

RType

Generated by Doxygen 1.9.1

1 Namespace Index	1
1.1 Namespace List	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	7
4.1 File List	7
5 Namespace Documentation	9
5.1 ecs Namespace Reference	9
5.1.1 Detailed Description	9
6 Class Documentation	11
6.1 ButtonComp Class Reference	11
6.1.1 Detailed Description	11
6.1.2 Constructor & Destructor Documentation	11
6.1.2.1 ButtonComp()	11
6.1.3 Member Data Documentation	12
6.1.3.1 sprite	12
6.1.3.2 texture	12
6.2 buttonSys Class Reference	13
6.3 ButtonSys Class Reference	13
6.3.1 Detailed Description	13
6.4 client Class Reference	13
6.5 ecs::ComponentArray< T > Class Template Reference	14
6.5.1 Detailed Description	14
6.5.2 Member Function Documentation	14
6.5.2.1 EntityDestroyed()	14
6.5.2.2 GetData()	15
6.5.2.3 InsertData()	15
6.5.2.4 RemoveData()	15
6.6 ecs::ComponentManager Class Reference	16
6.6.1 Detailed Description	16
6.6.2 Member Function Documentation	16
6.6.2.1 AddComponent()	16
6.6.2.2 EntityDestroyed()	17
6.6.2.3 GetComponent()	17
6.6.2.4 GetComponentType()	18
6.6.2.5 RegisterComponent()	18
6.6.2.6 RemoveComponent()	18

6.7 ecs::Coordinator Class Reference	19
6.7.1 Detailed Description	19
6.7.2 Member Function Documentation	19
6.7.2.1 AddComponent()	20
6.7.2.2 CreateEntity()	21
6.7.2.3 DestroyEntity()	21
6.7.2.4 GetComponent()	21
6.7.2.5 GetComponentType()	22
6.7.2.6 RegisterComponent()	22
6.7.2.7 RegisterSystem()	22
6.7.2.8 RemoveComponent()	22
6.7.2.9 setSystemSignature()	23
6.8 EnemyComp Class Reference	23
6.9 EnemySys Class Reference	23
6.9.1 Detailed Description	24
6.10 EnnemyComp Class Reference	24
6.10.1 Detailed Description	24
6.10.2 Constructor & Destructor Documentation	24
6.10.2.1 EnnemyComp()	24
6.11 EnnemySys Class Reference	25
6.11.1 Detailed Description	25
6.11.2 Member Function Documentation	25
6.11.2.1 drawEnemies()	25
6.11.2.2 getEnemyByld()	26
6.12 ecs::EntityManager Class Reference	26
6.12.1 Detailed Description	27
6.12.2 Member Function Documentation	27
6.12.2.1 CreateEntity()	27
6.12.2.2 DestroyEntity()	27
6.12.2.3 GetSignature()	27
6.12.2.4 SetSignature()	28
6.13 Game Class Reference	28
6.13.1 Detailed Description	29
6.13.2 Constructor & Destructor Documentation	29
6.13.2.1 Game()	29
6.13.3 Member Function Documentation	29
6.13.3.1 eventHandler()	29
6.13.3.2 gameLoop()	30
6.13.3.3 getLife()	30
6.13.3.4 Init()	30
6.13.3.5 isMenu()	30
6.13.3.6 Server()	31

6.13.3.7 setLife()	31
6.14 GameContainer Class Reference	31
6.14.1 Detailed Description	32
6.14.2 Constructor & Destructor Documentation	32
6.14.2.1 GameContainer()	32
6.14.3 Member Data Documentation	32
6.14.3.1 event	32
6.14.3.2 window	33
6.15 GameGraphic Class Reference	33
6.15.1 Detailed Description	34
6.15.2 Constructor & Destructor Documentation	34
6.15.2.1 GameGraphic()	34
6.15.3 Member Function Documentation	34
6.15.3.1 eventHandler()	34
6.15.4 Member Data Documentation	34
6.15.4.1 _container	35
6.16 ecs::IComponentArray Class Reference	35
6.16.1 Detailed Description	35
6.16.2 Constructor & Destructor Documentation	36
6.16.2.1 IComponentArray()	36
6.16.3 Member Function Documentation	36
6.16.3.1 EntityDestroyed()	36
6.16.3.2 operator=() [1/2]	36
6.16.3.3 operator=() [2/2]	37
6.17 IGameContext Class Reference	37
6.17.1 Detailed Description	38
6.17.2 Member Function Documentation	38
6.17.2.1 eventHandler()	38
6.17.2.2 gameLoop()	38
6.17.2.3 Init()	39
6.17.2.4 isMenu()	39
6.17.2.5 Server()	39
6.18 InteractServer Class Reference	40
6.18.1 Detailed Description	40
6.18.2 Constructor & Destructor Documentation	41
6.18.2.1 InteractServer()	41
6.18.2.2 ~InteractServer()	41
6.18.3 Member Function Documentation	41
6.18.3.1 createServer()	41
6.18.3.2 getServerInfos()	42
6.18.3.3 test()	42
6.19 Menu Class Reference	42

6.19.1 Detailed Description	43
6.19.2 Constructor & Destructor Documentation	43
6.19.2.1 Menu()	43
6.19.3 Member Function Documentation	43
6.19.3.1 eventHandler()	43
6.19.3.2 gameLoop()	44
6.19.3.3 Init()	44
6.19.3.4 isMenu()	44
6.19.3.5 Server()	45
6.20 Network Class Reference	45
6.20.1 Detailed Description	45
6.21 ParallaxComp Class Reference	46
6.21.1 Detailed Description	46
6.22 ParallaxSys Class Reference	46
6.22.1 Detailed Description	47
6.22.2 Member Function Documentation	47
6.22.2.1 MoveAndDraw()	47
6.23 Server Class Reference	47
6.24 SpaceShipComp Class Reference	47
6.24.1 Detailed Description	48
6.24.2 Constructor & Destructor Documentation	48
6.24.2.1 SpaceShipComp()	48
6.25 SpaceShipSys Class Reference	48
6.25.1 Detailed Description	49
6.25.2 Member Function Documentation	49
6.25.2.1 animAndDraw()	49
6.25.2.2 getPlayer()	50
6.25.2.3 getPlayerById()	50
6.25.2.4 move()	50
6.25.2.5 removePlayer()	50
6.25.2.6 shoot()	51
6.26 ecs::System Class Reference	51
6.26.1 Detailed Description	51
6.26.2 Member Data Documentation	52
6.26.2.1 mapEntities	52
6.27 ecs::SystemManager Class Reference	52
6.27.1 Detailed Description	52
6.27.2 Member Function Documentation	52
6.27.2.1 ChangedEntitySignature()	52
6.27.2.2 CreateSystem()	53
6.27.2.3 RemoveOldEntity()	53
6.27.2.4 SetSystemSignature()	53

6.28 TCP_Connection Class Reference	54
6.28.1 Detailed Description	54
6.28.2 Constructor & Destructor Documentation	54
6.28.2.1 TCP_Connection()	54
6.28.3 Member Function Documentation	55
6.28.3.1 handleWrite()	55
6.28.3.2 socket()	55
6.29 TCP_Server Class Reference	55
6.29.1 Detailed Description	56
6.29.2 Constructor & Destructor Documentation	56
6.29.2.1 TCP_Server()	56
6.29.3 Member Function Documentation	56
6.29.3.1 getConnexions()	56
6.30 TextureComp Class Reference	57
6.30.1 Detailed Description	57
6.30.2 Constructor & Destructor Documentation	57
6.30.2.1 TextureComp()	57
6.30.3 Member Data Documentation	57
6.30.3.1 texture	58
6.30.3.2 typeT	58
6.31 UDP_Server Class Reference	58
6.31.1 Detailed Description	59
6.31.2 Constructor & Destructor Documentation	59
6.31.2.1 UDP_Server()	59
7 File Documentation	61
7.1 inc/client/Bullets.hpp File Reference	61
7.1.1 Detailed Description	61
7.2 inc/client/Button.hpp File Reference	61
7.2.1 Detailed Description	62
7.3 inc/client/Ennemy.hpp File Reference	62
7.3.1 Detailed Description	62
7.4 inc/client/GameContainer.hpp File Reference	62
7.4.1 Detailed Description	63
7.5 inc/client/GameGraphic.hpp File Reference	63
7.5.1 Detailed Description	63
7.6 inc/client/interactServer.hpp File Reference	63
7.6.1 Detailed Description	64
7.7 inc/client/Parallax.hpp File Reference	64
7.7.1 Detailed Description	64
7.8 inc/client/Screens.hpp File Reference	64
7.8.1 Detailed Description	64

