Hi there! This document serves as the documentation for all the code I've written for the programming challenge.

My main source of design influence was the Unity Game Engine. Similar to how Components are attached to GameObjects that are placed in a Unity Scene, in my engine Attachments are attached to Entities, which are then placed in a Level.

What follows is an automatically generated documentation file that outlines and explains the object oriented structure behind my code. The explanations for the methods and classes have been generated using the Javadoc style comments I've written throughout my code files.

Ubisoft Next - Oktay Comu

0.0

Generated by Doxygen 1.8.18

1 Hierarchical Index 1

1.1 Class Hierarchy . . . 1

2 Class Index 3

2.1 Class List . . . 3

3 Class Documentation 5

3.1 Attachment Class Reference . . . 5

3.1.1 Detailed Description . . . 5

3.1.2 Constructor & Destructor Documentation . . . 6

3.1.2.1 Attachment() . . . 6

3.1.3 Member Function Documentation . . . 6

3.1.3.1 GetEntity() . . . 6

3.1.3.2 Init() . . . 6

3.1.3.3 IsActive() . . . 7

3.1.3.4 OnCollision() . . . 7

3.1.3.5 SetActive() . . . 7

3.1.3.6 Update() . . . 7

3.2 BulletEntity Class Reference . . . 8

3.2.1 Detailed Description . . . 8

3.2.2 Constructor & Destructor Documentation . . . 8

3.2.2.1 BulletEntity() . . . 8

3.3 BulletTraveler Class Reference . . . 9

3.3.1 Detailed Description . . . 9

3.3.2 Constructor & Destructor Documentation . . . 9

3.3.2.1 BulletTraveler() . . . 9

3.3.3 Member Function Documentation . . . 9

3.3.3.1 Update() . . . 10

3.4 CircleCollider Class Reference . . . 10

3.4.1 Detailed Description . . . 10

3.4.2 Constructor & Destructor Documentation . . . 11

3.4.2.1 CircleCollider() . . . 11

3.4.3 Member Function Documentation . . . 11

3.4.3.1 CollidesWith() . . . 11

3.4.4 Member Data Documentation . . . 11

3.4.4.1 radius . . . 11

3.5 Collider Class Reference . . . 12

3.5.1 Detailed Description . . . 12

3.5.2 Constructor & Destructor Documentation . . . 12

3.5.2.1 Collider() . . . 12

3.5.3 Member Function Documentation . . . 12

3.5.3.1 CollidesWith() . . . 12

3.6 ColorableTraveler Class Reference . . . 13

3.6.1 Detailed Description . . . 13
3.6.2 Constructor & Destructor Documentation . . . 13
3.6.2.1 ColorableTraveler() . . . 13
3.6.3 Member Function Documentation . . . 14
3.6.3.1 Update() . . . 14
3.6.4 Member Data Documentation . . . 14
3.6.4.1 colorType . . . 14
3.7 ColorType Class Reference . . . 14
3.8 EnemyEntity Class Reference . . . 15
3.8.1 Detailed Description . . . 15
3.8.2 Constructor & Destructor Documentation . . . 15
3.8.2.1 EnemyEntity() . . . 15
3.9 EnemyRenderer Class Reference . . . 16
3.9.1 Detailed Description . . . 16
3.9.2 Constructor & Destructor Documentation . . . 16
3.9.2.1 EnemyRenderer() . . . 16
3.9.3 Member Function Documentation . . . 16
3.9.3.1 Draw() . . . 17
3.10 EnemySpawner Class Reference . . . 17
3.10.1 Detailed Description . . . 17
3.10.2 Constructor & Destructor Documentation . . . 17
3.10.2.1 EnemySpawner() . . . 17
3.10.3 Member Function Documentation . . . 18
3.10.3.1 Init() . . . 18
3.10.3.2 SpawnEnemy() . . . 18
3.10.3.3 SpawnWall() . . . 18
3.10.3.4 Update() . . . 18
3.10.4 Member Data Documentation . . . 19
3.10.4.1 enemySpawnInterval . . . 19
3.10.4.2 wallSpawnInterval . . . 19
3.11 EnemyTraveler Class Reference . . . 19
3.11.1 Detailed Description . . . 20
3.11.2 Constructor & Destructor Documentation . . . 20
3.11.2.1 EnemyTraveler() . . . 20
3.11.3 Member Function Documentation . . . 20
3.11.3.1 OnCollision() . . . 20
3.11.3.2 ShootBullet() . . . 20
3.11.3.3 Update() . . . 21
3.12 Entity Class Reference . . . 21
3.12.1 Detailed Description . . . 22
3.12.2 Constructor & Destructor Documentation . . . 22
3.12.2.1 Entity() . . . 22

3.12.2.2 ~Entity() . . . 22
3.12.3 Member Function Documentation . . . 22
3.12.3.1 AddAttachment() . . . 22
3.12.3.2 GetAttachment() . . . 22
3.12.3.3 GetAttachments() . . . 23
3.12.3.4 Init() . . . 23
3.12.3.5 IsActive() . . . 23
3.12.3.6 OnCollision() . . . 23
3.12.3.7 Render() . . . 23
3.12.3.8 SetActive() . . . 24
3.12.3.9 Update() . . . 24
3.13 GameController Class Reference . . . 24
3.13.1 Detailed Description . . . 25
3.13.2 Constructor & Destructor Documentation . . . 25
3.13.2.1 GameController() . . . 25
3.13.3 Member Function Documentation . . . 25
3.13.3.1 EndCurrentGame() . . . 25
3.13.3.2 EndGame() . . . 25
3.13.3.3 IncrementScore() . . . 26
3.13.3.4 Init() . . . 26
3.13.3.5 SpawnPlayer() . . . 26
3.13.3.6 StartGame() . . . 26
3.13.3.7 Update() . . . 26
3.13.4 Member Data Documentation . . . 26
3.13.4.1 currentController . . . 27
3.14 Level Class Reference . . . 27
3.14.1 Detailed Description . . . 27
3.14.2 Constructor & Destructor Documentation . . . 27
3.14.2.1 Level() . . . 28
3.14.2.2 ~Level() . . . 28
3.14.3 Member Function Documentation . . . 28
3.14.3.1 AddEntity() . . . 28
3.14.3.2 DestroyEntity() . . . 28
3.14.3.3 FindAttachment() . . . 28
3.14.3.4 Init() . . . 29
3.14.3.5 MarkForDestruction() . . . 29
3.14.3.6 RemoveEntity() . . . 29
3.14.3.7 Render() . . . 29
3.14.3.8 SpawnEntity() . . . 29
3.14.3.9 Update() . . . 30
3.14.4 Member Data Documentation . . . 30
3.14.4.1 currentLevel . . . 30

3.15 LineRenderer Class Reference . . . 30
3.15.1 Detailed Description . . . 31
3.15.2 Constructor & Destructor Documentation . . . 31
3.15.2.1 LineRenderer() . . . 31
3.15.3 Member Function Documentation . . . 31
3.15.3.1 Draw() . . . 31
3.16 LinkedList< T > Class Template Reference . . . 31
3.16.1 Detailed Description . . . 32
3.16.2 Constructor & Destructor Documentation . . . 32
3.16.2.1 LinkedList() . . . 32
3.16.2.2 ~LinkedList() . . . 32
3.16.3 Member Function Documentation . . . 32
3.16.3.1 Add() . . . 32
3.16.3.2 Contains() . . . 33
3.16.3.3 Get() . . . 33
3.16.3.4 IsEmpty() . . . 33
3.16.3.5 Iterator() . . . 34
3.16.3.6 Remove() . . . 34
3.16.3.7 Size() . . . 34
3.17 LinkedListIterator< T > Class Template Reference . . . 34
3.17.1 Detailed Description . . . 35
3.17.2 Constructor & Destructor Documentation . . . 35
3.17.2.1 LinkedListIterator() . . . 35
3.17.3 Member Function Documentation . . . 35
3.17.3.1 HasNext() . . . 35
3.17.3.2 Next() . . . 36
3.17.3.3 Reset() . . . 36
3.18 node< T > Struct Template Reference . . . 36
3.19 PlayerEntity Class Reference . . . 36
3.19.1 Detailed Description . . . 37
3.19.2 Constructor & Destructor Documentation . . . 37
3.19.2.1 PlayerEntity() . . . 37
3.20 PlayerRenderer Class Reference . . . 37
3.20.1 Detailed Description . . . 37
3.20.2 Constructor & Destructor Documentation . . . 38
3.20.2.1 PlayerRenderer() . . . 38
3.20.3 Member Function Documentation . . . 38
3.20.3.1 Draw() . . . 38
3.21 PlayerTraveler Class Reference . . . 38
3.21.1 Detailed Description . . . 39
3.21.2 Constructor & Destructor Documentation . . . 39
3.21.2.1 PlayerTraveler() . . . 39

