D[][][] AllTextures { get; }
```

Property Value

Texture2D[][]

HitmanTextures

```
public Texture2D[] HitmanTextures { get; }
```

Property Value

Texture2D[]

RobotTextures

```
public Texture2D[] RobotTextures { get; }
```

Property Value

Texture2D[]

SoldierTextures

```
public Texture2D[] SoldierTextures { get; }
```

Property Value

Texture2D[]

StartTexture

```
public Texture2D[] StartTexture { get; }
```

Property Value

Texture2D[]

SurvivorTextures

```
public Texture2D[] SurvivorTextures { get; }
```

Property Value

Texture2D[]

Class RemoteBullet

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Represents a remote bullet in the game, which can be drawn on the screen.

```
public class RemoteBullet : IDrawable
```

Inheritance

[object](#) ← RemoteBullet

Implements

[IDrawable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

Gets or sets the unique identifier for the bullet.

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

Gets or sets the position of the bullet.

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#) ↗

Velocity

Gets or sets the velocity of the bullet.

```
public Vector2 Velocity { get; set; }
```

Property Value

[Vector2](#) ↗

Z

Gets the Z-order of the bullet for drawing purposes.

```
public int Z { get; }
```

Property Value

[int](#) ↗

Methods

Draw()

Draws the bullet on the screen.

```
public void Draw()
```

Class RemoteGun

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Represents a remote gun in the game, which can be drawn on the screen.

```
public class RemoteGun : IDrawable
```

Inheritance

[object](#) ← RemoteGun

Implements

[IDrawable](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

Direction

Gets or sets the direction the gun is facing.

```
public Vector2 Direction
```

Field Value

[Vector2](#)

Position

Gets or sets the position of the gun.

```
public Vector2 Position
```

Field Value

[Vector2](#)

Properties

Id

Gets or sets the unique identifier for the gun.

```
public int Id { get; set; }
```

Property Value

[int](#)

Z

Gets the Z-order of the gun for drawing purposes.

```
public int Z { get; }
```

Property Value

[int](#)

Methods

Draw()

Draws the gun on the screen.

```
public void Draw()
```

Enum ShootingMode

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Represents the different shooting modes available for a player.

```
public enum ShootingMode
```

Fields

Automatic = 2

Burst = 1

Semiautomatic = 0

Class Weapon

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

Implementing all functionalitys of the weapons

```
public class Weapon
```

Inheritance

[object](#) ← Weapon

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Damage

```
public required float Damage { get; set; }
```

Property Value

[float](#)

IsUnlocked

```
public bool IsUnlocked { get; set; }
```

Property Value

[bool](#)

Name

```
public required string Name { get; init; }
```

Property Value

[string ↗](#)

Price

```
public required int Price { get; init; }
```

Property Value

[int ↗](#)

ShootingInterval

```
public required float ShootingInterval { get; init; }
```

Property Value

[float ↗](#)

ShootingMode

```
public required ShootingMode ShootingMode { get; init; }
```

Property Value

[ShootingMode](#)

Type

```
public required WeaponType Type { get; init; }
```

Property Value

[WeaponType](#)

Enum WeaponType

Namespace: [WaveZ.Client.Gameplay.Entities.Player](#)

Assembly: client.dll

```
public enum WeaponType
```

Fields

Pistol = 0

SilencedPistol = 1

Uzi = 2

Namespace WaveZ.Client.Gameplay.Entities.Player.Targeting

Classes

[PlayerInputTargeting](#)

Represents the player input targeting system, which calculates the target movement direction based on player input.

Interfaces

[IPlayerTargeting](#)

Defines the interface for player targeting functionality.

Interface IPlayerTargeting

Namespace: [WaveZ.Client.Gameplay.Entities.Player.Targeting](#)

Assembly: client.dll

Defines the interface for player targeting functionality.

```
public interface IPlayerTargeting
```

Methods

GetTargetMovement(Player)

Gets the target movement direction for the player.

```
Vector2 GetTargetMovement(Player player)
```

Parameters

player [Player](#)

The player for whom the target movement is being calculated.

Returns

[Vector2](#)

A normalized vector containing the direction the player should move to.

Class PlayerInputTargeting

Namespace: [WaveZ.Client.Gameplay.Entities.Player.Targeting](#)

Assembly: client.dll

Represents the player input targeting system, which calculates the target movement direction based on player input.

```
public sealed class PlayerInputTargeting : IPlayerTargeting
```

Inheritance

[object](#) ← PlayerInputTargeting

Implements

[IPlayerTargeting](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

GetTargetMovement(Player)

Gets the target movement direction for the player based on keyboard input.

```
public Vector2 GetTargetMovement(Player player)
```

Parameters

player [Player](#)

The player for whom the target movement is being calculated.

Returns

[Vector2](#)

A normalized vector containing the direction the player should move to.

Namespace WaveZ.Client.Gameplay.UI

Classes

[GameplayOverlay](#)

Class GameplayOverlay

Namespace: [WaveZ.Client.Gameplay.UI](#)

Assembly: client.dll

```
public class GameplayOverlay
```

Inheritance

[object](#) ← GameplayOverlay

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

GameplayOverlay(Player, GameplayState, List<int>, float)

Initializes a new instance of the [GameplayOverlay](#) class.

```
public GameplayOverlay(Player player, GameplayState gameplayState, List<int> scores,  
float health)
```

Parameters

player [Player](#)

The player instance.

gameplayState [GameplayState](#)

The current gameplay state.

scores [List](#)<[int](#)>

The list of player scores.

health [float](#)

The player's health.

Properties

CurrentWave

```
public static int CurrentWave { get; set; }
```

Property Value

[int ↗](#)

Methods

Draw()

Draws the gameplay overlay on the screen.

```
public void Draw()
```

Unload()

```
public void Unload()
```

Update(Player, List<int>, float)

Updates the gameplay overlay with the latest player, scores, and health information.