7.9 inc/client/SpaceShip.hpp File Reference	65
7.9.1 Detailed Description	65
7.10 inc/client/Texture.hpp File Reference	65
7.10.1 Detailed Description	66
7.11 inc/Component.hpp File Reference	66
7.11.1 Detailed Description	66
7.12 inc/ComponentArray.hpp File Reference	66
7.12.1 Detailed Description	67
7.13 inc/ComponentManager.hpp File Reference	67
7.13.1 Detailed Description	67
7.14 inc/Coordinator.hpp File Reference	67
7.14.1 Detailed Description	68
7.15 inc/Entity.hpp File Reference	68
7.15.1 Detailed Description	68
7.16 inc/EntityManager.hpp File Reference	68
7.16.1 Detailed Description	69
7.17 inc/Network.hpp File Reference	69
7.17.1 Detailed Description	69
7.18 inc/Signature.hpp File Reference	69
7.18.1 Detailed Description	70
7.19 inc/System.hpp File Reference	70
7.19.1 Detailed Description	70
7.20 inc/SystemManager.hpp File Reference	70
7.20.1 Detailed Description	71
Index	73

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

ecs	Entity Component System	9
---------------------	---	-------------------

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

ButtonComp	11
ButtonSys	13
client	13
ecs::ComponentManager	16
ecs::Coordinator	19
boost::enable_shared_from_this	
TCP_Connection	54
UDP_Server	58
EnemyComp	23
EnnemyComp	24
ecs::EntityManager	26
GameContainer	31
GameGraphic	33
ecs::IComponentArray	35
ecs::ComponentArray< T >	14
IGameContext	37
Game	28
Menu	42
InteractServer	40
Network	45
ParallaxComp	46
Server	47
SpaceShipComp	47
ecs::System	51
EnemySys	23
EnnemySys	25
ParallaxSys	46
SpaceShipSys	48
buttonSys	13
ecs::SystemManager	52
TCP_Server	55
TextureComp	57

Chapter 3

Class Index

3.1 Class List

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

ButtonComp	
Button	11
buttonSys	13
ButtonSys	
Button system	13
client	13
ecs::ComponentArray< T >	
Array holding components Inherits from IComponentArray Class	14
ecs::ComponentManager	
Manager handling components	16
ecs::Coordinator	
Combine and handling each manager	19
EnemyComp	23
EnemySys	
Initialise Ennemies mmoves	23
EnnemyComp	
Ennemy Component	24
EnnemySys	
Ennemy System	25
ecs::EntityManager	
Manager handling entities	26
Game	
Game	28
GameContainer	
Game container	31
GameGraphic	
GameGraphic	33
ecs::IComponentArray	
Setup core of ArrayComponent throw Interface	35
IGameContext	
GameGraphic	37
InteractServer	
Class representing the InteractServer	40
Menu	
Menu	42

Network	
Class representing UDP & TCP Protocol Network	45
ParallaxComp	
Parallax Component	46
ParallaxSys	
Parallax System	46
Server	47
SpaceShipComp	
SpaceShip Component	47
SpaceShipSys	
SpaceShip System	48
ecs::System	
System classs	51
ecs::SystemManager	
Class representing the system manager	52
TCP_Connection	
Class representing the TCP Connection	54
TCP_Server	
Class representing the TCP Server	55
TextureComp	
Texture Component	57
UDP_Server	
Class representing the UDP_Server	58

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

inc/ Component.hpp	
Component Type	66
inc/ ComponentArray.hpp	
Array holding components	66
inc/ ComponentManager.hpp	
Manager handling components	67
inc/ Coordinator.hpp	
Combine and handling each manager	67
inc/ Entity.hpp	
Entity Type	68
inc/ EntityManager.hpp	
Manager handling entities	68
inc/ Network.hpp	
UDP & TCP Protocol with Boost::Asio	69
inc/ Server.hpp	??
inc/ Signature.hpp	
Binary signature	69
inc/ System.hpp	
System	70
inc/ SystemManager.hpp	
Manager handling systems	70
inc/ test.hpp	??
inc/client/ Bullets.hpp	
Bullets	61
inc/client/ Button.hpp	
Button	61
inc/client/ Enemy.hpp	
Enemy	62
inc/client/ GameContainer.hpp	
GameContainer	62
inc/client/ GameGraphic.hpp	
GameGraphic	63
inc/client/ InteractServer.hpp	
InteractServer	63
inc/client/ Parallax.hpp	
Parallax	64

inc/client/ Screens.hpp	
Screens	64
inc/client/ SpaceShip.hpp	
SpaceShip	65
inc/client/ Texture.hpp	
Texture	65
inc/ecs/ Enemy.hpp	??

Chapter 5

Namespace Documentation

5.1 ecs Namespace Reference

Entity Component [System](#).

Classes

- class [IComponentArray](#)
Setup core of ArrayComponent throw Interface.
- class [ComponentArray](#)
Array holding components Inherits from [IComponentArray](#) Class.
- class [ComponentManager](#)
Manager handling components.
- class [Coordinator](#)
Combine and handling each manager.
- class [EntityManager](#)
Manager handling entities.
- class [System](#)
[System](#) classs.
- class [SystemManager](#)
class representing the system manager

5.1.1 Detailed Description

Entity Component [System](#).

Chapter 6

Class Documentation

6.1 ButtonComp Class Reference

Button.

```
#include <Button.hpp>
```

Public Member Functions

- [ButtonComp](#) ()
Button delete.
- [ButtonComp](#) (std::string filepath)
Button.

Public Attributes

- sf::Texture [texture](#)
Button texture.
- sf::Sprite [sprite](#)
Button sprite.

6.1.1 Detailed Description

Button.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 ButtonComp()

```
ButtonComp::ButtonComp (  
    std::string filepath ) [inline]
```

Button.

Parameters

<i>texture</i>	
<i>pos</i>	
<i>size</i>	
<i>text</i>	
<i>font</i>	
<i>color</i>	

Returns[ButtonComp](#)

6.1.3 Member Data Documentation

6.1.3.1 sprite

```
sf::Sprite ButtonComp::sprite
```

Button sprite.

Parameters

<i>sprite</i>	
---------------	--

Returns

sf::Sprite

6.1.3.2 texture

```
sf::Texture ButtonComp::texture
```

Button texture.

Parameters

<i>texture</i>	
----------------	--

Returns

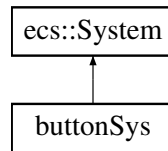
sf::Texture

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

- [inc/client/Button.hpp](#)

6.2 buttonSys Class Reference

Inheritance diagram for buttonSys:



Public Member Functions

- void [drawButtons](#) (std::shared_ptr< sf::RenderWindow > window)
Button drawer.

Additional Inherited Members

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

- [inc/client/Button.hpp](#)

6.3 ButtonSys Class Reference

Button system.

```
#include <Button.hpp>
```

6.3.1 Detailed Description

Button system.

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

- [inc/client/Button.hpp](#)

6.4 client Class Reference

Public Member Functions

- **client** (boost::asio::io_service &io_service, const std::string &server, const std::string &path)

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

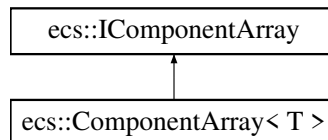
- [inc/test.hpp](#)

6.5 ecs::ComponentArray< T > Class Template Reference

Array holding components Inherits from [IComponentArray](#) Class.

```
#include <ComponentArray.hpp>
```

Inheritance diagram for ecs::ComponentArray< T >:



Public Member Functions

- [ComponentArray](#) ()
ComponentArray Constructor.
- [~ComponentArray](#) ()=default
ComponentArray Destructor.
- void [InsertData](#) (Entity entity, T component)
Insert Data Set Component inside ComponentArray.
- void [RemoveData](#) (Entity entity)
Remove Data Remove a specific Component of the ComponentArray.
- T & [GetData](#) (Entity entity)
Get Data Get data on a specific entity inside ComponentArray.
- void [EntityDestroyed](#) (Entity entity) override
Delete Entity Destroy a specific entity in ComponentArray.

6.5.1 Detailed Description

```
template<typename T>
class ecs::ComponentArray< T >
```

Array holding components Inherits from [IComponentArray](#) Class.

6.5.2 Member Function Documentation

6.5.2.1 EntityDestroyed()

```
template<typename T >
void ecs::ComponentArray< T >::EntityDestroyed (
    Entity entity ) [inline], [override], [virtual]
```

Delete Entity Destroy a specific entity in [ComponentArray](#).

Parameters

<i>entity</i>	Entity
---------------	--------

Implements [ecs::IComponentArray](#).

6.5.2.2 GetData()

```
template<typename T >
T& ecs::ComponentArray< T >::GetData (
    Entity entity ) [inline]
```

Get Data Get data on a specific entity inside [ComponentArray](#).

Parameters

<i>entity</i>	Entity
---------------	--------

Returns

T&

6.5.2.3 InsertData()

```
template<typename T >
void ecs::ComponentArray< T >::InsertData (
    Entity entity,
    T component ) [inline]
```

Insert Data Set Component inside [ComponentArray](#).