3.21.3 Member Function Documentation . . . 39
3.21.3.1 IsMouthOpen() . . . 39
3.21.3.2 MakeImmune() . . . 39
3.21.3.3 OnCollision() . . . 40
3.21.3.4 ShootBullet() . . . 40
3.21.3.5 TunnelUpdate() . . . 40
3.21.3.6 Update() . . . 40
3.22 RegularPolygonRenderer Class Reference . . . 41
3.22.1 Detailed Description . . . 41
3.22.2 Constructor & Destructor Documentation . . . 41
3.22.2.1 RegularPolygonRenderer() . . . 41
3.22.3 Member Function Documentation . . . 42
3.22.3.1 Draw() . . . 42
3.22.4 Member Data Documentation . . . 42
3.22.4.1 radius . . . 42
3.22.4.2 vertexCount . . . 42
3.23 Renderer Class Reference . . . 42
3.23.1 Detailed Description . . . 43
3.23.2 Constructor & Destructor Documentation . . . 43
3.23.2.1 Renderer() . . . 43
3.23.3 Member Function Documentation . . . 43
3.23.3.1 Draw() . . . 43
3.23.3.2 Init() . . . 44
3.23.3.3 Update() . . . 44
3.23.4 Member Data Documentation . . . 44
3.23.4.1 r . . . 44
3.24 TextRenderer Class Reference . . . 44
3.24.1 Detailed Description . . . 45
3.24.2 Constructor & Destructor Documentation . . . 45
3.24.2.1 TextRenderer() . . . 45
3.24.3 Member Function Documentation . . . 45
3.24.3.1 Draw() . . . 45
3.24.4 Member Data Documentation . . . 45
3.24.4.1 position . . . 46
3.24.4.2 text . . . 46
3.25 Transform Class Reference . . . 46
3.25.1 Constructor & Destructor Documentation . . . 46
3.25.1.1 Transform() . . . 46
3.25.2 Member Function Documentation . . . 47
3.25.2.1 Init() . . . 47
3.25.2.2 Update() . . . 47
3.25.3 Member Data Documentation . . . 47

3.25.3.1 position . . . 47
3.25.3.2 rotation . . . 47
3.25.3.3 scale . . . 48
3.26 Tunnel Class Reference . . . 48
3.26.1 Detailed Description . . . 48
3.26.2 Constructor & Destructor Documentation . . . 48
3.26.2.1 Tunnel() . . . 48
3.26.2.2 ~Tunnel() . . . 49
3.26.3 Member Function Documentation . . . 49
3.26.3.1 AddTraveler() . . . 49
3.26.3.2 ClearTunnel() . . . 49
3.26.3.3 Init() . . . 49
3.26.3.4 RemoveTraveler() . . . 49
3.26.3.5 Update() . . . 49
3.26.4 Member Data Documentation . . . 50
3.26.4.1 innerRadius . . . 50
3.26.4.2 outerRadius . . . 50
3.26.4.3 spinSpeed . . . 50
3.26.4.4 trackCount . . . 50
3.27 TunnelRenderer Class Reference . . . 51
3.27.1 Detailed Description . . . 51
3.27.2 Constructor & Destructor Documentation . . . 51
3.27.2.1 TunnelRenderer() . . . 51
3.27.3 Member Function Documentation . . . 51
3.27.3.1 Draw() . . . 52
3.27.3.2 Init() . . . 52
3.28 TunnelTraveler Class Reference . . . 52
3.28.1 Detailed Description . . . 53
3.28.2 Constructor & Destructor Documentation . . . 53
3.28.2.1 TunnelTraveler() . . . 53
3.28.3 Member Function Documentation . . . 53
3.28.3.1 Init() . . . 53
3.28.3.2 TunnelUpdate() . . . 53
3.28.3.3 Update() . . . 54
3.28.4 Member Data Documentation . . . 54
3.28.4.1 radius . . . 54
3.28.4.2 track . . . 54
3.28.4.3 velocity . . . 54
3.29 Vector2 Class Reference . . . 55
3.29.1 Detailed Description . . . 55
3.29.2 Constructor & Destructor Documentation . . . 55
3.29.2.1 Vector2() [1/2] . . . 55

3.29.2.2 Vector2() [2/2] . . . 55
3.29.3 Member Function Documentation . . . 56
3.29.3.1 Distance() . . . 56
3.29.4 Member Data Documentation . . . 56
3.29.4.1 x . . . 56
3.30 WallEntity Class Reference . . . 56
3.30.1 Detailed Description . . . 57
3.30.2 Constructor & Destructor Documentation . . . 57
3.30.2.1 WallEntity() . . . 57
3.31 WallRenderer Class Reference . . . 57
3.31.1 Detailed Description . . . 57
3.31.2 Constructor & Destructor Documentation . . . 58
3.31.2.1 WallRenderer() . . . 58
3.31.3 Member Function Documentation . . . 58
3.31.3.1 Draw() . . . 58

Index **59**

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

- Attachment . . . 5
  - Collider . . . 12
    - CircleCollider . . . 10
  - EnemySpawner . . . 17
  - GameController . . . 24
  - Renderer . . . 42
    - EnemyRenderer . . . 16
    - LineRenderer . . . 30
    - PlayerRenderer . . . 37
    - RegularPolygonRenderer . . . 41
    - TextRenderer . . . 44
    - TunnelRenderer . . . 51
    - WallRenderer . . . 57
  - Transform . . . 46
  - Tunnel . . . 48
  - TunnelTraveler . . . 52
    - ColorableTraveler . . . 13
      - BulletTraveler . . . 9
      - EnemyTraveler . . . 19
      - PlayerTraveler . . . 38
- ColorType . . . 14
- Entity . . . 21
  - BulletEntity . . . 8
  - EnemyEntity . . . 15
  - PlayerEntity . . . 36
  - WallEntity . . . 56
- Level . . . 27
- LinkedList< T > . . . 31
- LinkedList< Attachment > . . . 31
- LinkedList< Entity > . . . 31
- LinkedList< TunnelTraveler > . . . 31
- LinkedListIterator< T > . . . 34
- node< T > . . . 36
- node< Attachment > . . . 36
- node< Entity > . . . 36
- node< TunnelTraveler > . . . 36
- Vector2 . . . 55

Chapter 2

Class Index

2.1 Class List

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

Attachment . . . 5
BulletEntity . . . 8
BulletTraveler . . . 9
CircleCollider . . . 10
Collider . . . 12
ColorableTraveler . . . 13
ColorType . . . 14
EnemyEntity . . . 15
EnemyRenderer . . . 16
EnemySpawner . . . 17
EnemyTraveler . . . 19
Entity . . . 21
GameController . . . 24
Level . . . 27
LineRenderer . . . 30
LinkedList< T > . . . 31
LinkedListIterator< T > . . . 34
node< T > . . . 36
PlayerEntity . . . 36
PlayerRenderer . . . 37
PlayerTraveler . . . 38
RegularPolygonRenderer . . . 41
Renderer . . . 42
TextRenderer . . . 44
Transform . . . 46
Tunnel . . . 48
TunnelRenderer . . . 51
TunnelTraveler . . . 52
Vector2 . . . 55
WallEntity . . . 56
WallRenderer . . . 57

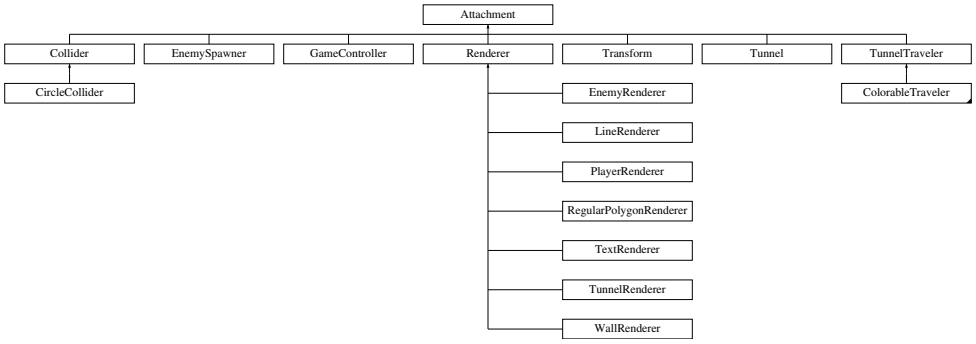
Chapter 3

Class Documentation

3.1 Attachment Class Reference

`#include <Attachment.h>`

Inheritance diagram for Attachment:



Public Member Functions

- Attachment (Entity *ent)
- bool IsActive ()
- void SetActive (bool val)
- Entity * GetEntity ()
- virtual void Init ()
- virtual void Update (float deltaTime)
- virtual void OnCollision (Collider *other)

3.1.1 Detailed Description

Baseclass for anything that can be attached to an Entity. Attachments can be used to define behaviour, controls or rendering options for an Entity.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Attachment()

```
Attachment::Attachment (
            Entity * ent )
```

Constructor for Attachment. Constructs an new Attachment.