```
public void Update(Player player, List<int> scores, float health)
```

Parameters

player [Player](#)

The player instance.

scores [List ↗ <int ↗ >](#)

The list of player scores.

`health` [float](#)

The player's health.

Namespace WaveZ.Client.Menu

Classes

[HowToScreen](#)

[MainMenuScreen](#)

Implementing MainMenuScreen (Starting Screen) to navigate through the Game

[StartGameModal](#)

Represents the modal window for starting a game.

Class HowToScreen

Namespace: [WaveZ.Client.Menu](#)

Assembly: client.dll

```
public class HowToScreen : IScreen
```

Inheritance

[object](#) ← HowToScreen

Implements

[IScreen](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

HowToScreen(Action<GameState>)

```
public HowToScreen(Action<GameState> onSwitchState)
```

Parameters

onSwitchState [Action](#)<[GameState](#)>

Properties

Z

```
public int Z { get; }
```

Property Value

[int](#)

Methods

Draw()

```
public void Draw()
```

Update()

```
public void Update()
```

Class MainMenuScreen

Namespace: [WaveZ.Client.Menu](#)

Assembly: client.dll

Implementing MainMenuScreen (Starting Screen) to navigate through the Game

```
public class MainMenuScreen : IScreen
```

Inheritance

[object](#) ← MainMenuScreen

Implements

[IScreen](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

MainMenuScreen(Action<GameState>)

```
public MainMenuScreen(Action<GameState> onSwitchState)
```

Parameters

onSwitchState [Action](#)<[GameState](#)>

Properties

Z

```
public int Z { get; }
```

Property Value

Methods

Draw()

```
public void Draw()
```

Update()

```
public void Update()
```

Class StartGameModal

Namespace: [WaveZ.Client.Menu](#)

Assembly: client.dll

Represents the modal window for starting a game.

```
public class StartGameModal
```

Inheritance

[object](#) ← StartGameModal

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

StartGameModal(Action, Action<string, int, string, string, string>?)

Initializes a new instance of the [StartGameModal](#) class.

```
public StartGameModal(Action onClose, Action<string, int, string, string, string>? onPlay)
```

Parameters

onClose [Action](#)

Action to be performed when the modal is closed.

onPlay [Action](#)<[string](#), [int](#), [string](#), [string](#), [string](#)>

Action to be performed when the play button is clicked.

Properties

getSelectedSkin

```
public static int getSelectedSkin { get; }
```

Property Value

[int](#)

Methods

Draw()

Draws the modal window.

```
public void Draw()
```

Update()

Updates the modal window.

```
public void Update()
```

Events

OnPlay

Event triggered when the play button is clicked.

```
public static event Action<string, int, string, string, string>? OnPlay
```

Event Type

[Action](#)<[string](#), [int](#), [string](#), [string](#), [string](#)>

Namespace WaveZ.Client.Screens

Interfaces

[IScreen](#)

defines methods for updating, drawing, and optionally unloading a screen

Interface IScreen

Namespace: [WaveZ.Client.Screens](#)

Assembly: client.dll

defines methods for updating, drawing, and optionally unloading a screen

```
public interface IScreen
```

Methods

Draw()

```
void Draw()
```

Unload()

```
void Unload()
```

Update()

```
void Update()
```

Namespace WaveZ.Client.UI

Classes

[Button](#)

Represents a clickable button with customizable appearance and behavior.

[TextBox](#)

Represents a text box control for user input with basic text editing functionality.

[Title](#)

Represents a title text with customizable appearance and rendering.

Class Button

Namespace: [WaveZ.Client.UI](#)

Assembly: client.dll

Represents a clickable button with customizable appearance and behavior.

```
public class Button
```

Inheritance

[object](#) ← Button

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Button(Rectangle, string, Action, Font, bool)

Initializes a new instance of the [Button](#) class with specified parameters, using a custom font.

```
public Button(Rectangle rectangle, string text, Action onClick, Font customFont, bool  
isPrimary = false)
```

Parameters

rectangle Rectangle

The rectangle defining the position and size of the button.

text [string](#)

The text displayed on the button.

onClick [Action](#)

The action to be executed when the button is clicked.

customFont Font

The custom font used for the button's text.

isPrimary [bool](#)

Indicates whether the button is a primary button.

Button(Rectangle, string, Action, bool)

Initializes a new instance of the [Button](#) class with specified parameters, using a default font.

```
public Button(Rectangle rectangle, string text, Action onClick, bool isPrimary = false)
```

Parameters

rectangle [Rectangle](#)

The rectangle defining the position and size of the button.

text [string](#)

The text displayed on the button.

onClick [Action](#)

The action to be executed when the button is clicked.

isPrimary [bool](#)

Indicates whether the button is a primary button.

Fields

OnClick

Gets or sets the action to be executed when the button is clicked.

```
public Action OnClick
```

Field Value

[Action](#)

Rectangle

Gets or sets the rectangle defining the position and size of the button.

```
public Rectangle Rectangle
```

Field Value

Rectangle

Text

Gets or sets the text displayed on the button.

```
public string Text
```

Field Value

[string](#)

Methods

Draw()

Draws the button using the default font size.

```
public void Draw()
```

DrawExt(float)

Draws the button using a specified font size.

```
public void DrawExt(float fontsize)
```

Parameters

fontsize [float](#)

The size of the font used for drawing the button's text.

Update()

Updates the state of the button, checking for click interactions.

```
public void Update()
```

Class TextBox

Namespace: [WaveZ.Client.UI](#)

Assembly: client.dll

Represents a text box control for user input with basic text editing functionality.

```
public class TextBox
```

Inheritance

[object](#) ← TextBox

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

TextBox(Rectangle, string)

Initializes a new instance of the [TextBox](#) class with specified parameters.

```
public TextBox(Rectangle rectangle, string placeholderText)
```

Parameters

rectangle Rectangle

The rectangle defining the position and size of the text box.

placeholderText [string](#)

The initial text displayed as a placeholder in the text box.

Properties

Rectangle

Gets the rectangle defining the position and size of the text box.

```
public Rectangle Rectangle { get; }
```

Property Value

Rectangle

Text

Gets or sets the current text entered in the text box.

```
public string Text { get; }
```

Property Value

[string](#) ↗

Methods

Draw()

Draws the text box and its current text content.

```
public void Draw()
```

Update()

Updates the state of the text box based on user interactions.

```
public void Update()
```

Class Title

Namespace: [WaveZ.Client.UI](#)

Assembly: client.dll

Represents a title text with customizable appearance and rendering.