Parameters

<i>entity</i>	Entity
<i>component</i>	Component

6.5.2.4 RemoveData()

```
template<typename T >
void ecs::ComponentArray< T >::RemoveData (
    Entity entity ) [inline]
```

Remove Data Remove a specific Component of the [ComponentArray](#).

Parameters

<i>entity</i>	Entity
---------------	--------

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

- inc/[ComponentArray.hpp](#)

6.6 ecs::ComponentManager Class Reference

Manager handling components.

```
#include <ComponentManager.hpp>
```

Public Member Functions

- `template<typename T >`
`void RegisterComponent ()`
Initialize a new component in ecs.
- `template<typename T >`
`ComponentType GetComponentType ()`
Get a Component.
- `template<typename T >`
`void AddComponent (Entity entity, T component)`
Add a component.
- `template<typename T >`
`void RemoveComponent (Entity entity)`
Remove a component.
- `template<typename T >`
`T & GetComponent (Entity entity)`
Get a component.
- `template<typename T >`
`void EntityDestroyed (Entity entity)`
Destroy an entity.

6.6.1 Detailed Description

Manager handling components.

6.6.2 Member Function Documentation

6.6.2.1 AddComponent()

```
template<typename T >
void ecs::ComponentManager::AddComponent (
    Entity entity,
    T component ) [inline]
```

Add a component.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>entity, T</i>	component
<i>component</i>	

6.6.2.2 EntityDestroyed()

```
template<typename T >
void ComponentManager::EntityDestroyed (
    Entity entity )
```

Destroy an entity.

Template Parameters

<i>T</i>	
----------	--

6.6.2.3 GetComponent()

```
template<typename T >
T& ecs::ComponentManager::GetComponent (
    Entity entity ) [inline]
```

Get a component.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>entity</i>	
---------------	--

Returns

T

6.6.2.4 GetComponentType()

```
template<typename T >
ComponentType ecs::ComponentManager::GetComponentType ( ) [inline]
```

Get a Component.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>component</i>	
------------------	--

6.6.2.5 RegisterComponent()

```
template<typename T >
void ecs::ComponentManager::RegisterComponent ( ) [inline]
```

Initialize a new component in ecs.

Template Parameters

<i>T</i>	
----------	--

6.6.2.6 RemoveComponent()

```
template<typename T >
void ecs::ComponentManager::RemoveComponent (
    Entity entity ) [inline]
```

Remove a component.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>entity</i>	
---------------	--

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

- [inc/ComponentManager.hpp](#)
- [src/ecs/ComponentManager.cpp](#)

6.7 ecs::Coordinator Class Reference

Combine and handling each manager.

```
#include <Coordinator.hpp>
```

Public Member Functions

- void [Init](#) ()
Initialise the coordinator.
- Entity [CreateEntity](#) ()
Create an entity.
- void [DestroyEntity](#) (Entity entity)
Destroy an entity.
- template<typename T >
void [RegisterComponent](#) ()
Register a component.
- template<typename T >
void [AddComponent](#) (Entity entity, T component)
Add a component.
- template<typename T >
void [RemoveComponent](#) (Entity entity)
Remove a component.
- template<typename T >
T & [GetComponent](#) (Entity entity)
Get a component.
- template<typename T >
ComponentType [GetComponentType](#) ()
Get a component.
- template<typename T >
std::shared_ptr< T > [RegisterSystem](#) ()
Register a system.
- template<typename T >
void [setSystemSignature](#) (Signature signature)
Set a system signature.

6.7.1 Detailed Description

Combine and handling each manager.

6.7.2 Member Function Documentation

6.7.2.1 AddComponent()

```
template<typename T >
void ecs::Coordinator::AddComponent (
    Entity entity,
    T component ) [inline]
```

Add a component.

Parameters

<i>T</i>	
----------	--

6.7.2.2 CreateEntity()

```
Entity Coordinator::CreateEntity ( )
```

Create an entity.

Returns

Entity

6.7.2.3 DestroyEntity()

```
void Coordinator::DestroyEntity (
    Entity entity )
```

Destroy an entity.

Parameters

<i>Entity</i>	<i>entity</i>
---------------	---------------

6.7.2.4 GetComponent()

```
template<typename T >
T& ecs::Coordinator::GetComponent (
    Entity entity ) [inline]
```

Get a component.

Parameters

<i>Entity</i>	
---------------	--

Returns

T

6.7.2.5 GetComponentType()

```
template<typename T >
ComponentType ecs::Coordinator::GetComponentType ( ) [inline]
```

Get a component.

Returns

ComponentType

6.7.2.6 RegisterComponent()

```
template<typename T >
void ecs::Coordinator::RegisterComponent ( ) [inline]
```

Register a component.

Parameters

<i>T</i>	
----------	--

6.7.2.7 RegisterSystem()

```
template<typename T >
std::shared_ptr<T> ecs::Coordinator::RegisterSystem ( ) [inline]
```

Register a system.

Template Parameters

<i>T</i>	
----------	--

Returns

std::shared_ptr<T>

6.7.2.8 RemoveComponent()

```
template<typename T >
void ecs::Coordinator::RemoveComponent (
    Entity entity ) [inline]
```

Remove a component.

Parameters

<i>T</i>	
----------	--

6.7.2.9 setSystemSignature()

```
template<typename T >
void ecs::Coordinator::setSystemSignature (
    Signature signature ) [inline]
```

Set a system signature.

Template Parameters

<i>T</i>	
----------	--

Parameters

<i>signature</i>	
------------------	--

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

- [inc/Coordinator.hpp](#)
- [src/ecs/Coordinator.cpp](#)

6.8 EnemyComp Class Reference

Public Attributes

- int **pos_x**
- int **pos_y**

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

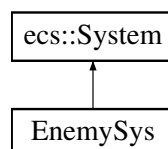
- [inc/ecs/Enemy.hpp](#)

6.9 EnemySys Class Reference

Initialise Enemies mmoves.

```
#include <Enemy.hpp>
```

Inheritance diagram for EnemySys:



Public Member Functions

- void **moove** ()

Additional Inherited Members

6.9.1 Detailed Description

Initialise Ennemies mmoves.

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

- inc/ecs/Enemy.hpp

6.10 EnemyComp Class Reference

Enemy Component.

```
#include <Enemy.hpp>
```

Public Member Functions

- [EnemyComp](#) (std::string filepath, ENEMY_TYPE type, sf::Vector2f position, uint id)
Construct a new Enemy Comp object.

Public Attributes

- ENEMY_TYPE **enemyType**
- sf::Sprite **sprite**
- uint **id**

6.10.1 Detailed Description

Enemy Component.

Initialise Ennemies pos.

6.10.2 Constructor & Destructor Documentation

6.10.2.1 EnemyComp()

```
EnemyComp::EnemyComp (
    std::string filepath,
    ENEMY_TYPE type,
    sf::Vector2f position,
    uint id )
```

Construct a new Enemy Comp object.

Parameters

<i>filepath</i>	
<i>type</i>	
<i>position</i>	
<i>id</i>	

Returns

[EnnemyComp](#)

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

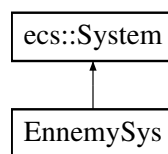
- inc/client/[Ennemy.hpp](#)
- src/client/Ennemy.cpp

6.11 EnnemySys Class Reference

Ennemy System.

```
#include <Ennemy.hpp>
```

Inheritance diagram for EnnemySys:



Public Member Functions

- void [drawEnemies](#) (std::shared_ptr< [GameContainer](#) > gameContainer)
Enemies drawer.
- std::shared_ptr< [EnnemyComp](#) > [getEnemyById](#) (int id)
Enemies comp.

Additional Inherited Members

6.11.1 Detailed Description

Ennemy System.

6.11.2 Member Function Documentation

6.11.2.1 drawEnemies()

```
void EnnemySys::drawEnemies (
    std::shared_ptr< GameContainer > gameContainer )
```

Enemies drawer.

Parameters

<i>gameContainer</i>	
----------------------	--

6.11.2.2 getEnemyById()

```
std::shared_ptr< EnemyComp > EnemySys::getEnemyById (
    int id )
```

Enemies comp.

Parameters

<i>id</i>	
-----------	--

Returns

std::shared_ptr<EnemyComp>

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

- inc/client/[Enemy.hpp](#)
- src/client/Enemy.cpp

6.12 ecs::EntityManager Class Reference

Manager handling entities.

```
#include <EntityManager.hpp>
```

Public Member Functions

- [EntityManager](#) ()
Initialise the manager.
- Entity [CreateEntity](#) ()
Create an entity.
- void [DestroyEntity](#) (Entity entity)
Destroy an entity.
- void [SetSignature](#) (Entity entity, Signature signature)
Set the signature of an entity.
- Signature [GetSignature](#) (Entity entity)
Get the signature of an entity.