Parameters

ent	The Entity to which this Attachment is attached.

3.1.3 Member Function Documentation

3.1.3.1 GetEntity()

```
Entity * Attachment::GetEntity ( )
```

Get the Entity that this Attachment is attached to.

Returns

The Entity that this Attachment is attached to.

3.1.3.2 Init()

```
void Attachment::Init ( )  [virtual]
```

Initialize this Attachment.

Reimplemented in GameController, Tunnel, TunnelTraveler, EnemySpawner, Transform, Renderer, and TunnelRenderer.

3.1.3.3 IsActive()

```
bool Attachment::IsActive ( )
```

Check if Attachment is active.

Returns

true of Attachment is active, false otherwise

3.1.3.4 OnCollision()

```
void Attachment::OnCollision (
            Collider * other )  [virtual]
```

Method to call when the Entity is colliding with another Entity that has a Collider.

Parameters

other	The other Collider with which the Entity collided.

Reimplemented in PlayerTraveler, and EnemyTraveler.

3.1.3.5 SetActive()

```
void Attachment::SetActive (
            bool val )
```

Activate or deactivate Attachment.

Parameters

val	Activate if true, deactivate otherwise.

3.1.3.6 Update()

```
void Attachment::Update (
            float deltaTime )  [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented in GameController, Tunnel, TunnelTraveler, EnemySpawner, PlayerTraveler, Transform, Renderer, ColorableTraveler, EnemyTraveler, and BulletTraveler.

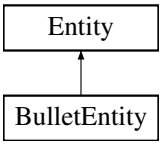
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/Attachment.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/Attachment.cpp

3.2 BulletEntity Class Reference

`#include <BulletEntity.h>`

Inheritance diagram for BulletEntity:



Public Member Functions

- BulletEntity ()

3.2.1 Detailed Description

Preset Bullet Entity.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 BulletEntity()

`BulletEntity::BulletEntity ( )`

Constructor for BulletEntity. Constructs an new BulletEntity.

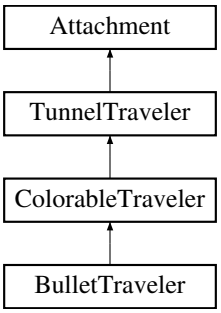
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/BulletEntity.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/BulletEntity.cpp

3.3 BulletTraveler Class Reference

`#include <BulletTraveler.h>`

Inheritance diagram for BulletTraveler:



Public Member Functions

- BulletTraveler (Entity *ent)
- virtual void Update (float deltaTime) override

Additional Inherited Members

3.3.1 Detailed Description

Traveler class representing a bullet.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 BulletTraveler()

```
BulletTraveler::BulletTraveler (
            Entity * ent )  [inline]
```

Constructor for BulletTraveler. Constructs an new BulletTraveler.

Parameters

ent	The Entity to which this BulletTraveler is attached.

3.3.3 Member Function Documentation

3.3.3.1 Update()

```
void BulletTraveler::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from ColorableTraveler.

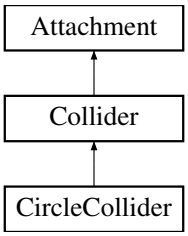
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/BulletTraveler.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/BulletTraveler.cpp

3.4 CircleCollider Class Reference

```
#include <CircleCollider.h>
```

Inheritance diagram for CircleCollider:



Public Member Functions

- CircleCollider (Entity *ent)
- virtual bool CollidesWith (Collider *other) override

Public Attributes

- float radius

3.4.1 Detailed Description

Collider with circular shape that isn't affected by Entity scale.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 CircleCollider()

```
CircleCollider::CircleCollider (
            Entity * ent )  [inline]
```

Constructor for CircleCollider. Constructs an new CircleCollider.

Parameters

ent	The Entity to which this CircleCollider is attached.

3.4.3 Member Function Documentation

3.4.3.1 CollidesWith()

```
bool CircleCollider::CollidesWith (
            Collider * other )  [override], [virtual]
```

Collision check with another Collider.

Parameters

other	The other Collider to check collision with.

Implements Collider.

3.4.4 Member Data Documentation

3.4.4.1 radius

```
float CircleCollider::radius
```

The radius of the circle that describes the CircleCollider

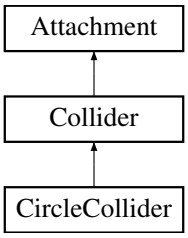
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/colliders/CircleCollider.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/colliders/CircleCollider.↩ cpp

3.5 Collider Class Reference

`#include <Collider.h>`

Inheritance diagram for Collider:



Public Member Functions

- Collider (Entity *ent)
- virtual bool CollidesWith (Collider *other)=0

3.5.1 Detailed Description

The Collider Attachment is used to check collisions with other entites that have Colliders.

3.5.2 Constructor & Destructor Documentation

3.5.2.1 Collider()

```
Collider::Collider (
            Entity * ent )  [inline]
```

Constructor for Collider. Constructs an new Collider.

Parameters

ent	The Entity to which this Collider is attached.

3.5.3 Member Function Documentation

3.5.3.1 CollidesWith()

```
virtual bool Collider::CollidesWith (
            Collider * other )  [pure virtual]
```

Collision check with another Collider.

Parameters

other	The other Collider to check collision with.

Implemented in CircleCollider.

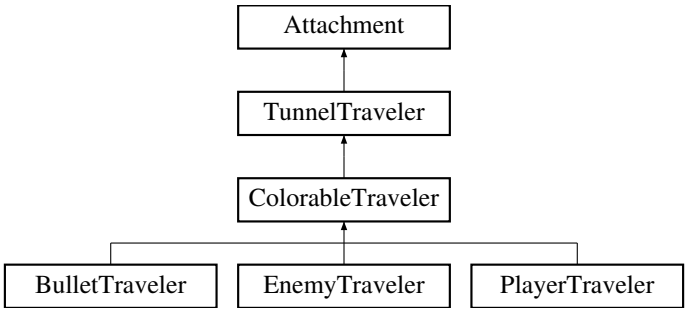
The documentation for this class was generated from the following file:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/colliders/Collider.h

3.6 ColorableTraveler Class Reference

`#include <ColorableTraveler.h>`

Inheritance diagram for ColorableTraveler:



Public Member Functions

- ColorableTraveler (Entity *ent)
- virtual void Update (float deltaTime) override

Public Attributes

- ColorType::Color colorType

3.6.1 Detailed Description

Traveler whose color changes according to its colorType.

3.6.2 Constructor & Destructor Documentation

3.6.2.1 ColorableTraveler()

```
ColorableTraveler::ColorableTraveler (
            Entity * ent )  [inline]
```

Constructor for ColorableTraveler. Constructs an new ColorableTraveler.

Parameters

ent	The Entity to which this ColorableTraveler is attached.

3.6.3 Member Function Documentation

3.6.3.1 Update()

```
void ColorableTraveler::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from TunnelTraveler.

Reimplemented in PlayerTraveler, EnemyTraveler, and BulletTraveler.

3.6.4 Member Data Documentation

3.6.4.1 colorType

```
ColorType::Color ColorableTraveler::colorType
```

The color type of this ColorableTraveler. The color type determines the color in which the Entity is rendered and which Entities it can interact with.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/ColorableTraveler.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/ColorableTraveler.cpp

3.7 ColorType Class Reference

Public Types

- enum **Color** { **red**, **green**, **blue**, **white** }

Static Public Member Functions

- static Color **RandomColor** ()

Static Public Attributes

- static const int **colorCount** = 3

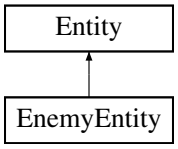
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/ColorType.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/ColorType.cpp

3.8 EnemyEntity Class Reference

`#include <EnemyEntity.h>`

Inheritance diagram for EnemyEntity:



Public Member Functions

- EnemyEntity ()

3.8.1 Detailed Description

Preset Enemy Entity.

3.8.2 Constructor & Destructor Documentation

3.8.2.1 EnemyEntity()

`EnemyEntity::EnemyEntity ( )`

Constructor for EnemyEntity. Constructs an new EnemyEntity.

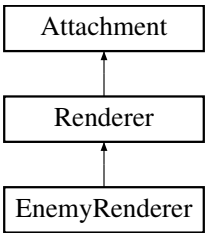
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/EnemyEntity.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/EnemyEntity.cpp

3.9 EnemyRenderer Class Reference

`#include <EnemyRenderer.h>`

Inheritance diagram for EnemyRenderer:



Public Member Functions

- EnemyRenderer (Entity *ent)
- virtual void Draw () override

Additional Inherited Members

3.9.1 Detailed Description

Renderer that is used to display enemy ships.

3.9.2 Constructor & Destructor Documentation

3.9.2.1 EnemyRenderer()