```
public class Title
```

Inheritance

[object](#) ← Title

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Title(string, Vector2, int, Color, Color)

Initializes a new instance of the [Title](#) class with specified parameters, using a default font.

```
public Title(string text, Vector2 position, int fontSize, Color primaryColor,  
Color secondaryColor)
```

Parameters

text [string](#)

The text content of the title.

position [Vector2](#)

The position where the title will be drawn.

fontSize [int](#)

The size of the font used to render the title.

primaryColor [Color](#)

The primary color of the title's text.

secondaryColor Color

The secondary color used for specific characters in the title.

Title(string, Vector2, int, Color, Color, Font)

Initializes a new instance of the [Title](#) class with specified parameters, allowing a custom font to be used.

```
public Title(string text, Vector2 position, int fontSize, Color primaryColor, Color  
secondaryColor, Font font)
```

Parameters

text [string](#)

The text content of the title.

position [Vector2](#)

The position where the title will be drawn.

fontSize [int](#)

The size of the font used to render the title.

primaryColor Color

The primary color of the title's text.

secondaryColor Color

The secondary color used for specific characters in the title.

font Font

The custom font used to render the title.

Properties

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Methods

Draw()

Draws the title text using the specified font, colors, and position.

```
public void Draw()
```

Namespace WaveZ.Client.Ui

Classes

[Dropdown](#)

Represents a dropdown menu control with selectable options.

Class Dropdown

Namespace: [WaveZ.Client.Ui](#)

Assembly: client.dll

Represents a dropdown menu control with selectable options.

```
public class Dropdown
```

Inheritance

[object](#) ← Dropdown

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

Dropdown(Rectangle, string[], int)

Initializes a new instance of the [Dropdown](#) class with specified parameters.

```
public Dropdown(Rectangle rectangle, string[] options, int selectedIndex)
```

Parameters

rectangle Rectangle

The rectangle defining the position and size of the dropdown.

options [string](#)[]

The array of options available in the dropdown.

selectedIndex [int](#)

The index of the initially selected option.

Properties

Options

Gets the array of options available in the dropdown.

```
public string[] Options { get; }
```

Property Value

[string](#)[]

Rectangle

Gets the rectangle defining the position and size of the dropdown.

```
public Rectangle Rectangle { get; }
```

Property Value

Rectangle

SelectedIndex

Gets or sets the index of the currently selected option.

```
public int SelectedIndex { get; }
```

Property Value

[int](#)[]

Methods

Draw()

Draws the dropdown and its current selected option.

```
public void Draw()
```

Update()

Updates the state of the dropdown, handling mouse interactions.

```
public void Update()
```

Namespace WaveZ.Server

Classes

[Barrel](#)

Represents a barrel that can be placed and exploded, dealing damage within a specified radius.

[BossProjectile](#)

Represents a projectile fired by a zombie boss, which moves in the direction of the closest player and does damage on collision with a player

[BossZombie](#)

Represents a boss zombie, that is a subclass of [Zombie](#), that can fire projectiles at players, spawning instances of [BossProjectile](#).

[ExplosionZombie](#)

Represents an explosion zombie that deals damage within a specified radius upon death, subclass of [Zombie](#).

[Globals](#)

[NetworkServer](#)

Manages server networking, including player connections, packet handling, and game state updates.

[Player](#)

Represents a player in the game, including their transform, health, and current skin and weapon.

[SwordsBoss](#)

Represents a SwordsBoss zombie with unique behavior and attacks, subclass of [Zombie](#).

[Wall](#)

Represents a wall in the game and manages wall instances and board generation.

[WaveController](#)

Controls the waves of enemies in the game, including initialization, spawning, and updating waves.

[Zombie](#)

Represents a zombie in the game, with various types of zombies inheriting from this class, on the server-side. Manages zombie instances, spawning, updating, and handling interactions with players.

Class Barrel

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a barrel that can be placed and exploded, dealing damage within a specified radius.

```
public class Barrel
```

Inheritance

[object](#) ← Barrel

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

Gets the unique identifier of the barrel

```
public int Id { get; }
```

Property Value

[int](#)

Methods

Explode(int)

Explodes the barrel with the specified identifier

```
public static void Explode(int id)
```

Parameters

id [int ↗](#)

The identifier of the barrel to explode

Place(Vector2)

Places a new barrel at the specified position

```
public static Barrel Place(Vector2 position)
```

Parameters

position [Vector2 ↗](#)

The position to place the barrel

Returns

[Barrel](#)

The placed barrel

Remove(int)

Removes the barrel with the specified identifier.

```
public static void Remove(int id)
```

Parameters

id [int ↗](#)

The identifier of the barrel to remove.

Class BossProjectile

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a projectile fired by a zombie boss, which moves in the direction of the closest player and does damage on collision with a player

```
public class BossProjectile
```

Inheritance

[object](#) ← BossProjectile

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

BossProjectile(int, Vector2, Vector2)

Initializes a new instance of the [BossProjectile](#) class.

```
public BossProjectile(int bossId, Vector2 position, Vector2 direction)
```

Parameters

bossId [int](#)

The identifier of the boss that fired the projectile.

position [Vector2](#)

The initial position of the projectile.

direction [Vector2](#)

The direction in which the projectile is moving.

Properties

BossId

```
public int BossId { get; }
```

Property Value

[int ↗](#)

Damage

```
public float Damage { get; }
```

Property Value

[float ↗](#)

Direction

```
public Vector2 Direction { get; }
```

Property Value

[Vector2 ↗](#)

Id

```
public int Id { get; }
```

Property Value

[int ↗](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#) ↗

Radius

```
public float Radius { get; }
```

Property Value

[float](#) ↗

Speed

```
public float Speed { get; }
```

Property Value

[float](#) ↗

Methods

AddProjectile(BossProjectile)

Adds a projectile to the list of active projectiles.

```
public static void AddProjectile(BossProjectile projectile)
```

Parameters

[projectile](#) [BossProjectile](#)

The projectile to add.

UpdateProjectiles()

Updates the positions of all active projectiles, notifies clients of any changes, and removes any projectiles that have collided with a player.

```
public static void UpdateProjectiles()
```

Class BossZombie

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a boss zombie, that is a subclass of [Zombie](#), that can fire projectiles at players, spawning instances of [BossProjectile](#).