6.12.1 Detailed Description

Manager handling entities.

6.12.2 Member Function Documentation

6.12.2.1 CreateEntity()

```
Entity EntityManager::CreateEntity ( )
```

Create an entity.

Returns

Entity

6.12.2.2 DestroyEntity()

```
void EntityManager::DestroyEntity (
    Entity entity )
```

Destroy an entity.

Parameters

<i>Entity</i>	<i>entity</i>
---------------	---------------

6.12.2.3 GetSignature()

```
Signature EntityManager::GetSignature (
    Entity entity )
```

Get the signature of an entity.

Parameters

<i>Entity</i>	<i>entity</i>
---------------	---------------

Returns

Signature

6.12.2.4 SetSignature()

```
void EntityManager::SetSignature (
    Entity entity,
    Signature signature )
```

Set the signature of an entity.

Parameters

<i>Entity</i>	entity, Signature signature
---------------	-----------------------------

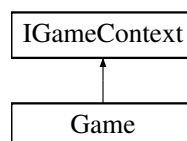
The documentation for this class was generated from the following files:

- [inc/EntityManager.hpp](#)
- [src/ecs/EntityManager.cpp](#)

6.13 Game Class Reference[Game.](#)

```
#include <GameGraphic.hpp>
```

Inheritance diagram for Game:

**Public Member Functions**

- [Game](#) ()
Game constructor.
- bool [gameLoop](#) (std::shared_ptr< [GameContainer](#) > gameContainer)
GameLoop.
- bool [eventHandler](#) (sf::Event &event)
Event handler.
- void [Init](#) (std::shared_ptr< [GameContainer](#) > gameContainer)
Initialize the game.
- void [Server](#) ()

- Initialize the game server.*
 - bool [isMenu](#) ()
Check if the menu is active.
- int [getLife](#) ()
Check remaining life.
- void [setLife](#) (int life)
Set the life.

6.13.1 Detailed Description

[Game](#).

6.13.2 Constructor & Destructor Documentation

6.13.2.1 Game()

```
Game::Game ( )
```

[Game](#) constructor.

Returns

[Game](#)

6.13.3 Member Function Documentation

6.13.3.1 eventHandler()

```
bool Game::eventHandler (
    sf::Event & event ) [virtual]
```

Event handler.

Parameters

<i>sf::Event</i>	&event
------------------	--------

Returns

bool

Implements [IGameContext](#).

6.13.3.2 gameLoop()

```
bool Game::gameLoop (
    std::shared_ptr< GameContainer > gameContainer ) [virtual]
```

GameLoop.

Parameters

<i>gameContainer</i>	
----------------------	--

Returns

bool

Implements [IGameContext](#).

6.13.3.3 getLife()

```
int Game::getLife ( )
```

Check remaining life.

Returns

int

6.13.3.4 Init()

```
void Game::Init (
    std::shared_ptr< GameContainer > gameContainer ) [virtual]
```

Initialize the game.

Parameters

<i>gameContainer</i>	
----------------------	--

Implements [IGameContext](#).

6.13.3.5 isMenu()

```
bool Game::isMenu ( ) [virtual]
```

Check if the menu is active.

Returns

bool

Implements [IGameContext](#).

6.13.3.6 Server()

```
void Game::Server ( ) [virtual]
```

Initialize the game server.

Returns

bool

Implements [IGameContext](#).

6.13.3.7 setLife()

```
void Game::setLife (
    int life )
```

Set the life.

Parameters

<i>life</i>	
-------------	--

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

- [inc/client/GameGraphic.hpp](#)
- [src/client/Game.cpp](#)

6.14 GameContainer Class Reference

[Game](#) container.

```
#include <GameContainer.hpp>
```

Public Member Functions

- [GameContainer](#) ()
Constructor of the [Game](#) Container.
- void [updateDelta](#) ()
Update Delta of the [Game](#) Container.

Public Attributes

- std::shared_ptr< sf::RenderWindow > [window](#)
Render window.
- std::shared_ptr< sf::Event > [event](#)
Event handler.
- char ** **argv**
- sf::Clock **clock**
- int32_t **delta**

6.14.1 Detailed Description

[Game](#) container.

6.14.2 Constructor & Destructor Documentation

6.14.2.1 GameContainer()

```
GameContainer::GameContainer ( )
```

Constructor of the [Game](#) Container.

Returns

[GameContainer](#)

6.14.3 Member Data Documentation

6.14.3.1 event

```
std::shared_ptr<sf::Event> GameContainer::event
```

Event handler.

Parameters

<i>event</i>	
--------------	--

Returns

`std::shared_ptr<sf::Event>`

6.14.3.2 window

`std::shared_ptr<sf::RenderWindow> GameContainer::window`

Render window.

Parameters

<i>window</i>	
---------------	--

Returns

`sf::RenderWindow`

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

- [inc/client/GameContainer.hpp](#)
- [src/client/GameContainer.cpp](#)

6.15 GameGraphic Class Reference

[GameGraphic](#).

```
#include <GameGraphic.hpp>
```

Public Member Functions

- [GameGraphic](#) ()
GameGraphic constructor.
- void [loop](#) ()
Graphic loop.
- bool [eventHandler](#) ()
Event handler.

Public Attributes

- `std::shared_ptr< GameContainer > _container`
CGet the shared container.

6.15.1 Detailed Description

[GameGraphic](#).

6.15.2 Constructor & Destructor Documentation

6.15.2.1 [GameGraphic](#)()

```
GameGraphic::GameGraphic ( )
```

[GameGraphic](#) constructor.

Returns

[GameGraphic](#)

6.15.3 Member Function Documentation

6.15.3.1 [eventHandler](#)()

```
bool GameGraphic::eventHandler ( )
```

Event handler.

Returns

bool

6.15.4 Member Data Documentation

6.15.4.1 _container

```
std::shared_ptr<GameContainer> GameGraphic::_container
```

CGet the shared container.

Returns

```
std::shared_ptr<IGameContainer>
```

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

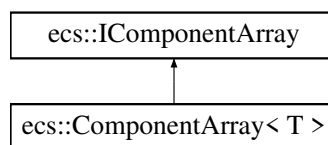
- inc/client/[GameGraphic.hpp](#)
- src/client/GameGraphic.cpp

6.16 ecs::IComponentArray Class Reference

Setup core of ArrayComponent throw Interface.

```
#include <ComponentArray.hpp>
```

Inheritance diagram for ecs::IComponentArray:



Public Member Functions

- [IComponentArray](#) ()
IComponentArray Constructor.
- virtual [~IComponentArray](#) ()=default
IComponentArray Destructor.
- virtual void [EntityDestroyed](#) (Entity entity)=0
Delete Entity.
- [IComponentArray](#) & [operator=](#) (const [IComponentArray](#) &)=delete
Delete Reference IComponentArray.
- [IComponentArray](#) (const [IComponentArray](#) &)=delete
Delete Reference IComponent.
- [IComponentArray](#) & [operator=](#) ([IComponentArray](#) &&)=delete
Delete Double Reference IComponent.

6.16.1 Detailed Description

Setup core of ArrayComponent throw Interface.

6.16.2 Constructor & Destructor Documentation

6.16.2.1 IComponentArray()

```
ecs::IComponentArray::IComponentArray (
    const IComponentArray & ) [delete]
```

Delete Reference IComponent.

Parameters

<i>const</i>	IComponentArray &
--------------	-----------------------------------

Returns

[IComponentArray](#)

6.16.3 Member Function Documentation

6.16.3.1 EntityDestroyed()

```
virtual void ecs::IComponentArray::EntityDestroyed (
    Entity entity ) [pure virtual]
```

Delete Entity.

Parameters

<i>Entity</i>	<i>entity</i>
---------------	---------------

Implemented in [ecs::ComponentArray< T >](#).

6.16.3.2 operator=() [1/2]

```
IComponentArray& ecs::IComponentArray::operator= (
    const IComponentArray & ) [delete]
```

Delete Reference [IComponentArray](#).

Parameters

<i>const</i>	IComponentArray &
--------------	-----------------------------------

Returns

[IComponentArray](#)**6.16.3.3 operator=()** [2/2]

```
IComponentArray& ecs::IComponentArray::operator= (
    IComponentArray && ) [delete]
```

Delete Double Reference IComponent.

Parameters

IComponentArray	&&
---------------------------------	----

Returns

[IComponentArray](#)

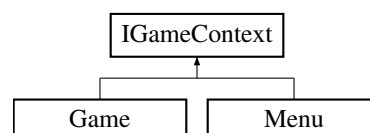
The documentation for this class was generated from the following file:

- [inc/ComponentArray.hpp](#)

6.17 IGameContext Class Reference[GameGraphic](#).

```
#include <GameGraphic.hpp>
```

Inheritance diagram for IGameContext:



Public Member Functions

- virtual bool [gameLoop](#) (std::shared_ptr< [GameContainer](#) > gameContainer)=0
GameLoop.
- virtual bool [eventHandler](#) (sf::Event &event)=0
Event handler.
- virtual bool [isMenu](#) ()=0
Check if the menu is active.
- virtual void [init](#) (std::shared_ptr< [GameContainer](#) > gameContainer)=0
Initialize the game container.
- virtual void [Server](#) ()=0
Check if the server is running.