```
EnemyRenderer::EnemyRenderer (
            Entity * ent )  [inline]
```

Constructor for EnemyRenderer. Constructs an new EnemyRenderer.

Parameters

ent	The Entity to which this EnemyRenderer is attached.

3.9.3 Member Function Documentation

3.9.3.1 Draw()

```
void EnemyRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

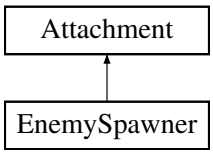
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/Enemy↩ Renderer.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/Enemy↩ Renderer.cpp

3.10 EnemySpawner Class Reference

```
#include <EnemySpawner.h>
```

Inheritance diagram for EnemySpawner:



Public Member Functions

- EnemySpawner (Entity *ent)
- void SpawnEnemy ()
- void SpawnWall ()
- virtual void Init () override
- virtual void Update (float deltaTime) override

Public Attributes

- float enemySpawnInterval
- float wallSpawnInterval

3.10.1 Detailed Description

Attachment that is used to spawn enemies and walls. Requires the Tunnel attachment.

3.10.2 Constructor & Destructor Documentation

3.10.2.1 EnemySpawner()

```
EnemySpawner::EnemySpawner (
            Entity * ent )  [inline]
```

Constructor for EnemySpawner. Constructs an new EnemySpawner.

Parameters

ent	The Entity to which this EnemySpawner is attached.

3.10.3 Member Function Documentation

3.10.3.1 Init()

```
void EnemySpawner::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Attachment.

3.10.3.2 SpawnEnemy()

```
void EnemySpawner::SpawnEnemy ( )
```

Spawn an enemy in a random track on the tunnel.

3.10.3.3 SpawnWall()

```
void EnemySpawner::SpawnWall ( )
```

Spawn a randomized array of walls.

3.10.3.4 Update()

```
void EnemySpawner::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from Attachment.

3.10.4 Member Data Documentation

3.10.4.1 enemySpawnInterval

```
float EnemySpawner::enemySpawnInterval
```

The time between two enemy spawns in miliseconds.

3.10.4.2 wallSpawnInterval

```
float EnemySpawner::wallSpawnInterval
```

The time between two wall spawns in miliseconds.

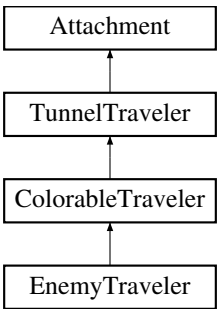
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/EnemySpawner.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/EnemySpawner.cpp

3.11 EnemyTraveler Class Reference

```
#include <EnemyTraveler.h>
```

Inheritance diagram for EnemyTraveler:



Public Member Functions

- EnemyTraveler (Entity *ent)
- void ShootBullet ()
- virtual void Update (float deltaTime) override
- virtual void OnCollision (Collider *other) override

Additional Inherited Members

3.11.1 Detailed Description

TunnelTraveler attachment that describes enemy ship behaviours.

3.11.2 Constructor & Destructor Documentation

3.11.2.1 EnemyTraveler()

```
EnemyTraveler::EnemyTraveler (
            Entity * ent )  [inline]
```

Constructor for EnemyTraveler. Constructs an new EnemyTraveler.

Parameters

ent	The Entity to which this EnemyTraveler is attached.

3.11.3 Member Function Documentation

3.11.3.1 OnCollision()

```
void EnemyTraveler::OnCollision (
            Collider * other )  [override], [virtual]
```

Method to call when the Entity is colliding with another Entity that has a Collider.

Parameters

other	The other Collider with which the Entity collided.

Reimplemented from Attachment.

3.11.3.2 ShootBullet()

```
void EnemyTraveler::ShootBullet ( )
```

Spawn a new random bullet.

3.11.3.3 Update()

```
void EnemyTraveler::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from ColorableTraveler.

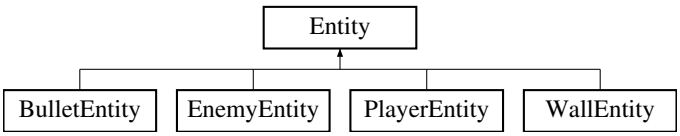
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/Enemy↩ Traveler.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/Enemy↩ Traveler.cpp

3.12 Entity Class Reference

```
#include <Entity.h>
```

Inheritance diagram for Entity:



Public Member Functions

- Entity ()
- ~Entity ()
- bool IsActive ()
- void SetActive (bool val)
- template<class T >
  T * AddAttachment ()
- template<class T >
  T * GetAttachment ()
- template<class T >
  LinkedList< T > * GetAttachments ()
- void Init ()
- void Update (float deltaTime)
- void Render ()
- void OnCollision (Collider *other)

3.12.1 Detailed Description

Baseclass for anything that can be placed in a Level. Every Entity is a collection of Attachments, all of which define its position, its behaviour and how its going to be rendered.

3.12.2 Constructor & Destructor Documentation

3.12.2.1 Entity()

```
Entity::Entity ( )
```

Constructor for Entity. Constructs an new Entity.

3.12.2.2 ~Entity()

```
Entity::~Entity ( )
```

Destructor for Entity. Destructs the Entity by deleting its Attachments.

3.12.3 Member Function Documentation

3.12.3.1 AddAttachment()

```
template<class T >
T * Entity::AddAttachment  [inline]
```

Create and add new Attachment of type T to the Entity.

Returns

The newly created Attachment.

3.12.3.2 GetAttachment()

```
template<class T >
T * Entity::GetAttachment  [inline]
```

Method to access Attachments on this Entity.

Returns

The first Attachment of type T that is found in the attachment list of the Entity.

3.12.3.3 GetAttachments()

```
template<class T >
LinkedList< T > * Entity::GetAttachments  [inline]
```

Method to retreive a list of Attachments on this Entity.

Returns

A LinkedList of the attachments found.

3.12.3.4 Init()

```
void Entity::Init ( )
```

Initialize this Entity. Initializes this Entity and the Attachments attached to it.

3.12.3.5 IsActive()

```
bool Entity::IsActive ( )
```

Check if Entity is active.

Returns

true of Entity is active, false otherwise

3.12.3.6 OnCollision()

```
void Entity::OnCollision (
            Collider * other )
```

Trigger Collision calls on all Attachments.

Parameters

other	The Collider that the Entity collided with.

3.12.3.7 Render()

```
void Entity::Render ( )
```

Render this Entity. Render this Entity according to its Renderer attachment, if it has one.

3.12.3.8 SetActive()

```
void Entity::SetActive (
            bool val )
```

Activate or deactivate Entity.

Parameters

val	Activate if true, deactivate otherwise.

3.12.3.9 Update()

```
void Entity::Update (
            float deltaTime )
```

Update this Entity. Updates every Attachment on this Entity.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

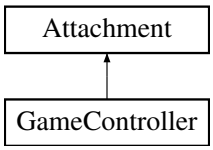
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/Entity.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/Entity.cpp

3.13 GameController Class Reference

```
#include <GameController.h>
```

Inheritance diagram for GameController:



Public Member Functions

- GameController (Entity *ent)
- void StartGame ()
- void EndGame ()
- void SpawnPlayer ()
- virtual void Init () override
- virtual void Update (float deltaTime) override

Static Public Member Functions

- static void IncrementScore ()
- static void EndCurrentGame ()

Static Public Attributes

- static GameController * currentController = NULL

3.13.1 Detailed Description

Main Attachment that controls the game. Starts and ends the game, keeps track of score, controls the Tunnel and Spawner.

3.13.2 Constructor & Destructor Documentation

3.13.2.1 GameController()

```
GameController::GameController (
            Entity * ent )  [inline]
```

Constructor for GameController. Constructs an new GameController.

Parameters

ent	The Entity to which this GameController is attached.

3.13.3 Member Function Documentation

3.13.3.1 EndCurrentGame()

```
void GameController::EndCurrentGame ( )  [static]
```

End the game being controlled by the currentController.

3.13.3.2 EndGame()

```
void GameController::EndGame ( )
```

End the game. Ending the game displays a gameover text and prompts the user to replay by pressing SPACE.

3.13.3.3 IncrementScore()

```
void GameController::IncrementScore ( )  [static]
```

Increment the score.

3.13.3.4 Init()

```
void GameController::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Attachment.

3.13.3.5 SpawnPlayer()

```
void GameController::SpawnPlayer ( )
```

Spawns a new player controlled Entity.