```
public class BossZombie : Zombie
```

Inheritance

[object](#) ← [Zombie](#) ← BossZombie

Inherited Members

[Zombie.ReadOnlyInstances](#) , [Zombie.Id](#) , [Zombie.type](#) , [Zombie.Transform](#) , [Zombie.AttackRadius](#) ,
[Zombie.Speed](#) , [Zombie.health](#) , [Zombie.SetupSubscriptions\(\)](#) , [Zombie.GenerateInstances\(uint\)](#) ,
[Zombie.GenerateInstancesFast\(uint\)](#) , [Zombie.GenerateInstancesExplosion\(uint\)](#) ,
[Zombie.GenerateInstancesBoss\(uint\)](#) , [Zombie.GenerateInstancesSwords\(uint\)](#) ,
[Zombie.UpdateInstances\(\)](#) , [Zombie.FindNearestPlayer\(\)](#) , [Zombie.GetZombieById\(int\)](#) ,
[Zombie.RemoveZombie\(Zombie\)](#) , [Zombie.Damage\(int\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

BossZombie()

```
public BossZombie()
```

Properties

MaxHealth

```
public override int MaxHealth { get; }
```

Property Value

[int ↗](#)

Methods

Update()

Updates the boss zombie's state, including firing projectiles at players.

```
protected override void Update()
```

Class ExplosionZombie

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents an explosion zombie that deals damage within a specified radius upon death, subclass of [Zombie](#).

```
public class ExplosionZombie : Zombie
```

Inheritance

[object](#) ← [Zombie](#) ← ExplosionZombie

Inherited Members

[Zombie.ReadOnlyInstances](#), [Zombie.Id](#), [Zombie.type](#), [Zombie.Transform](#), [Zombie.AttackRadius](#),
[Zombie.Speed](#), [Zombie.health](#), [Zombie.SetupSubscriptions\(\)](#), [Zombie.GenerateInstances\(uint\)](#),
[Zombie.GenerateInstancesFast\(uint\)](#), [Zombie.GenerateInstancesExplosion\(uint\)](#),
[Zombie.GenerateInstancesBoss\(uint\)](#), [Zombie.GenerateInstancesSwords\(uint\)](#),
[Zombie.UpdateInstances\(\)](#), [Zombie.FindNearestPlayer\(\)](#), [Zombie.GetZombieById\(int\)](#),
[Zombie.RemoveZombie\(Zombie\)](#), [Zombie.Update\(\)](#), [object.Equals\(object\)](#),
[object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

ExplosionZombie()

```
public ExplosionZombie()
```

Fields

isExploding

```
public bool isExploding
```

Field Value

[bool](#) ↗

Properties

MaxHealth

```
public override int MaxHealth { get; }
```

Property Value

[int](#) ↗

Methods

Damage(int)

Triggers an explosion if health drops to zero.

```
public override void Damage(int amount)
```

Parameters

amount [int](#) ↗

The amount of damage to inflict.

Class Globals

Namespace: [WaveZ.Server](#)

Assembly: server.dll

```
public static class Globals
```

Inheritance

[object](#) ← Globals

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

IsTestModeEnabled

```
public static readonly bool IsTestModeEnabled
```

Field Value

[bool](#)

TickCountMs

```
public const int TickDurationMs = 15
```

Field Value

[int](#)

Class NetworkServer

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Manages server networking, including player connections, packet handling, and game state updates.

```
public static class NetworkServer
```

Inheritance

[object](#) ← NetworkServer

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

ScreenHeight

```
public const int ScreenHeight = 450
```

Field Value

[int](#)

ScreenWidth

```
public const int ScreenWidth = 800
```

Field Value

[int](#)

Methods

SendPacketToAll<T>(T)

```
public static void SendPacketToAll<T>(T packet) where T : class, new()
```

Parameters

packet T

Type Parameters

T

SendPacket<T>(int, T)

```
public static void SendPacket<T>(int peerId, T packet) where T : class, new()
```

Parameters

peerId int

packet T

Type Parameters

T

Start(int)

Starts the server on the specified port.

```
public static void Start(int port)
```

Parameters

port int

The port to start the server on.

Stop()

```
public static void Stop()
```

Update()

```
public static void Update()
```

Events

PlayerJoined

```
public static event Action<int, string, int>? PlayerJoined
```

Event Type

[Action<int, string, int>](#)

PlayerTransformUpdated

```
public static event Action<int, Transform2D>? PlayerTransformUpdated
```

Event Type

[Action<int, Transform2D>](#)

Class Player

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a player in the game, including their transform, health, and current skin and weapon.

```
public sealed class Player
```

Inheritance

[object](#) ← Player

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

Instances

```
public static readonly Dictionary<int, Player> Instances
```

Field Value

[Dictionary](#)<[int](#), [Player](#)>

Properties

CurrentSkin

```
public int CurrentSkin { get; set; }
```

Property Value

[int](#)

CurrentWeaponIndex

```
public int CurrentWeaponIndex { get; set; }
```

Property Value

[int ↗](#)

Health

```
public float Health { get; set; }
```

Property Value

[float ↗](#)

Name

```
public required string Name { get; init; }
```

Property Value

[string ↗](#)

PeerId

```
public required int PeerId { get; init; }
```

Property Value

[int ↗](#)

Transform

```
public Transform2D Transform { get; }
```

Property Value

[Transform2D](#)

Methods

SetupSubscriptions()

Sets up network subscriptions for player join and transform update events.

```
public static void SetupSubscriptions()
```

Class SwordsBoss

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a SwordsBoss zombie with unique behavior and attacks, subclass of [Zombie](#).

```
public class SwordsBoss : Zombie
```

Inheritance

[object](#) ← [Zombie](#) ← SwordsBoss

Inherited Members

[Zombie.ReadOnlyInstances](#) , [Zombie.Id](#) , [Zombie.type](#) , [Zombie.Transform](#) , [Zombie.AttackRadius](#) ,
[Zombie.Speed](#) , [Zombie.health](#) , [Zombie.SetupSubscriptions\(\)](#) , [Zombie.GenerateInstances\(uint\)](#) ,
[Zombie.GenerateInstancesFast\(uint\)](#) , [Zombie.GenerateInstancesExplosion\(uint\)](#) ,
[Zombie.GenerateInstancesBoss\(uint\)](#) , [Zombie.GenerateInstancesSwords\(uint\)](#) ,
[Zombie.UpdateInstances\(\)](#) , [Zombie.FindNearestPlayer\(\)](#) , [Zombie.GetZombieById\(int\)](#) ,
[Zombie.RemoveZombie\(Zombie\)](#) , [Zombie.Damage\(int\)](#) , [object.Equals\(object\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

SwordsBoss()

```
public SwordsBoss()
```

Properties

MaxHealth

```
public override int MaxHealth { get; }
```

Property Value

Methods

Update()

Updates the SwordsBoss state, including rotation, extension timing, and damage checking.