6.17.1 Detailed Description

[GameGraphic](#).

6.17.2 Member Function Documentation

6.17.2.1 [eventHandler\(\)](#)

```
virtual bool IGameContext::eventHandler (
    sf::Event & event ) [pure virtual]
```

Event handler.

Parameters

<i>sf::Event</i>	&event
------------------	--------

Returns

virtual bool

Implemented in [Game](#), and [Menu](#).

6.17.2.2 [gameLoop\(\)](#)

```
virtual bool IGameContext::gameLoop (
    std::shared_ptr< GameContainer > gameContainer ) [pure virtual]
```

GameLoop.

Parameters

<i>gameContainer</i>	
----------------------	--

Returns

virtual bool

Implemented in [Game](#), and [Menu](#).**6.17.2.3 Init()**

```
virtual void IGameContext::Init (
    std::shared_ptr< GameContainer > gameContainer ) [pure virtual]
```

Initialize the game container.

Parameters

<i>gameContainer</i>	
----------------------	--

Returns

virtual bool

Implemented in [Game](#), and [Menu](#).**6.17.2.4 isMenu()**

```
virtual bool IGameContext::isMenu ( ) [pure virtual]
```

Check if the menu is active.

Returns

virtual bool

Implemented in [Game](#), and [Menu](#).**6.17.2.5 Server()**

```
virtual void IGameContext::Server ( ) [pure virtual]
```

Check if the server is running.

Parameters

Server	
------------------------	--

Returns

virtual bool

Implemented in [Game](#), and [Menu](#).

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

- inc/client/[GameGraphic.hpp](#)

6.18 InteractServer Class Reference

class representing the [InteractServer](#)

```
#include <interactServer.hpp>
```

Public Member Functions

- [InteractServer](#) (std::shared_ptr< [SpaceShipSys](#) > spaceShipSys, std::shared_ptr< [EnemySys](#) > [EnemySys](#))
InteractServer constructor.
- [~InteractServer](#) ()
InteractServer destructor.
- void [createServer](#) (std::shared_ptr< [GameContainer](#) > container)
Create server.
- void [StartReceive](#) ()
Start receive information.
- void [ServerLoop](#) ()
Server loop.
- void [sendServerInfos](#) ()
Send server information.
- void [test](#) (const boost::system::error_code &ec, std::size_t bytes_transferred)
test the communication between server and client
- void [getServerInfos](#) (const boost::system::error_code &ec, std::size_t bytes_transferred)
get the server informations

Public Attributes

- int **life**

6.18.1 Detailed Description

class representing the [InteractServer](#)

6.18.2 Constructor & Destructor Documentation

6.18.2.1 InteractServer()

```
InteractServer::InteractServer (
    std::shared_ptr< SpaceShipSys > spaceShipSys,
    std::shared_ptr< EnnemySys > EnnemySys )
```

[InteractServer](#) constructor.

Parameters

<i>spaceShipSys</i> , EnnemySys	
---	--

Returns

[InteractServer](#)

6.18.2.2 ~InteractServer()

```
InteractServer::~~InteractServer ( ) [inline]
```

[InteractServer](#) destructor.

Returns

[~InteractServer](#)

6.18.3 Member Function Documentation

6.18.3.1 createServer()

```
void InteractServer::createServer (
    std::shared_ptr< GameContainer > container )
```

Create server.

Parameters

<i>container</i>	
------------------	--

6.18.3.2 `getServerInfos()`

```
void InteractServer::getServerInfos (
    const boost::system::error_code & ec,
    std::size_t bytes_transferred )
```

get the server informations

Parameters

<i>ec</i>	
<i>bytes_transferred</i>	

6.18.3.3 `test()`

```
void InteractServer::test (
    const boost::system::error_code & ec,
    std::size_t bytes_transferred )
```

test the communication between server and client

Parameters

<i>ec</i>	
<i>bytes_transferred</i>	

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

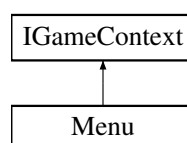
- [inc/client/interactServer.hpp](#)
- [src/client/interactServer.cpp](#)

6.19 Menu Class Reference

[Menu.](#)

```
#include <GameGraphic.hpp>
```

Inheritance diagram for Menu:



Public Member Functions

- [Menu](#) ()
Menu constructor.
- bool [gameLoop](#) (std::shared_ptr< [GameContainer](#) > gameContainer)
GameLoop.
- bool [eventHandler](#) (sf::Event &event)
Event handler.
- bool [isMenu](#) ()
Check if the menu is active.
- void [Init](#) (std::shared_ptr< [GameContainer](#) > gameContainer)
Initialize the menu.
- void [Server](#) ()
Server checker.

6.19.1 Detailed Description

[Menu](#).

6.19.2 Constructor & Destructor Documentation

6.19.2.1 Menu()

```
Menu::Menu ( )
```

[Menu](#) constructor.

Returns

[Menu](#)

6.19.3 Member Function Documentation

6.19.3.1 eventHandler()

```
bool Menu::eventHandler (
    sf::Event & event ) [virtual]
```

Event handler.

Parameters

<i>sf::Event</i>	&event
------------------	--------

Returns

bool

Implements [IGameContext](#).

6.19.3.2 gameLoop()

```
bool Menu::gameLoop (
    std::shared_ptr< GameContainer > gameContainer ) [virtual]
```

GameLoop.

Parameters

<i>gameContainer</i>	
----------------------	--

Returns

bool

Implements [IGameContext](#).

6.19.3.3 Init()

```
void Menu::Init (
    std::shared_ptr< GameContainer > gameContainer ) [inline], [virtual]
```

Initialize the menu.

Parameters

<i>gameContainer</i>	
----------------------	--

Implements [IGameContext](#).

6.19.3.4 isMenu()

```
bool Menu::isMenu ( ) [virtual]
```

Check if the menu is active.

Returns

bool

Implements [IGameContext](#).

6.19.3.5 Server()

```
void Menu::Server ( ) [inline], [virtual]
```

[Server](#) checker.

Parameters

Server	
------------------------	--

Implements [IGameContext](#).

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

- inc/client/[GameGraphic.hpp](#)
- src/client/Menu.cpp

6.20 Network Class Reference

class representing UDP & TCP Protocol [Network](#)

```
#include <Network.hpp>
```

Public Member Functions

- [Network](#) ()
Constructor of the [Network](#).
- [~Network](#) ()=default
Destructor of the [Network](#).
- void [callable](#) ()
Callable.

6.20.1 Detailed Description

class representing UDP & TCP Protocol [Network](#)

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

- inc/[Network.hpp](#)
- src/network/Network.cpp

6.21 ParallaxComp Class Reference

Parallax Component.

```
#include <Parallax.hpp>
```

Public Member Functions

- **ParallaxComp** (std::string filepath, float speedMove)

Public Attributes

- sf::Sprite **sprite1**
- sf::Sprite **sprite2**
- float **speedMove**

6.21.1 Detailed Description

Parallax Component.

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

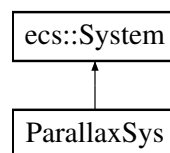
- inc/client/[Parallax.hpp](#)
- src/client/Parallax.cpp

6.22 ParallaxSys Class Reference

Parallax System.

```
#include <Parallax.hpp>
```

Inheritance diagram for ParallaxSys:



Public Member Functions

- void [MoveAndDraw](#) (std::shared_ptr< sf::RenderWindow > window)
Move and draw.

Additional Inherited Members

6.22.1 Detailed Description

Parallax System.

6.22.2 Member Function Documentation

6.22.2.1 MoveAndDraw()

```
void ParallaxSys::MoveAndDraw (
    std::shared_ptr< sf::RenderWindow > window )
```

Move and draw.

Parameters

<i>window</i>	
---------------	--

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

- inc/client/[Parallax.hpp](#)
- src/client/Parallax.cpp

6.23 Server Class Reference

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

- inc/Server.hpp
- src/server/Server.cpp

6.24 SpaceShipComp Class Reference

SpaceShip Component.

```
#include <SpaceShip.hpp>
```

Public Member Functions

- [SpaceShipComp](#) (std::string filepath, std::string filepathBullet, int nbPlayer, bool isPlayer)
Construct a new Space Ship Comp object.
- void [shoot](#) ()
Shoot function.

Public Attributes

- sf::Sprite **sprite**
- int32_t **lastShoot**
- bool **isPlayer**
- bool **isShooting**
- std::vector< sf::Sprite > **bullets**
- int **nbPlayer**

6.24.1 Detailed Description

SpaceShip Component.