3.13.3.6 StartGame()

```
void GameController::StartGame ( )
```

Start the game. Starting the game resets the score and activates Attachments that drive the game.

3.13.3.7 Update()

```
void GameController::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from Attachment.

3.13.4 Member Data Documentation

3.13.4.1 currentController

```
GameController * GameController::currentController = NULL  [static]
```

The current GameController that is controlling the game. This is the last GameController object to be instantiated.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/GameController.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/GameController.cpp

3.14 Level Class Reference

```
#include <Level.h>
```

Public Member Functions

- Level ()
- ~Level ()
- void Init ()
- void Update (float deltaTime)
- void Render ()
- void AddEntity (Entity *ent)
- void RemoveEntity (Entity *ent)
- void MarkForDestruction (Entity *ent)

Static Public Member Functions

- static void SpawnEntity (Entity *ent)
- static void DestroyEntity (Entity *ent)
- template<class T >
  static T * FindAttachment ()
- template<class T >
  static T * **FindAttachment** ()

Static Public Attributes

- static Level * currentLevel = NULL

3.14.1 Detailed Description

Class representing a level in the game. Each Level is simply a list of Entities that are present in that level.

3.14.2 Constructor & Destructor Documentation

3.14.2.1 Level()

```
Level::Level ( )
```

Constructor for Level. Constructs an new Level.

3.14.2.2 ~Level()

```
Level::~Level ( )
```

Destructor for Level. Destructs the Level by deleting the Entities in it.

3.14.3 Member Function Documentation

3.14.3.1 AddEntity()

```
void Level::AddEntity (
            Entity * ent )
```

Add an Entity to this Level.

Parameters

ent	Pointer to an Entity object.

3.14.3.2 DestroyEntity()

```
void Level::DestroyEntity (
            Entity * ent )  [static]
```

Destroy an Entity on the current Level.

Parameters

ent	The Entity to destroy.

3.14.3.3 FindAttachment()

```
template<class T >
static T* Level::FindAttachment ( )  [static]
```

Look for an Attachment of type T in the current level.

3.14.3.4 Init()

```
void Level::Init ( )
```

Initialize all Entities in this Level.

3.14.3.5 MarkForDestruction()

```
void Level::MarkForDestruction (
            Entity * ent )
```

Mark an Entity for destruction. Marked Entities will be destroyed after the next Update call.

3.14.3.6 RemoveEntity()

```
void Level::RemoveEntity (
            Entity * ent )
```

Remove an Entity from this Level.

Parameters

ent	Pointer to an Entity object.

3.14.3.7 Render()

```
void Level::Render ( )
```

Render all Entities in this Level.

3.14.3.8 SpawnEntity()

```
void Level::SpawnEntity (
            Entity * ent )  [static]
```

Spawn an Entity on the current Level.

Parameters

ent	The Entity to spawn.

3.14.3.9 Update()

```
void Level::Update (
            float deltaTime )
```

Update all Entities in this Level.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

3.14.4 Member Data Documentation

3.14.4.1 currentLevel

```
Level * Level::currentLevel = NULL  [static]
```

The current Level that is being run. This is the last Level object to be instantiated.

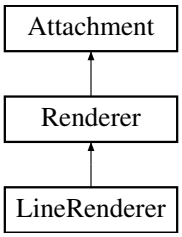
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/Level.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/Level.cpp

3.15 LineRenderer Class Reference

```
#include <LineRenderer.h>
```

Inheritance diagram for LineRenderer:



Public Member Functions

- LineRenderer (Entity *ent)
- void Draw () override

Additional Inherited Members

3.15.1 Detailed Description

Renderer that draws a line. The LineRenderer draws a line according to its list of points.

3.15.2 Constructor & Destructor Documentation

3.15.2.1 LineRenderer()

```
LineRenderer::LineRenderer (
            Entity * ent )  [inline]
```

Constructor for LineRenderer. Constructs an new LineRenderer.

Parameters

ent	The Entity to which this LineRenderer is attached.

3.15.3 Member Function Documentation

3.15.3.1 Draw()

```
void LineRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/LineRenderer.↩h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/LineRenderer.↩cpp

3.16 LinkedList< T > Class Template Reference

```
#include <LinkedList.h>
```

Public Member Functions

- LinkedList ()
- ~LinkedList ()
- bool IsEmpty ()
- T * Get (int index)
- void Add (T *element)
- void Remove (T *element)
- int Size ()
- bool Contains (T *element)
- LinkedListIterator< T > * Iterator ()

3.16.1 Detailed Description

template<class T>
class LinkedList< T >

Generic LinkedList class. A generic LinkedList class that can store elements of type T.

3.16.2 Constructor & Destructor Documentation

3.16.2.1 LinkedList()

```
template<class T >
LinkedList< T >::LinkedList
```

Constructor for LinkedList. Constructs an empty LinkedList.

3.16.2.2 ~LinkedList()

```
template<class T >
LinkedList< T >::~LinkedList
```

Destructor for LinkedList. Destructs the LinkedList by deallocating all nodes.

3.16.3 Member Function Documentation

3.16.3.1 Add()

```
template<class T >
void LinkedList< T >::Add (
            T * element )
```

Push method for LinkedList. Places a new element at the end of the LinkedList.

Parameters

element	The element to be added to the LinkedList.

3.16.3.2 Contains()

```
template<class T >
bool LinkedList< T >::Contains (
            T * element )
```

Check if LinkedList contains an element.

Parameters

element	The element to be checked for.

3.16.3.3 Get()

```
template<class T >
T * LinkedList< T >::Get (
            int index )
```

Indexing method for LinkedList. Index into LinkedList and return the indexed element.

Parameters

index	The index of the element to return.

Returns

Pointer to the element at given index.

3.16.3.4 IsEmpty()

```
template<class T >
bool LinkedList< T >::IsEmpty
```

Check if LinkedList is empty.

Returns

true if LinkedList is empty, false otherwise.

3.16.3.5 Iterator()

```
template<class T >
LinkedListIterator< T > * LinkedList< T >::Iterator
```

Iterator for the LinkedList

Returns

A new iterator that points to the head of the LinkedList.

3.16.3.6 Remove()

```
template<class T >
void LinkedList< T >::Remove (
            T * element )
```

Remove method for LinkedList. Removes the given element from the LinkedList.

Parameters

element	The element to be removed from the LinkedList.

3.16.3.7 Size()

```
template<class T >
int LinkedList< T >::Size
```

Size of LinkedList.

Returns

Current size of the LinkedList.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/LinkedList.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/LinkedList.cpp

3.17 LinkedListIterator< T > Class Template Reference

```
#include <LinkedList.h>
```

Public Member Functions

- LinkedListIterator (node< T > *start)
- void Reset ()
- T * Next ()
- bool HasNext ()

3.17.1 Detailed Description

template<class T>
class LinkedListIterator< T >

Generic class used to iterate over LinkedLists. A generic class that can be used to iterate over a LinkedList specialized to type T.

3.17.2 Constructor & Destructor Documentation

3.17.2.1 LinkedListIterator()

```
template<class T >
LinkedListIterator< T >::LinkedListIterator (
            node< T > * start )
```

Constructor for LinkedListIterator. Constructs a LinkedListIterator starting at the node <start>.

Parameters

start	The node at which iteration will start.

3.17.3 Member Function Documentation

3.17.3.1 HasNext()

```
template<class T >
bool LinkedListIterator< T >::HasNext
```

Check for next element. Check and return whether or not there is any more elements left to iterate over.

Returns

true if there is an element left to iterate over, false otherwise.

3.17.3.2 Next()

```
template<class T >
T * LinkedListIterator< T >::Next
```

Get next item the the iterator. Return the next element that the iterator is pointing to.

Returns

A pointer to next element if there is a next element to be found, NULL otherwise.

3.17.3.3 Reset()

```
template<class T >
void LinkedListIterator< T >::Reset
```

Reset the iterator to the start. Reset the iterator to point to the starting node.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/LinkedList.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/LinkedList.cpp

3.18 node< T > Struct Template Reference

Public Member Functions

- **node** (T *dat)

Public Attributes

- T * **data**
- node< T > * **next**

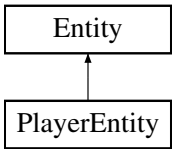
The documentation for this struct was generated from the following file:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/LinkedList.h

3.19 PlayerEntity Class Reference

```
#include <PlayerEntity.h>
```

Inheritance diagram for PlayerEntity:



Public Member Functions

- PlayerEntity ()

3.19.1 Detailed Description

Preset player entity.

3.19.2 Constructor & Destructor Documentation

3.19.2.1 PlayerEntity()

```
PlayerEntity::PlayerEntity ( )
```

Constructor for PlayerEntity. Constructs an new PlayerEntity.