```
protected override void Update()
```

Class Wall

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a wall in the game and manages wall instances and board generation.

```
public class Wall
```

Inheritance

[object](#) ← Wall

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Position

```
public required Vector2 Position { get; init; }
```

Property Value

[Vector2](#)

Methods

AddInstances(IEnumerable<Wall>)

```
public static void AddInstances(IEnumerable<Wall> instances)
```

Parameters

instances [IEnumerable](#)<[Wall](#)>

ClearInstances()

```
public static void ClearInstances()
```

GenerateBoard(int, int, int)

Generates a game board with walls, including outer walls and random inner walls.

```
public static void GenerateBoard(int width, int height, int numWalls)
```

Parameters

`width` [int](#)

The width of the game board.

`height` [int](#)

The height of the game board.

`numWalls` [int](#)

The number of random inner walls to generate.

SetupSubscriptions()

Sets up subscriptions for wall-related network events.

```
public static void SetupSubscriptions()
```

Class WaveController

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Controls the waves of enemies in the game, including initialization, spawning, and updating waves.

```
public class WaveController
```

Inheritance

[object](#) ← WaveController

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

Initialize(int, int)

Initializes the wave controller with the specified map dimensions and sets up spawn points for the zombies.

```
public static void Initialize(int width, int height)
```

Parameters

width [int](#)

The width of the map in grid cells.

height [int](#)

The height of the map in grid cells.

Update()

```
public static void Update()
```

Events

WaveStarted

```
public static event Action<int>? WaveStarted
```

Event Type

[Action](#) <[int](#)>

Class Zombie

Namespace: [WaveZ.Server](#)

Assembly: server.dll

Represents a zombie in the game, with various types of zombies inheriting from this class, on the server-side. Manages zombie instances, spawning, updating, and handling interactions with players.

```
public class Zombie
```

Inheritance

[object](#) ← Zombie

Derived

[BossZombie](#), [ExplosionZombie](#), [SwordsBoss](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Fields

AttackRadius

```
protected float AttackRadius
```

Field Value

[float](#)

Id

```
public required int Id
```

Field Value

[int](#)↗

Speed

`public float Speed`

Field Value

[float](#)↗

Transform

`public required Transform2D Transform`

Field Value

[Transform2D](#)

health

`public int health`

Field Value

[int](#)↗

type

`public required string type`

Field Value

[string](#)↗

Properties

MaxHealth

```
public virtual int MaxHealth { get; }
```

Property Value

[int ↗](#)

ReadOnlyInstances

```
public static IReadonlyCollection<Zombie> ReadOnlyInstances { get; }
```

Property Value

[IReadonlyCollection ↗ <Zombie>](#)

Methods

Damage(int)

```
public virtual void Damage(int amount)
```

Parameters

amount [int ↗](#)

FindNearestPlayer()

Finds the nearest player to the zombie.

```
public Player? FindNearestPlayer()
```

Returns

[Player](#)

The nearest player if found; otherwise, null.

GenerateInstances(uint)

```
public static List<Zombie> GenerateInstances(uint count)
```

Parameters

`count` [uint](#)

Returns

[List](#)<[Zombie](#)>

GenerateInstancesBoss(uint)

```
public static List<Zombie> GenerateInstancesBoss(uint count)
```

Parameters

`count` [uint](#)

Returns

[List](#)<[Zombie](#)>

GenerateInstancesExplosion(uint)

```
public static List<Zombie> GenerateInstancesExplosion(uint count)
```

Parameters

count [uint](#)

Returns

[List](#)<[Zombie](#)>

GenerateInstancesFast(uint)

```
public static List<Zombie> GenerateInstancesFast(uint count)
```

Parameters

count [uint](#)

Returns

[List](#)<[Zombie](#)>

GenerateInstancesSwords(uint)

```
public static List<Zombie> GenerateInstancesSwords(uint count)
```

Parameters

count [uint](#)

Returns

[List](#)<[Zombie](#)>

GetZombieById(int)

```
public static Zombie? GetZombieById(int id)
```

Parameters

`id int`

Returns

[Zombie](#)

RemoveZombie(Zombie)

```
public static void RemoveZombie(Zombie zombie)
```

Parameters

`zombie Zombie`

SetupSubscriptions()

```
public static void SetupSubscriptions()
```

Update()

```
protected virtual void Update()
```

UpdateInstances()

```
public static void UpdateInstances()
```

Namespace WaveZ.Shared

Classes

[UpdatePlayerHealthPacket](#)

Class UpdatePlayerHealthPacket

Namespace: [WaveZ.Shared](#)

Assembly: shared.dll

```
public class UpdatePlayerHealthPacket
```

Inheritance

[object](#) ← UpdatePlayerHealthPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Health

```
public float Health { get; set; }
```

Property Value

[float](#)

Namespace WaveZ.Shared.Networking

Classes

[BarrelDestroyedPacket](#)

[BarrelExplodePacket](#)

[BarrelPlacedPacket](#)

[BossProjectileRemovedPacket](#)

[BossProjectileSpawnedPacket](#)

[BossProjectileUpdatedPacket](#)

[BulletDamagePacket](#)

[BulletUpdatePacket](#)

[CreateMapPacket](#)

[MapCreatedPacket](#)

[PlaceBarrelPacket](#)

[PlayerHealthUpdatedPacket](#)

[PlayerJoinAcceptedPacket](#)

[PlayerJoinAcceptedPacket.ConnectedPlayer](#)

[PlayerJoinPacket](#)

[PlayerJoinedPacket](#)

[PlayerSkinWeaponUpdatePacket](#)

[RemoveBulletPacket](#)

[SwordsBossUpdatePacket](#)

[TransformUpdatePacket](#)

[UpdatedBarrelExplodePacket](#)

[UpdatedBulletPacket](#)

[UpdatedDestroyedBarrelPacket](#)

[UpdatedPlayerSkinWeaponPacket](#)

[UpdatedRemoveBulletPacket](#)

[UpdatedTransformPacket](#)

[UpdatedVectorPacket](#)

[VectorUpdatePacket](#)

[WallPacket](#)

[WaveStartedPacket](#)

[ZombieDiedPacket](#)

[ZombieExplodedPacket](#)

[ZombieExplosionCompletePacket](#)

[ZombieHealthUpdatedPacket](#)

[ZombieSpawnedPacket](#)

[ZombieTransformUpdatedPacket](#)

Class BarrelDestroyedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BarrelDestroyedPacket
```