6.24.2 Constructor & Destructor Documentation

6.24.2.1 SpaceShipComp()

```
SpaceShipComp::SpaceShipComp (
    std::string filepath,
    std::string filepathBullet,
    int nbPlayer,
    bool isPlayer )
```

Construct a new Space Ship Comp object.

Parameters

<i>filepath</i>	
<i>filepathBullet</i>	
<i>nbPlayer</i>	
<i>isPlayer</i>	

Returns

[SpaceShipComp](#)

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

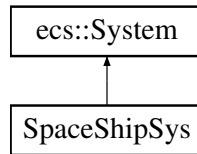
- inc/client/[SpaceShip.hpp](#)
- src/client/SpaceShip.cpp

6.25 SpaceShipSys Class Reference

SpaceShip System.


```
#include <SpaceShip.hpp>
```

Inheritance diagram for SpaceShipSys:



Public Member Functions

- void [animAndDraw](#) (std::shared_ptr< [GameContainer](#) > gameContainer)
Animation.
- void [move](#) (int32_t delta)
Move.
- void [shoot](#) (sf::Clock &clock)
Shoot.
- std::shared_ptr< [SpaceShipComp](#) > [getPlayer](#) ()
Get player.
- std::shared_ptr< [SpaceShipComp](#) > [getPlayerById](#) (int id)
Get player by id.
- void [removePlayer](#) (int id)
Remove.

Additional Inherited Members

6.25.1 Detailed Description

SpaceShip System.

6.25.2 Member Function Documentation

6.25.2.1 animAndDraw()

```
void SpaceShipSys::animAndDraw (
    std::shared_ptr< GameContainer > gameContainer )
```

Animation.

Parameters

<i>gameContainer</i>	
----------------------	--

6.25.2.2 getPlayer()

```
std::shared_ptr< SpaceShipComp > SpaceShipSys::getPlayer ( )
```

Get player.

Returns

std::shared_ptr<SpaceShipComp>

6.25.2.3 getPlayerById()

```
std::shared_ptr< SpaceShipComp > SpaceShipSys::getPlayerById (
    int id )
```

Get player by id.

Parameters

<i>id</i>	
-----------	--

Returns

std::shared_ptr<SpaceShipComp>

6.25.2.4 move()

```
void SpaceShipSys::move (
    int32_t delta )
```

Move.

Parameters

<i>delta</i>	
--------------	--

6.25.2.5 removePlayer()

```
void SpaceShipSys::removePlayer (
    int id )
```

Remove.

Parameters

<code>id</code>	>
-----------------	---

6.25.2.6 shoot()

```
void SpaceShipSys::shoot (
    sf::Clock & clock )
```

Shoot.

Parameters

<code>clock</code>	
--------------------	--

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

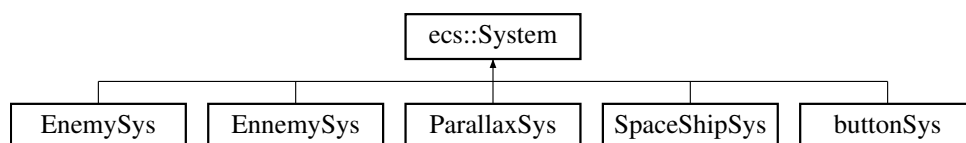
- [inc/client/SpaceShip.hpp](#)
- [src/client/SpaceShip.cpp](#)

6.26 ecs::System Class Reference

[System](#) classes.

```
#include <System.hpp>
```

Inheritance diagram for `ecs::System`:



Public Attributes

- `std::set< Entity >` [mapEntities](#)
Set entity.

6.26.1 Detailed Description

[System](#) classes.

6.26.2 Member Data Documentation

6.26.2.1 mapEntities

```
std::set<Entity> ecs::System::mapEntities
```

Set entity.

Returns

```
std::set<Entity>
```

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

- [inc/System.hpp](#)

6.27 ecs::SystemManager Class Reference

class representing the system manager

```
#include <SystemManager.hpp>
```

Public Member Functions

- `template<typename T>`
`std::shared_ptr< T > CreateSystem ()`
Create a [System](#).
- `template<typename T>`
`void SetSystemSignature (Signature signature)`
Set Binary Signature to [System](#).
- `void RemoveOldEntity (Entity entity)`
Remove a specific Entity.
- `void ChangedEntitySignature (Entity entity, Signature entitySignature)`
Change Entity Signature.

6.27.1 Detailed Description

class representing the system manager

6.27.2 Member Function Documentation

6.27.2.1 ChangedEntitySignature()

```
void SystemManager::ChangedEntitySignature (
    Entity entity,
    Signature entitySignature )
```

Change Entity Signature.

Parameters

<i>Entity</i>	entity
<i>Signature</i>	: Current Signature Entity

6.27.2.2 CreateSystem()

```
template<typename T >
std::shared_ptr<T> ecs::SystemManager::CreateSystem ( ) [inline]
```

Create a [System](#).

Returns

std::shared_ptr<T>

6.27.2.3 RemoveOldEntity()

```
void SystemManager::RemoveOldEntity (
    Entity entity )
```

Remove a specific Entity.

Parameters

<i>Entity</i>	entity
---------------	--------

6.27.2.4 SetSystemSignature()

```
template<typename T >
void ecs::SystemManager::SetSystemSignature (
    Signature signature ) [inline]
```

Set Binary Signature to [System](#).

Parameters

<i>Signature</i>	: Binary Signature
------------------	--------------------

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

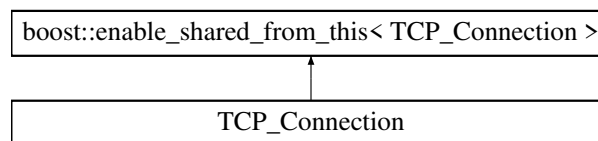
- [inc/SystemManager.hpp](#)
- [src/ecs/SystemManager.cpp](#)

6.28 TCP_Connection Class Reference

class representing the TCP Connection

```
#include <Network.hpp>
```

Inheritance diagram for TCP_Connection:



Public Types

- typedef boost::shared_ptr< [TCP_Connection](#) > **pointer**

Public Member Functions

- [TCP_Connection](#) (boost::asio::io_context &ioc)
Create TCP Connection.
- boost::asio::ip::tcp::socket & [socket](#) ()
Create TCP Socket.
- void [start](#) ()
Runs TCP Connection.
- void [handleWrite](#) (const boost::system::error_code &)
Handle Write.

6.28.1 Detailed Description

class representing the TCP Connection

6.28.2 Constructor & Destructor Documentation

6.28.2.1 TCP_Connection()

```
TCP_Connection::TCP_Connection (
    boost::asio::io_context & ioc ) [explicit]
```

Create TCP Connection.

Parameters

<code>boost::asio::io_context</code>	<code>&</code>
--------------------------------------	--------------------

Returns

explicit [TCP_Connection](#)

6.28.3 Member Function Documentation

6.28.3.1 handleWrite()

```
void TCP_Connection::handleWrite (
    const boost::system::error_code & ec )
```

Handle Write.

Parameters

<code>const</code>	<code>boost::system::error_code &</code>
--------------------	--

6.28.3.2 socket()

```
boost::asio::ip::tcp::socket & TCP_Connection::socket ( )
```

Create TCP Socket.

Returns

`boost::asio::ip::tcp::socket&`

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

- [inc/Network.hpp](#)
- [src/network/Network.cpp](#)

6.29 TCP_Server Class Reference

class representing the TCP [Server](#)

```
#include <Network.hpp>
```

Public Member Functions

- [TCP_Server](#) (boost::asio::io_context &)
Create TCP [Server](#).
- std::vector< TCP_Connection::pointer > [getConnexions](#) () const
Get Client Connection.

6.29.1 Detailed Description

class representing the TCP [Server](#)

6.29.2 Constructor & Destructor Documentation

6.29.2.1 TCP_Server()

```
TCP_Server::TCP_Server (
    boost::asio::io_context & ioc ) [explicit]
```

Create TCP [Server](#).

Parameters

<code>boost::asio::io_context&</code>	
---	--

Returns

explicit [UDP_Server](#)

6.29.3 Member Function Documentation

6.29.3.1 getConnexions()

```
std::vector< TCP_Connection::pointer > TCP_Server::getConnexions ( ) const
```

Get Client Connection.

Returns

std::vector<TCP_Connection::pointer>

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

- inc/[Network.hpp](#)
- src/network/[Network.cpp](#)

6.30 TextureComp Class Reference

Texture Component.

```
#include <Texture.hpp>
```

Public Member Functions

- [TextureComp](#) (std::string filepath)
*Construct a new Texture Comp object *.*

Public Attributes

- TEXTURE_TYPE [typeT](#)
Construct a new Texture Comp object.
- sf::Texture [texture](#)
Construct a new Texture Comp object.