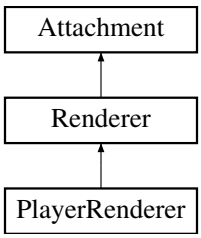
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/PlayerEntity.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/PlayerEntity.cpp

3.20 PlayerRenderer Class Reference

```
#include <PlayerRenderer.h>
```

Inheritance diagram for PlayerRenderer:



Public Member Functions

- PlayerRenderer (Entity *ent)
- virtual void Draw () override

Additional Inherited Members

3.20.1 Detailed Description

Renderer used to display the player.

3.20.2 Constructor & Destructor Documentation

3.20.2.1 PlayerRenderer()

```
PlayerRenderer::PlayerRenderer (
            Entity * ent )  [inline]
```

Constructor for PlayerRenderer. Constructs an new PlayerRenderer.

Parameters

ent	The Entity to which this PlayerRenderer is attached.

3.20.3 Member Function Documentation

3.20.3.1 Draw()

```
void PlayerRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

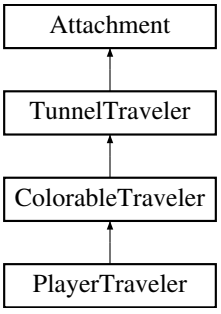
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/PlayerRenderer.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/PlayerRenderer.cpp

3.21 PlayerTraveler Class Reference

```
#include <PlayerTraveler.h>
```

Inheritance diagram for PlayerTraveler:



Public Member Functions

- PlayerTraveler (Entity *ent)
- bool IsMouthOpen ()
- void MakeImmune (float duration)
- void ShootBullet ()
- virtual void Update (float deltaTime) override
- virtual void TunnelUpdate (float deltaTime, Tunnel *refTunnel) override
- virtual void OnCollision (Collider *other) override

Additional Inherited Members

3.21.1 Detailed Description

Attachment for a TunnelTraveler that is player controlled using the mouse.

3.21.2 Constructor & Destructor Documentation

3.21.2.1 PlayerTraveler()

```
PlayerTraveler::PlayerTraveler (
            Entity * ent )  [inline]
```

Constructor for PlayerTraveler. Constructs an new PlayerTraveler.

Parameters

ent	The Entity to which this PlayerTraveler is attached.

3.21.3 Member Function Documentation

3.21.3.1 IsMouthOpen()

```
bool PlayerTraveler::IsMouthOpen ( )
```

Method to inquire upon the openness of the player's mouth.

3.21.3.2 MakeImmune()

```
void PlayerTraveler::MakeImmune (
            float duration )
```

Make the player immmune for a duration.

Parameters

duration	The duration for which the player remains immune in miliseconds.

3.21.3.3 OnCollision()

```
void PlayerTraveler::OnCollision (
            Collider * other )  [override], [virtual]
```

Method to call when the Entity is colliding with another Entity that has a Collider.

Parameters

other	The other Collider with which the Entity collided.

Reimplemented from Attachment.

3.21.3.4 ShootBullet()

```
void PlayerTraveler::ShootBullet ( )
```

Spawn a new bullet that has the same colorType as the player.

3.21.3.5 TunnelUpdate()

```
void PlayerTraveler::TunnelUpdate (
            float deltaTime,
            Tunnel * refTunnel )  [override], [virtual]
```

Update method to be called by a Tunnel class. Handles updates related to the Tunnel.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.
refTunnel	The Tunnel that this TunnelTraveler is attached to.

Reimplemented from TunnelTraveler.

3.21.3.6 Update()

```
void PlayerTraveler::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from ColorableTraveler.

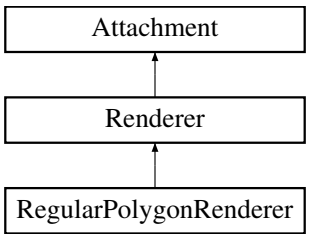
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/PlayerTraveler.↩ h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/PlayerTraveler.↩ cpp

3.22 RegularPolygonRenderer Class Reference

`#include <RegularPolygonRenderer.h>`

Inheritance diagram for RegularPolygonRenderer:



Public Member Functions

- RegularPolygonRenderer (Entity *ent)
- void Draw () override

Public Attributes

- float radius
- float vertexCount

3.22.1 Detailed Description

Class used to render regular polygons.

3.22.2 Constructor & Destructor Documentation

3.22.2.1 RegularPolygonRenderer()

```
RegularPolygonRenderer::RegularPolygonRenderer (
            Entity * ent )  [inline]
```

Constructor for RegularPolygonRenderer. Constructs an new RegularPolygonRenderer.

Parameters

ent	The Entity to which this RegularPolygonRenderer is attached.

3.22.3 Member Function Documentation

3.22.3.1 Draw()

```
void RegularPolygonRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

3.22.4 Member Data Documentation

3.22.4.1 radius

```
float RegularPolygonRenderer::radius
```

The polygon's radius

3.22.4.2 vertexCount

```
float RegularPolygonRenderer::vertexCount
```

The polygon's number of vertices

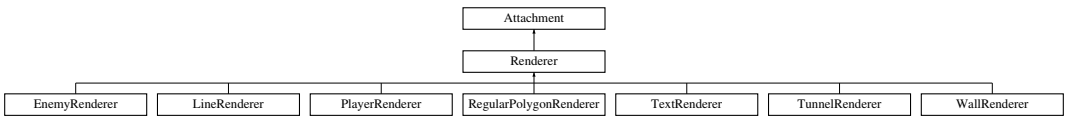
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/Regular↩ PolygonRenderer.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/Regular↩ PolygonRenderer.cpp

3.23 Renderer Class Reference

```
#include <Renderer.h>
```

Inheritance diagram for Renderer:



Public Member Functions

- Renderer (Entity *ent)
- virtual void Draw ()=0
- virtual void Init () override
- virtual void Update (float deltaTime) override

Public Attributes

- float r
- float **g**
- float **b**

3.23.1 Detailed Description

Abstract Renderer class. Declares the signature of the Draw method which will draw onto the screen according to subclass implementations.

3.23.2 Constructor & Destructor Documentation

3.23.2.1 Renderer()

```
Renderer::Renderer (
            Entity * ent )  [inline]
```

Constructor for Renderer. Constructs an new Renderer.

Parameters

ent	The Entity to which this Renderer is attached.

3.23.3 Member Function Documentation

3.23.3.1 Draw()

```
virtual void Renderer::Draw ( )  [pure virtual]
```

Draw to screen according to the specifications of this Renderer.

Implemented in TextRenderer, RegularPolygonRenderer, LineRenderer, TunnelRenderer, EnemyRenderer, PlayerRenderer, and WallRenderer.

3.23.3.2 Init()

```
void Renderer::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Attachment.

Reimplemented in TunnelRenderer.

3.23.3.3 Update()

```
void Renderer::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from Attachment.

3.23.4 Member Data Documentation

3.23.4.1 r

```
float Renderer::r
```

Floats defining the color of the drawing

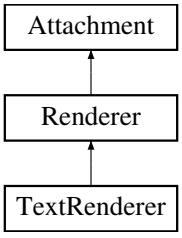
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/Renderer.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/Renderer.cpp

3.24 TextRenderer Class Reference

```
#include <TextRenderer.h>
```

Inheritance diagram for TextRenderer:



Public Member Functions

- TextRenderer (Entity *ent)
- void Draw () override

Public Attributes

- char text [64]
- Vector2 position

3.24.1 Detailed Description

Renderer used to display text on to the screen.

3.24.2 Constructor & Destructor Documentation

3.24.2.1 TextRenderer()

```
TextRenderer::TextRenderer (
            Entity * ent )  [inline]
```

Constructor for TextRenderer. Constructs an new TextRenderer.

Parameters

ent	The Entity to which this TextRenderer is attached.

3.24.3 Member Function Documentation

3.24.3.1 Draw()

```
void TextRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

3.24.4 Member Data Documentation

3.24.4.1 position

```
Vector2 TextRenderer::position
```

The position where the text will be displayed.

3.24.4.2 text

```
char TextRenderer::text[64]
```

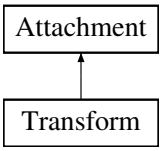
The text that will be displayed by the renderer. Max 63 characters.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/TextRenderer.↩ h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/TextRenderer.↩ cpp

3.25 Transform Class Reference

Inheritance diagram for Transform:



Public Member Functions

- Transform (Entity *ent)
- virtual void Init () override
- virtual void Update (float deltaTime) override

Public Attributes

- Vector2 position
- Vector2 scale
- float rotation

3.25.1 Constructor & Destructor Documentation

3.25.1.1 Transform()

```
Transform::Transform (
            Entity * ent )  [inline]
```

Constructor for Transform. Constructs an new Transform.