Inheritance

[object](#) ← BarrelDestroyedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class BarrelExplodePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BarrelExplodePacket
```

Inheritance

[object](#) ← BarrelExplodePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class BarrelPlacedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BarrelPlacedPacket
```

Inheritance

[object](#) ← BarrelPlacedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Class BossProjectileRemovedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BossProjectileRemovedPacket
```

Inheritance

[object](#) ← BossProjectileRemovedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class BossProjectileSpawnedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BossProjectileSpawnedPacket
```

Inheritance

[object](#) ← BossProjectileSpawnedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Direction

```
public Vector2 Direction { get; set; }
```

Property Value

[Vector2](#)

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#) ↗

Class BossProjectileUpdatedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BossProjectileUpdatedPacket
```

Inheritance

[object](#) ← BossProjectileUpdatedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Class BulletDamagePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BulletDamagePacket
```

Inheritance

[object](#) ← BulletDamagePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Damage

```
public int Damage { get; set; }
```

Property Value

[int](#)

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class BulletUpdatePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class BulletUpdatePacket
```

Inheritance

[object](#) ← BulletUpdatePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Class CreateMapPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class CreateMapPacket
```

Inheritance

[object](#) ← CreateMapPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

WallPositions

```
public Vector2[] WallPositions { get; set; }
```

Property Value

[Vector2](#)[]

Class MapCreatedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class MapCreatedPacket
```

Inheritance

[object](#) ← MapCreatedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

WallPositions

```
public Vector2[] WallPositions { get; set; }
```

Property Value

[Vector2](#)[]

Class PlaceBarrelPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlaceBarrelPacket
```

Inheritance

[object](#) ← PlaceBarrelPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Class PlayerHealthUpdatedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlayerHealthUpdatedPacket
```

Inheritance

[object](#) ← PlayerHealthUpdatedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Health

```
public float Health { get; set; }
```

Property Value

[float](#)

PeerId

```
public int PeerId { get; set; }
```

Property Value

[int](#)

Class PlayerJoinAcceptedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlayerJoinAcceptedPacket
```

Inheritance

[object](#) ← PlayerJoinAcceptedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

ConnectedPlayers

```
public PlayerJoinAcceptedPacket.ConnectedPlayer[]? ConnectedPlayers { get; set; }
```

Property Value

[ConnectedPlayer\[\]](#)

PeerId

```
public int PeerId { get; set; }
```

Property Value

[int](#)

Class PlayerJoinAcceptedPacket.ConnectedPlayer

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlayerJoinAcceptedPacket.ConnectedPlayer
```

Inheritance

[object](#) ← PlayerJoinAcceptedPacket.ConnectedPlayer

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

PeerId

```
public int PeerId { get; set; }
```

Property Value

[int](#)

Skin

```
public int Skin { get; set; }
```

Property Value

[int](#)

Username

```
public string Username { get; set; }
```

Property Value

[string](#) ↗

Class PlayerJoinPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlayerJoinPacket
```

Inheritance

[object](#) ← PlayerJoinPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

SelectedSkin

```
public int SelectedSkin { get; set; }
```

Property Value

[int](#)

Username

```
public string Username { get; set; }
```

Property Value

[string](#)

Class PlayerJoinedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlayerJoinedPacket
```

Inheritance

[object](#) ← PlayerJoinedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

PeerId

```
public int PeerId { get; set; }
```

Property Value

[int](#)

SelectedSkin

```
public int SelectedSkin { get; set; }
```

Property Value

[int](#)

Username

```
public string Username { get; set; }
```

Property Value

[string](#) ↗

Class PlayerSkinWeaponUpdatePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class PlayerSkinWeaponUpdatePacket
```

Inheritance

[object](#) ← PlayerSkinWeaponUpdatePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Skin

```
public int Skin { get; set; }
```

Property Value

[int](#)

WeaponIndex

```
public int WeaponIndex { get; set; }
```

Property Value

[int](#)

Class RemoveBulletPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class RemoveBulletPacket
```

Inheritance

[object](#) ← RemoveBulletPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class SwordsBossUpdatePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class SwordsBossUpdatePacket
```

Inheritance

[object](#) ← SwordsBossUpdatePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

IsExtended

```
public bool IsExtended { get; set; }
```

Property Value

[bool](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#) ↗

Rotation

```
public float Rotation { get; set; }
```

Property Value

[float](#) ↗

Class TransformUpdatePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class TransformUpdatePacket
```

Inheritance

[object](#) ← TransformUpdatePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Transform

```
public Transform2D Transform { get; set; }
```

Property Value

[Transform2D](#)

Class UpdatedBarrelExplodePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedBarrelExplodePacket
```

Inheritance

[object](#) ← UpdatedBarrelExplodePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class UpdatedBulletPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedBulletPacket
```

Inheritance

[object](#) ← UpdatedBulletPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Velocity

```
public Vector2 Velocity { get; set; }
```

Property Value

[Vector2](#) ↗

Class UpdatedDestroyedBarrelPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedDestroyedBarrelPacket
```

Inheritance

[object](#) ← UpdatedDestroyedBarrelPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class UpdatedPlayerSkinWeaponPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedPlayerSkinWeaponPacket
```

Inheritance

[object](#) ← UpdatedPlayerSkinWeaponPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

PeerId

```
public int PeerId { get; set; }
```

Property Value

[int](#)

Skin

```
public int Skin { get; set; }
```

Property Value

[int](#)

WeaponIndex

```
public int WeaponIndex { get; set; }
```

Property Value

[int ↗](#)

Class UpdatedRemoveBulletPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedRemoveBulletPacket
```