6.30.1 Detailed Description

Texture Component.

6.30.2 Constructor & Destructor Documentation

6.30.2.1 TextureComp()

```
TextureComp::TextureComp (
    std::string filepath )
```

Construct a new Texture Comp object *.

Parameters

<i>filepath</i>	*
-----------------	---

Returns

[TextureComp](#)

6.30.3 Member Data Documentation

6.30.3.1 texture

```
sf::Texture TextureComp::texture
```

Construct a new Texture Comp object.

Parameters

<i>texture</i>	
----------------	--

Returns

sf::Texture

6.30.3.2 typeT

```
TEXTURE_TYPE TextureComp::typeT
```

Construct a new Texture Comp object.

Parameters

<i>typeT</i>	
--------------	--

Returns

TEXTURE_TYPE

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

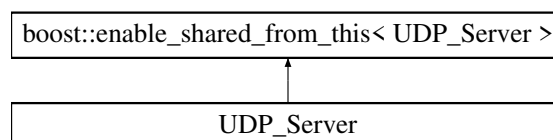
- [inc/client/Texture.hpp](#)
- [src/client/Enemy.cpp](#)

6.31 UDP_Server Class Reference

class representing the [UDP_Server](#)

```
#include <Network.hpp>
```

Inheritance diagram for UDP_Server:



Public Member Functions

- [UDP_Server](#) (boost::asio::io_context &)
Destructor of the [Network](#).

6.31.1 Detailed Description

class representing the [UDP_Server](#)

6.31.2 Constructor & Destructor Documentation

6.31.2.1 UDP_Server()

```
UDP_Server::UDP_Server (
    boost::asio::io_context & ioc ) [explicit]
```

Destructor of the [Network](#).

Parameters

<i>boost::asio::io_context&</i>	
-------------------------------------	--

Returns

explicit [UDP_Server](#)

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

- inc/[Network.hpp](#)
- src/network/[Network.cpp](#)

Chapter 7

File Documentation

7.1 inc/client/Bullets.hpp File Reference

Bullets.

7.1.1 Detailed Description

Bullets.

7.2 inc/client/Button.hpp File Reference

Button.

```
#include "System.hpp"
#include "Coordinator.hpp"
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <memory>
#include <string>
```

Classes

- class [ButtonComp](#)
Button.
- class [buttonSys](#)

Variables

- [ecs::Coordinator](#) [gCoordinator](#)

7.2.1 Detailed Description

Button.

7.3 inc/client/Enemy.hpp File Reference

Enemy.

```
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <string>
#include <algorithm>
#include "Coordinator.hpp"
#include "client/GameContainer.hpp"
#include "client/Texture.hpp"
```

Classes

- class [EnemyComp](#)
Enemy Component.
- class [EnemySys](#)
Enemy System.

Enumerations

- enum `ENEMY_TYPE { BASIC_ENEMY }`

7.3.1 Detailed Description

Enemy.

7.4 inc/client/GameContainer.hpp File Reference

[GameContainer](#).

```
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
```

Classes

- class [GameContainer](#)
Game container.

7.4.1 Detailed Description

[GameContainer.](#)

7.5 inc/client/GameGraphic.hpp File Reference

[GameGraphic.](#)

```
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include "Coordinator.hpp"
#include "client/Button.hpp"
#include "client/Parallax.hpp"
#include "client/SpaceShip.hpp"
#include "client/GameContainer.hpp"
#include "client/interactServer.hpp"
#include "client/Enemy.hpp"
```

Classes

- class [IGameContext](#)
[GameGraphic.](#)
- class [Menu](#)
[Menu.](#)
- class [Game](#)
[Game.](#)
- class [GameGraphic](#)
[GameGraphic.](#)

Variables

- [ecs::Coordinator](#) [gCoordinator](#)

7.5.1 Detailed Description

[GameGraphic.](#)

7.6 inc/client/interactServer.hpp File Reference

[interactServer](#)

```
#include <iostream>
#include <boost/asio.hpp>
#include <boost/array.hpp>
#include <boost/bind.hpp>
#include "client/SpaceShip.hpp"
#include "client/Enemy.hpp"
```

Classes

- class [InteractServer](#)
class representing the [InteractServer](#)

7.6.1 Detailed Description

InteractServer

7.7 inc/client/Parallax.hpp File Reference

Parallax.

```
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <stdexcept>
#include "Coordinator.hpp"
```

Classes

- class [ParallaxComp](#)
Parallax Component.
- class [ParallaxSys](#)
Parallax System.

Variables

- [ecs::Coordinator](#) gCoordinator

7.7.1 Detailed Description

Parallax.

7.8 inc/client/Screens.hpp File Reference

Screens.

7.8.1 Detailed Description

Screens.

7.9 inc/client/SpaceShip.hpp File Reference

SpaceShip.

```
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <string>
#include <algorithm>
#include "Coordinator.hpp"
#include "client/GameContainer.hpp"
```

Classes

- class [SpaceShipComp](#)
SpaceShip Component.
- class [SpaceShipSys](#)
SpaceShip System.

7.9.1 Detailed Description

SpaceShip.

7.10 inc/client/Texture.hpp File Reference

Texture.

```
#include <memory>
#include <SFML/Graphics.hpp>
#include <SFML/Window.hpp>
#include <SFML/Audio.hpp>
#include <string>
#include <algorithm>
#include "Coordinator.hpp"
#include "client/GameContainer.hpp"
```

Classes

- class [TextureComp](#)
Texture Component.

Enumerations

- enum **TEXTURE_TYPE** { **ENEMY** }

7.10.1 Detailed Description

Texture.

7.11 inc/Component.hpp File Reference

Component Type.

```
#include <stdint>
```

Typedefs

- using **ComponentType** = std::uint8_t

Variables

- const ComponentType **MAX_COMPONENTS** = 32

7.11.1 Detailed Description

Component Type.

7.12 inc/ComponentArray.hpp File Reference

Array holding components.

```
#include "Component.hpp"
#include "Entity.hpp"
#include "EntityManager.hpp"
#include <cassert>
#include <unordered_map>
```

Classes

- class [ecs::IComponentArray](#)
Setup core of ArrayComponent throw Interface.
- class [ecs::ComponentArray< T >](#)
Array holding components Inherits from [IComponentArray](#) Class.

Namespaces

- [ecs](#)
Entity Component [System](#).

7.12.1 Detailed Description

Array holding components.

7.13 inc/ComponentManager.hpp File Reference

Manager handling components.

```
#include <memory>
#include "Component.hpp"
#include "ComponentArray.hpp"
#include "Entity.hpp"
#include "Signature.hpp"
```

Classes

- class [ecs::ComponentManager](#)
Manager handling components.

Namespaces

- [ecs](#)
Entity Component [System](#).

7.13.1 Detailed Description

Manager handling components.

7.14 inc/Coordinator.hpp File Reference

Combine and handling each manager.

```
#include "ComponentManager.hpp"
#include "EntityManager.hpp"
#include "SystemManager.hpp"
```

Classes

- class [ecs::Coordinator](#)
Combine and handling each manager.

Namespaces

- [ecs](#)

Entity Component [System](#).

Variables

- [ecs::Coordinator](#) `gCoordinator`

7.14.1 Detailed Description

Combine and handling each manager.

7.15 inc/Entity.hpp File Reference

Entity Type.

```
#include <stdint>
```

Typedefs

- using **Entity** = std::uint32_t

Variables

- const Entity **MAX_ENTITIES** = 5000

7.15.1 Detailed Description

Entity Type.

7.16 inc/EntityManager.hpp File Reference

Manager handling entities.

```
#include <array>
#include <iostream>
#include <stdlib.h>
#include <queue>
#include "Entity.hpp"
#include "Signature.hpp"
```

Classes

- class [ecs::EntityManager](#)
Manager handling entities.

Namespaces

- [ecs](#)
Entity Component [System](#).

7.16.1 Detailed Description

Manager handling entities.

7.17 inc/Network.hpp File Reference

UDP & TCP Protocol with Boost::Asio.

```
#include <boost/asio.hpp>
#include <boost/array.hpp>
#include <boost/bind.hpp>
#include <iostream>
#include <boost/enable_shared_from_this.hpp>
#include <thread>
```

Classes

- class [TCP_Connection](#)
class representing the TCP Connection
- class [TCP_Server](#)
class representing the TCP [Server](#)
- class [UDP_Server](#)
class representing the [UDP_Server](#)
- class [Network](#)
class representing UDP & TCP Protocol [Network](#)

7.17.1 Detailed Description

UDP & TCP Protocol with Boost::Asio.

7.18 inc/Signature.hpp File Reference

Binary signature.

```
#include <bitset>
#include "Component.hpp"
```

Typedefs

- using **Signature** = std::bitset< MAX_COMPONENTS >

7.18.1 Detailed Description

Binary signature.