Parameters

ent	The Entity to which this Transform is attached.

3.25.2 Member Function Documentation

3.25.2.1 Init()

```
void Transform::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Attachment.

3.25.2.2 Update()

```
void Transform::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from Attachment.

3.25.3 Member Data Documentation

3.25.3.1 position

```
Vector2 Transform::position
```

Position of the Entity that the Transform is attached to.

3.25.3.2 rotation

```
float Transform::rotation
```

Rotation of the Entity that the Transform is attached to.

3.25.3.3 scale

```
Vector2 Transform::scale
```

Local scale factor of the Entity that the Transform is attached to.

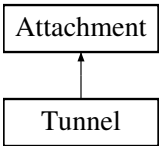
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/Transform.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/Transform.cpp

3.26 Tunnel Class Reference

```
#include <Tunnel.h>
```

Inheritance diagram for Tunnel:



Public Member Functions

- Tunnel (Entity *ent)
- ~Tunnel ()
- void AddTraveler (TunnelTraveler *traveler)
- void RemoveTraveler (TunnelTraveler *traveler)
- void ClearTunnel ()
- virtual void Init () override
- virtual void Update (float deltaTime) override

Public Attributes

- float innerRadius
- float outerRadius
- float spinSpeed
- int trackCount

3.26.1 Detailed Description

Attachment defining the properties and behaviour of a Tunnel. The tunnel keeps track of and updates the Tunnel↩ Travelers that have been attached to it.

3.26.2 Constructor & Destructor Documentation

3.26.2.1 Tunnel()

```
Tunnel::Tunnel (
            Entity * ent )
```

Constructor for Tunnel. Constructs an new Tunnel.

Parameters

ent	The Entity to which this Tunnel is attached.

3.26.2.2 ~Tunnel()

```
Tunnel::~Tunnel ( )
```

Destructor for Tunnel. Destructs the Tunnel by deleting the TunnelTravelers in it.

3.26.3 Member Function Documentation

3.26.3.1 AddTraveler()

```
void Tunnel::AddTraveler (
            TunnelTraveler * traveler )
```

Method to add a TunnelTraveler to the Tunnel.

3.26.3.2 ClearTunnel()

```
void Tunnel::ClearTunnel ( )
```

Destroys all TunnelTravelers on the Tunnel.

3.26.3.3 Init()

```
void Tunnel::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Attachment.

3.26.3.4 RemoveTraveler()

```
void Tunnel::RemoveTraveler (
            TunnelTraveler * traveler )
```

Method to remove a TunnelTraveler from the Tunnel.

3.26.3.5 Update()

```
void Tunnel::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from Attachment.

3.26.4 Member Data Documentation

3.26.4.1 innerRadius

```
float Tunnel::innerRadius
```

The tunnels's inner radius

3.26.4.2 outerRadius

```
float Tunnel::outerRadius
```

The polygon's outer radius

3.26.4.3 spinSpeed

```
float Tunnel::spinSpeed
```

The speed at which the tunnel spins in radians per second.

3.26.4.4 trackCount

```
int Tunnel::trackCount
```

The number of tracks on the tunnel

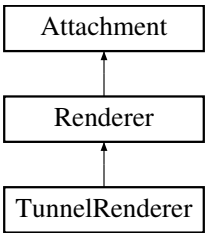
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/Tunnel.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/Tunnel.cpp

3.27 TunnelRenderer Class Reference

`#include <TunnelRenderer.h>`

Inheritance diagram for TunnelRenderer:



Public Member Functions

- TunnelRenderer (Entity *ent)
- void Draw () override
- virtual void Init () override

Additional Inherited Members

3.27.1 Detailed Description

Renderer used to render a "Tempest-like" tunnel

3.27.2 Constructor & Destructor Documentation

3.27.2.1 TunnelRenderer()

```
TunnelRenderer::TunnelRenderer (
            Entity * ent )  [inline]
```

Constructor for TunnelRenderer. Constructs an new TunnelRenderer.

Parameters

ent	The Entity to which this TunnelRenderer is attached.

3.27.3 Member Function Documentation

3.27.3.1 Draw()

```
void TunnelRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

3.27.3.2 Init()

```
void TunnelRenderer::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Renderer.

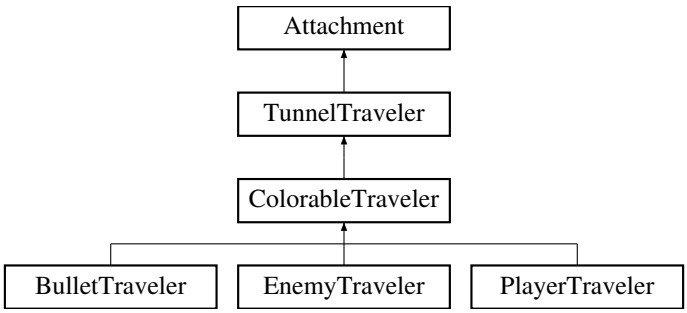
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/TunnelRenderer.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/TunnelRenderer.cpp

3.28 TunnelTraveler Class Reference

```
#include <TunnelTraveler.h>
```

Inheritance diagram for TunnelTraveler:



Public Member Functions

- TunnelTraveler (Entity *ent)
- virtual void TunnelUpdate (float deltaTime, Tunnel *refTunnel)
- virtual void Init () override
- virtual void Update (float deltaTime) override

Public Attributes

- float radius
- float velocity
- int track

3.28.1 Detailed Description

Attachment for Entities that can travel in a Tunnel.

3.28.2 Constructor & Destructor Documentation

3.28.2.1 TunnelTraveler()

```
TunnelTraveler::TunnelTraveler (
            Entity * ent )  [inline]
```

Constructor for TunnelTraveler. Constructs an new TunnelTraveler.

Parameters

ent	The Entity to which this TunnelTraveler is attached.

3.28.3 Member Function Documentation

3.28.3.1 Init()

```
void TunnelTraveler::Init ( )  [override], [virtual]
```

Initialize this Attachment.

Reimplemented from Attachment.

3.28.3.2 TunnelUpdate()

```
void TunnelTraveler::TunnelUpdate (
            float deltaTime,
            Tunnel * refTunnel )  [virtual]
```

Update method to be called by a Tunnel class. Handles updates related to the Tunnel.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.
refTunnel	The Tunnel that this TunnelTraveler is attached to.

Reimplemented in PlayerTraveler.

3.28.3.3 Update()

```
void TunnelTraveler::Update (
            float deltaTime )  [override], [virtual]
```

Update this Attachment.

Parameters

deltaTime	Time elapsed between the last frame and the current frame.

Reimplemented from Attachment.

Reimplemented in PlayerTraveler, ColorableTraveler, EnemyTraveler, and BulletTraveler.

3.28.4 Member Data Documentation

3.28.4.1 radius

```
float TunnelTraveler::radius
```

Float representing the radius that the traveler is placed in within the tunnel.

3.28.4.2 track

```
int TunnelTraveler::track
```

Integer representing the track number that the traveler is placed in within the tunnel.

3.28.4.3 velocity

```
float TunnelTraveler::velocity
```

Float representing the speed at which the radius is changing relative to seconds.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/Tunnel↩ Traveler.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/travelers/Tunnel↩ Traveler.cpp

3.29 Vector2 Class Reference

`#include <Vector2.h>`

Public Member Functions

- Vector2 ()
- Vector2 (float iX, float iY)

Static Public Member Functions

- static float Distance (Vector2 v1, Vector2 v2)

Public Attributes

- float x
- float **y**

Static Public Attributes

- static const Vector2 **zero**
- static const Vector2 **up**
- static const Vector2 **down**
- static const Vector2 **left**
- static const Vector2 **right**
- static const Vector2 **one**

3.29.1 Detailed Description

2 Dimensional vector class.

3.29.2 Constructor & Destructor Documentation

3.29.2.1 Vector2() [1/2]

```
Vector2::Vector2 ( )
```

Constructor for Vector2. Instantiates a Vector2 with values (0, 0).

3.29.2.2 Vector2() [2/2]

```
Vector2::Vector2 (
            float iX,
            float iY )
```

Constructor for Vector2.

Parameters

iX	The initial x value of the Vector.
iY	The initial y value of the Vector.

3.29.3 Member Function Documentation

3.29.3.1 Distance()

```
float Vector2::Distance (
            Vector2 v1,
            Vector2 v2 )  [static]
```

Calculate the distance between two vectors

Parameters

v1	First vector
v2	Second vector

3.29.4 Member Data Documentation

3.29.4.1 x

```
float Vector2::x
```

Values of the vector.