Inheritance

[object](#) ← UpdatedRemoveBulletPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class UpdatedTransformPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedTransformPacket
```

Inheritance

[object](#) ← UpdatedTransformPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

PeerId

```
public int PeerId { get; set; }
```

Property Value

[int](#)

Transform

```
public Transform2D Transform { get; set; }
```

Property Value

[Transform2D](#)

Class UpdatedVectorPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class UpdatedVectorPacket
```

Inheritance

[object](#) ← UpdatedVectorPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Direction

```
public Vector2 Direction { get; set; }
```

Property Value

[Vector2](#)

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#) ↗

Class VectorUpdatePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class VectorUpdatePacket
```

Inheritance

[object](#) ← VectorUpdatePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Direction

```
public Vector2 Direction { get; set; }
```

Property Value

[Vector2](#)

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Class WallPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class WallPacket
```

Inheritance

[object](#) ← WallPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Position

```
public Vector2 Position { get; set; }
```

Property Value

[Vector2](#)

Class WaveStartedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class WaveStartedPacket
```

Inheritance

[object](#) ← WaveStartedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

WaveNumber

```
public int WaveNumber { get; set; }
```

Property Value

[int](#)

Class ZombieDiedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class ZombieDiedPacket
```

Inheritance

[object](#) ← ZombieDiedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class ZombieExplodedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class ZombieExplodedPacket
```

Inheritance

[object](#) ← ZombieExplodedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class ZombieExplosionCompletePacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class ZombieExplosionCompletePacket
```

Inheritance

[object](#) ← ZombieExplosionCompletePacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class ZombieHealthUpdatedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public class ZombieHealthUpdatedPacket
```

Inheritance

[object](#) ← ZombieHealthUpdatedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Health

```
public int Health { get; set; }
```

Property Value

[int](#)

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Class ZombieSpawnedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public sealed class ZombieSpawnedPacket
```

Inheritance

[object](#) ← ZombieSpawnedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Type

```
public string Type { get; set; }
```

Property Value

[string](#)

Class ZombieTransformUpdatedPacket

Namespace: [WaveZ.Shared.Networking](#)

Assembly: shared.dll

```
public sealed class ZombieTransformUpdatedPacket
```

Inheritance

[object](#) ← ZombieTransformUpdatedPacket

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

Id

```
public int Id { get; set; }
```

Property Value

[int](#)

Transform

```
public Transform2D Transform { get; set; }
```

Property Value

[Transform2D](#)

Namespace WaveZ.Shared.Transforms

Structs

[Transform2D](#)

Represents a position and rotation in 2D space.

Struct Transform2D

Namespace: [WaveZ.Shared.Transforms](#)

Assembly: shared.dll

Represents a position and rotation in 2D space.

```
public struct Transform2D : INetSerializable
```

Implements

INetSerializable

Inherited Members

[ValueType.Equals\(object\)](#) , [ValueType.GetHashCode\(\)](#) , [ValueType.ToString\(\)](#) ,
[object.Equals\(object, object\)](#) , [object.GetType\(\)](#) , [object.ReferenceEquals\(object, object\)](#)

Constructors

Transform2D()

Represents a position and rotation in 2D space.

```
public Transform2D()
```

Fields

Position

```
public Vector2 Position
```

Field Value

[Vector2](#)

Rotation

```
public float Rotation
```

Field Value

[float ↗](#)

Methods

Deserialize(NetDataReader)

```
public void Deserialize(NetDataReader reader)
```

Parameters

reader NetDataReader

Serialize(NetDataWriter)

```
public void Serialize(NetDataWriter writer)
```

Parameters

writer NetDataWriter

Namespace WaveZ.Test

Classes

[Elapsed Time](#)

Represents elapsed time since start of test and start of frame

[TestChain](#)

Represents collection of tests. Fails, if one test fails; otherwise succeeds.

[TestMap](#)

Represents state needed for test, can be sent to server.

[TestResult](#)

Represents result of completed test as algebraic data type.

[TestResult.Fail](#)

[TestResult.Success](#)

Interfaces

[ITest](#)

Represents a test case. Has to be setup using [Setup\(int, int\)](#).

Class ElapsedTime

Namespace: [WaveZ.Test](#)

Assembly: test.dll

Represents elapsed time since start of test and start of frame

```
public record ElapsedTime : IEquatable<ElapsedTime>
```

Inheritance

[object](#) ← ElapsedTime

Implements

[IEquatable](#)<[ElapsedTime](#)>

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

ElapsedTime(float, float)

Represents elapsed time since start of test and start of frame

```
public ElapsedTime(float Total, float Frame)
```

Parameters

Total [float](#)

Frame [float](#)

Properties

Frame

```
public float Frame { get; init; }
```

Property Value

[float](#) ↗

Total

```
public float Total { get; init; }
```

Property Value

[float](#) ↗

Interface ITest

Namespace: [WaveZ.Test](#)

Assembly: test.dll

Represents a test case. Has to be setup using [Setup\(int, int\)](#).

```
public interface ITest
```

Methods

Draw()

```
void Draw()
```

Setup(int, int)

```
void Setup(int screenWidth, int screenHeight)
```

Parameters

screenWidth [int](#)

screenHeight [int](#)

Update(ElapsedTime)

```
TestResult? Update(ElapsedTime time)
```

Parameters

time [ElapsedTime](#)

Returns

[TestResult](#)

TestResult, if test completed; else **null**

Class TestChain

Namespace: [WaveZ.Test](#)

Assembly: test.dll

Represents collection of tests. Fails, if one test fails; otherwise succeeds.

```
public sealed class TestChain : ITest
```

Inheritance

[object](#) ← TestChain

Implements

[ITest](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

TestChain(IEnumerable<ITest>)

Represents collection of tests. Fails, if one test fails; otherwise succeeds.

```
public TestChain(IEnumerable<ITest> tests)
```

Parameters

tests [IEnumerable](#)<[ITest](#)>

Methods

Draw()

```
public void Draw()
```

Setup(int, int)

```
public void Setup(int screenWidth, int screenHeight)
```

Parameters

screenWidth [int](#)

screenHeight [int](#)

Update(ElapsedTime)

```
public TestResult? Update(ElapsedTime time)
```

Parameters

time [ElapsedTime](#)

Returns

[TestResult](#)

[TestResult](#), if test completed; else [null](#)

Class TestMap

Namespace: [WaveZ.Test](#)

Assembly: test.dll

Represents state needed for test, can be sent to server.

```
public sealed class TestMap
```

Inheritance

[object](#) ← TestMap

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Properties

WallPosition

```
public IEnumerable<Vector2> WallPosition { get; init; }
```

Property Value

[IEnumerable](#) <[Vector2](#)>

Methods

Submit(Action?)