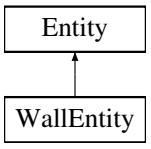
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/Vector2.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/Vector2.cpp

3.30 WallEntity Class Reference

```
#include <WallEntity.h>
```

Inheritance diagram for WallEntity:



Public Member Functions

- WallEntity ()

3.30.1 Detailed Description

Preset wall entity.

3.30.2 Constructor & Destructor Documentation

3.30.2.1 WallEntity()

```
WallEntity::WallEntity ( )
```

Constructor for WallEntity. Constructs an new WallEntity.

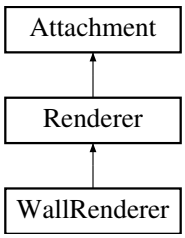
The documentation for this class was generated from the following files:

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/WallEntity.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/presets/WallEntity.cpp

3.31 WallRenderer Class Reference

```
#include <WallRenderer.h>
```

Inheritance diagram for WallRenderer:



Public Member Functions

- WallRenderer (Entity *ent)
- virtual void Draw () override

Additional Inherited Members

3.31.1 Detailed Description

Renderer for the wall obstacles.

3.31.2 Constructor & Destructor Documentation

3.31.2.1 WallRenderer()

```
WallRenderer::WallRenderer (
            Entity * ent )  [inline]
```

Constructor for WallRenderer. Constructs an new WallRenderer.

Parameters

ent	The Entity to which this WallRenderer is attached.

3.31.3 Member Function Documentation

3.31.3.1 Draw()

```
void WallRenderer::Draw ( )  [override], [virtual]
```

Draw to screen according to the specifications of this Renderer.

Implements Renderer.

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

- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/WallRenderer.h
- F:/ext_workspace/UbisoftNext/UbiNext2020_0/GameTest/glowEngine/attachments/renderers/WallRenderer.cpp

Index

~Entity
Entity, 22
~Level
Level, 28
~LinkedList
LinkedList< T >, 32
~Tunnel
Tunnel, 49

Add
LinkedList< T >, 32
AddAttachment
Entity, 22
AddEntity
Level, 28
AddTraveler
Tunnel, 49
Attachment, 5
Attachment, 6
GetEntity, 6
Init, 6
IsActive, 6
OnCollision, 7
SetActive, 7
Update, 7

BulletEntity, 8
BulletEntity, 8
BulletTraveler, 9
BulletTraveler, 9
Update, 9

CircleCollider, 10
CircleCollider, 11
CollidesWith, 11
radius, 11
ClearTunnel
Tunnel, 49
Collider, 12
Collider, 12
CollidesWith, 12
CollidesWith
CircleCollider, 11
Collider, 12
ColorableTraveler, 13
ColorableTraveler, 13
colorType, 14
Update, 14
ColorType, 14
colorType
ColorableTraveler, 14
Contains
LinkedList< T >, 33
currentController
GameController, 26
currentLevel
Level, 30

DestroyEntity
Level, 28
Distance
Vector2, 56
Draw
EnemyRenderer, 16
LineRenderer, 31
PlayerRenderer, 38
RegularPolygonRenderer, 42
Renderer, 43
TextRenderer, 45
TunnelRenderer, 51
WallRenderer, 58

EndCurrentGame
GameController, 25
EndGame
GameController, 25
EnemyEntity, 15
EnemyEntity, 15
EnemyRenderer, 16
Draw, 16
EnemyRenderer, 16
EnemySpawner, 17
EnemySpawner, 17
enemySpawnInterval, 19
Init, 18
SpawnEnemy, 18
SpawnWall, 18
Update, 18
wallSpawnInterval, 19
enemySpawnInterval
EnemySpawner, 19
EnemyTraveler, 19
EnemyTraveler, 20
OnCollision, 20
ShootBullet, 20
Update, 20
Entity, 21
~Entity, 22
AddAttachment, 22
Entity, 22

GetAttachment, 22
GetAttachments, 22
Init, 23
IsActive, 23
OnCollision, 23
Render, 23
SetActive, 23
Update, 24

FindAttachment
Level, 28

GameController, 24
currentController, 26
EndCurrentGame, 25
EndGame, 25
GameController, 25
IncrementScore, 25
Init, 26
SpawnPlayer, 26
StartGame, 26
Update, 26
Get
LinkedList< T >, 33
GetAttachment
Entity, 22
GetAttachments
Entity, 22
GetEntity
Attachment, 6

HasNext
LinkedListIterator< T >, 35

IncrementScore
GameController, 25
Init
Attachment, 6
EnemySpawner, 18
Entity, 23
GameController, 26
Level, 29
Renderer, 43
Transform, 47
Tunnel, 49
TunnelRenderer, 52
TunnelTraveler, 53
innerRadius
Tunnel, 50
IsActive
Attachment, 6
Entity, 23
IsEmpty
LinkedList< T >, 33
IsMouthOpen
PlayerTraveler, 39
Iterator
LinkedList< T >, 33

Level, 27
~Level, 28
AddEntity, 28
currentLevel, 30
DestroyEntity, 28
FindAttachment, 28
Init, 29
Level, 27
MarkForDestruction, 29
RemoveEntity, 29
Render, 29
SpawnEntity, 29
Update, 29
LineRenderer, 30
Draw, 31
LineRenderer, 31
LinkedList
LinkedList< T >, 32
LinkedList< T >, 31
~LinkedList, 32
Add, 32
Contains, 33
Get, 33
IsEmpty, 33
Iterator, 33
LinkedList, 32
Remove, 34
Size, 34
LinkedListIterator
LinkedListIterator< T >, 35
LinkedListIterator< T >, 34
HasNext, 35
LinkedListIterator, 35
Next, 35
Reset, 36

MakeImmune
PlayerTraveler, 39
MarkForDestruction
Level, 29

Next
LinkedListIterator< T >, 35
node< T >, 36

OnCollision
Attachment, 7
EnemyTraveler, 20
Entity, 23
PlayerTraveler, 40
outerRadius
Tunnel, 50

PlayerEntity, 36
PlayerEntity, 37
PlayerRenderer, 37
Draw, 38
PlayerRenderer, 38
PlayerTraveler, 38
IsMouthOpen, 39

MakeImmune, 39
OnCollision, 40
PlayerTraveler, 39
ShootBullet, 40
TunnelUpdate, 40
Update, 40
position
TextRenderer, 45
Transform, 47

r
Renderer, 44
radius
CircleCollider, 11
RegularPolygonRenderer, 42
TunnelTraveler, 54
RegularPolygonRenderer, 41
Draw, 42
radius, 42
RegularPolygonRenderer, 41
vertexCount, 42
Remove
LinkedList< T >, 34
RemoveEntity
Level, 29
RemoveTraveler
Tunnel, 49
Render
Entity, 23
Level, 29
Renderer, 42
Draw, 43
Init, 43
r, 44
Renderer, 43
Update, 44
Reset
LinkedListIterator< T >, 36
rotation
Transform, 47

scale
Transform, 47
SetActive
Attachment, 7
Entity, 23
ShootBullet
EnemyTraveler, 20
PlayerTraveler, 40
Size
LinkedList< T >, 34
SpawnEnemy
EnemySpawner, 18
SpawnEntity
Level, 29
SpawnPlayer
GameController, 26
SpawnWall
EnemySpawner, 18
spinSpeed
Tunnel, 50
StartGame
GameController, 26

text
TextRenderer, 46
TextRenderer, 44
Draw, 45
position, 45
text, 46
TextRenderer, 45
track
TunnelTraveler, 54
trackCount
Tunnel, 50
Transform, 46
Init, 47
position, 47
rotation, 47
scale, 47
Transform, 46
Update, 47
Tunnel, 48
~Tunnel, 49
AddTraveler, 49
ClearTunnel, 49
Init, 49
innerRadius, 50
outerRadius, 50
RemoveTraveler, 49
spinSpeed, 50
trackCount, 50
Tunnel, 48
Update, 49
TunnelRenderer, 51
Draw, 51
Init, 52
TunnelRenderer, 51
TunnelTraveler, 52
Init, 53
radius, 54
track, 54
TunnelTraveler, 53
TunnelUpdate, 53
Update, 54
velocity, 54
TunnelUpdate
PlayerTraveler, 40
TunnelTraveler, 53

Update
Attachment, 7
BulletTraveler, 9
ColorableTraveler, 14
EnemySpawner, 18
EnemyTraveler, 20
Entity, 24
GameController, 26

Level, 29
PlayerTraveler, 40
Renderer, 44
Transform, 47
Tunnel, 49
TunnelTraveler, 54

Vector2, 55
Distance, 56
Vector2, 55
x, 56
velocity
TunnelTraveler, 54
vertexCount
RegularPolygonRenderer, 42

WallEntity, 56
WallEntity, 57
WallRenderer, 57
Draw, 58
WallRenderer, 58
wallSpawnInterval
EnemySpawner, 19

x
Vector2, 56