Send state to server. onReady is called when state is ready on server.

```
public void Submit(Action? onReady)
```

Parameters

onReady [Action](#)

Class TestResult

Namespace: [WaveZ.Test](#)

Assembly: test.dll

Represents result of completed test as algebraic data type.

```
public record TestResult : IEquatable<TestResult>
```

Inheritance

[object](#) ← TestResult

Implements

[IEquatable](#) <[TestResult](#)>

Derived

[TestResult.Fail](#), [TestResult.Success](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Class TestResult.Fail

Namespace: [WaveZ.Test](#)

Assembly: test.dll

```
public record TestResult.Fail : TestResult, IEquatable<TestResult>,
IEquatable<TestResult.Fail>
```

Inheritance

[object](#) ← [TestResult](#) ← TestResult.Fail

Implements

[IEquatable](#)<[TestResult](#)>, [IEquatable](#)<[TestResult.Fail](#)>

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Constructors

Fail(string)

```
public Fail(string Reason)
```

Parameters

Reason [string](#)

Properties

Reason

```
public string Reason { get; init; }
```

Property Value

[string](#) ↗

Class TestResult.Success

Namespace: [WaveZ.Test](#)

Assembly: test.dll

```
public record TestResult.Success : TestResult, IEquatable<TestResult>,
IEquatable<TestResult.Success>
```

Inheritance

[object](#) ← [TestResult](#) ← TestResult.Success

Implements

[IEquatable](#)<[TestResult](#)>, [IEquatable](#)<[TestResult.Success](#)>

Inherited Members

[object.Equals\(object\)](#), [object.Equals\(object, object\)](#), [object.GetHashCode\(\)](#), [object.GetType\(\)](#),
[object.MemberwiseClone\(\)](#), [object.ReferenceEquals\(object, object\)](#), [object.ToString\(\)](#)

Namespace WaveZ.Test.PlayerTargeting

Classes

[Orders](#)

Collection of different kinds of orders.

[PlayerOrderTargeting](#)

Provides movement to player using a queue of orders.

Interfaces

[IOrder](#)

Provides movement until completed.

Interface IOrder

Namespace: [WaveZ.Test.PlayerTargeting](#)

Assembly: test.dll

Provides movement until completed.

```
public interface IOrder
```

Methods

GetTargetMovement(Player)

```
Vector2 GetTargetMovement(Player player)
```

Parameters

player [Player](#)

Returns

[Vector2](#)

IsCompleted(Player)

```
bool IsCompleted(Player player)
```

Parameters

player [Player](#)

Returns

[bool](#)

Class Orders

Namespace: [WaveZ.Test.PlayerTargeting](#)

Assembly: test.dll

Collection of different kinds of orders.

```
public static class Orders
```

Inheritance

[object](#) ← Orders

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

MoveTo(Vector2)

Move to [position](#) until distance is less than [0.001](#).

```
public static IOrder MoveTo(Vector2 position)
```

Parameters

[position](#) [Vector2](#)

Returns

[IOrder](#)

Class PlayerOrderTargeting

Namespace: [WaveZ.Test.PlayerTargeting](#)

Assembly: test.dll

Provides movement to player using a queue of orders.

```
public sealed class PlayerOrderTargeting : IPlayerTargeting
```

Inheritance

[object](#) ← PlayerOrderTargeting

Implements

[IPlayerTargeting](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Constructors

PlayerOrderTargeting(IEnumerable<IOrder>)

Provides movement to player using a queue of orders.

```
public PlayerOrderTargeting(IEnumerable<IOrder> orders)
```

Parameters

orders [IEnumerable](#)<[IOrder](#)>

Methods

GetTargetMovement(Player)

Gets the target movement direction for the player.

```
public Vector2 GetTargetMovement(Player player)
```

Parameters

player [Player](#)

The player for whom the target movement is being calculated.

Returns

[Vector2](#)

A normalized vector containing the direction the player should move to.

Namespace WaveZ.Test.Tests

Classes

[PlayerMovementTest](#)

[PlayerWallCollisionTest](#)

Class PlayerMovementTest

Namespace: [WaveZ.Test.Tests](#)

Assembly: test.dll

```
public class PlayerMovementTest : ITest
```

Inheritance

[object](#) ← PlayerMovementTest

Implements

[ITest](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

Draw()

```
public void Draw()
```

Setup(int, int)

```
public void Setup(int screenWidth, int screenHeight)
```

Parameters

screenWidth [int](#)

screenHeight [int](#)

Update(ElapsedTime)

```
public TestResult? Update(ElapsedTime time)
```

Parameters

time [ElapsedTime](#)

Returns

[TestResult](#)

TestResult, if test completed; else **null**

Class PlayerWallCollisionTest

Namespace: [WaveZ.Test.Tests](#)

Assembly: test.dll

```
public class PlayerWallCollisionTest : ITest
```

Inheritance

[object](#) ← PlayerWallCollisionTest

Implements

[ITest](#)

Inherited Members

[object.Equals\(object\)](#) , [object.Equals\(object, object\)](#) , [object.GetHashCode\(\)](#) , [object.GetType\(\)](#) ,
[object.MemberwiseClone\(\)](#) , [object.ReferenceEquals\(object, object\)](#) , [object.ToString\(\)](#)

Methods

Draw()

```
public void Draw()
```

Setup(int, int)

```
public void Setup(int screenWidth, int screenHeight)
```

Parameters

screenWidth [int](#)

screenHeight [int](#)

Update(ElapsedTime)

```
public TestResult? Update(ElapsedTime time)
```

Parameters

time [ElapsedTime](#)

Returns

[TestResult](#)

TestResult, if test completed; else **null**