7.19 inc/System.hpp File Reference

System.

```
#include <set>
#include <memory>
#include "Entity.hpp"
#include "Component.hpp"
```

Classes

- class [ecs::System](#)
System classs.

Namespaces

- [ecs](#)
Entity Component [System](#).

7.19.1 Detailed Description

System.

7.20 inc/SystemManager.hpp File Reference

Manager handling systems.

```
#include <memory>
#include <bitset>
#include <cassert>
#include <unordered_map>
#include <stdexcept>
#include "Entity.hpp"
#include "Signature.hpp"
#include "System.hpp"
```

Classes

- class [ecs::SystemManager](#)
class representing the system manager

Namespaces

- [ecs](#)
Entity Component [System](#).

7.20.1 Detailed Description

Manager handling systems.

Index

- `_container`
 - `GameGraphic`, [34](#)
 - `~InteractServer`
 - `InteractServer`, [41](#)
- `AddComponent`
 - `ecs::ComponentManager`, [16](#)
 - `ecs::Coordinator`, [19](#)
- `animAndDraw`
 - `SpaceShipSys`, [49](#)
- `ButtonComp`, [11](#)
 - `ButtonComp`, [11](#)
 - `sprite`, [12](#)
 - `texture`, [12](#)
- `ButtonSys`, [13](#)
- `buttonSys`, [13](#)
- `ChangedEntitySignature`
 - `ecs::SystemManager`, [52](#)
- `client`, [13](#)
- `CreateEntity`
 - `ecs::Coordinator`, [21](#)
 - `ecs::EntityManager`, [27](#)
- `createServer`
 - `InteractServer`, [41](#)
- `CreateSystem`
 - `ecs::SystemManager`, [53](#)
- `DestroyEntity`
 - `ecs::Coordinator`, [21](#)
 - `ecs::EntityManager`, [27](#)
- `drawEnemies`
 - `EnnemySys`, [25](#)
- `ecs`, [9](#)
- `ecs::ComponentArray< T >`, [14](#)
 - `EntityDestroyed`, [14](#)
 - `GetData`, [15](#)
 - `InsertData`, [15](#)
 - `RemoveData`, [15](#)
- `ecs::ComponentManager`, [16](#)
 - `AddComponent`, [16](#)
 - `EntityDestroyed`, [17](#)
 - `GetComponent`, [17](#)
 - `GetComponentType`, [17](#)
 - `RegisterComponent`, [18](#)
 - `RemoveComponent`, [18](#)
- `ecs::Coordinator`, [19](#)
 - `AddComponent`, [19](#)
 - `CreateEntity`, [21](#)
- `DestroyEntity`, [21](#)
- `GetComponent`, [21](#)
- `GetComponentType`, [21](#)
- `RegisterComponent`, [22](#)
- `RegisterSystem`, [22](#)
- `RemoveComponent`, [22](#)
- `setSystemSignature`, [23](#)
- `ecs::EntityManager`, [26](#)
 - `CreateEntity`, [27](#)
 - `DestroyEntity`, [27](#)
 - `GetSignature`, [27](#)
 - `SetSignature`, [28](#)
- `ecs::IComponentArray`, [35](#)
 - `EntityDestroyed`, [36](#)
 - `IComponentArray`, [36](#)
 - `operator=`, [36](#), [37](#)
- `ecs::System`, [51](#)
 - `mapEntities`, [52](#)
- `ecs::SystemManager`, [52](#)
 - `ChangedEntitySignature`, [52](#)
 - `CreateSystem`, [53](#)
 - `RemoveOldEntity`, [53](#)
 - `SetSystemSignature`, [53](#)
- `EnemyComp`, [23](#)
- `EnemySys`, [23](#)
- `EnnemyComp`, [24](#)
 - `EnnemyComp`, [24](#)
- `EnnemySys`, [25](#)
 - `drawEnemies`, [25](#)
 - `getEnemyByld`, [26](#)
- `EntityDestroyed`
 - `ecs::ComponentArray< T >`, [14](#)
 - `ecs::ComponentManager`, [17](#)
 - `ecs::IComponentArray`, [36](#)
- `event`
 - `GameContainer`, [32](#)
- `eventHandler`
 - `Game`, [29](#)
 - `GameGraphic`, [34](#)
 - `IGameContext`, [38](#)
 - `Menu`, [43](#)
- `Game`, [28](#)
 - `eventHandler`, [29](#)
 - `Game`, [29](#)
 - `gameLoop`, [29](#)
 - `getLife`, [30](#)
 - `Init`, [30](#)
 - `isMenu`, [30](#)
 - `Server`, [31](#)

- setLife, 31
- GameContainer, 31
 - event, 32
 - GameContainer, 32
 - window, 33
- GameGraphic, 33
 - _container, 34
 - eventHandler, 34
 - GameGraphic, 34
- gameLoop
 - Game, 29
 - IGameContext, 38
 - Menu, 44
- GetComponent
 - ecs::ComponentManager, 17
 - ecs::Coordinator, 21
- GetComponentType
 - ecs::ComponentManager, 17
 - ecs::Coordinator, 21
- getConnections
 - TCP_Server, 56
- GetData
 - ecs::ComponentArray< T >, 15
- getEnemyById
 - EnemySys, 26
- getLife
 - Game, 30
- getPlayer
 - SpaceShipSys, 50
- getPlayerById
 - SpaceShipSys, 50
- getServerInfos
 - InteractServer, 42
- GetSignature
 - ecs::EntityManager, 27
- handleWrite
 - TCP_Connection, 55
- IComponentArray
 - ecs::IComponentArray, 36
- IGameContext, 37
 - eventHandler, 38
 - gameLoop, 38
 - Init, 39
 - isMenu, 39
 - Server, 39
- inc/client/Bullets.hpp, 61
- inc/client/Button.hpp, 61
- inc/client/Enemy.hpp, 62
- inc/client/GameContainer.hpp, 62
- inc/client/GameGraphic.hpp, 63
- inc/client/interactServer.hpp, 63
- inc/client/Parallax.hpp, 64
- inc/client/Screens.hpp, 64
- inc/client/SpaceShip.hpp, 65
- inc/client/Texture.hpp, 65
- inc/Component.hpp, 66
- inc/ComponentArray.hpp, 66
- inc/ComponentManager.hpp, 67
- inc/Coordinator.hpp, 67
- inc/Entity.hpp, 68
- inc/EntityManager.hpp, 68
- inc/Network.hpp, 69
- inc/Signature.hpp, 69
- inc/System.hpp, 70
- inc/SystemManager.hpp, 70
- Init
 - Game, 30
 - IGameContext, 39
 - Menu, 44
- InsertData
 - ecs::ComponentArray< T >, 15
- InteractServer, 40
 - ~InteractServer, 41
 - createServer, 41
 - getServerInfos, 42
 - InteractServer, 41
 - test, 42
- isMenu
 - Game, 30
 - IGameContext, 39
 - Menu, 44
- mapEntities
 - ecs::System, 52
- Menu, 42
 - eventHandler, 43
 - gameLoop, 44
 - Init, 44
 - isMenu, 44
 - Menu, 43
 - Server, 45
- move
 - SpaceShipSys, 50
- MoveAndDraw
 - ParallaxSys, 47
- Network, 45
- operator=
 - ecs::IComponentArray, 36, 37
- ParallaxComp, 46
- ParallaxSys, 46
 - MoveAndDraw, 47
- RegisterComponent
 - ecs::ComponentManager, 18
 - ecs::Coordinator, 22
- RegisterSystem
 - ecs::Coordinator, 22
- RemoveComponent
 - ecs::ComponentManager, 18
 - ecs::Coordinator, 22
- RemoveData
 - ecs::ComponentArray< T >, 15
- RemoveOldEntity

- ecs::SystemManager, [53](#)
- removePlayer
 - SpaceShipSys, [50](#)
- Server, [47](#)
 - Game, [31](#)
 - IGameContext, [39](#)
 - Menu, [45](#)
- setLife
 - Game, [31](#)
- SetSignature
 - ecs::EntityManager, [28](#)
- SetSystemSignature
 - ecs::SystemManager, [53](#)
- setSystemSignature
 - ecs::Coordinator, [23](#)
- shoot
 - SpaceShipSys, [51](#)
- socket
 - TCP_Connection, [55](#)
- SpaceShipComp, [47](#)
 - SpaceShipComp, [48](#)
- SpaceShipSys, [48](#)
 - animAndDraw, [49](#)
 - getPlayer, [50](#)
 - getPlayerById, [50](#)
 - move, [50](#)
 - removePlayer, [50](#)
 - shoot, [51](#)
- sprite
 - ButtonComp, [12](#)
- TCP_Connection, [54](#)
 - handleWrite, [55](#)
 - socket, [55](#)
 - TCP_Connection, [54](#)
- TCP_Server, [55](#)
 - getConnections, [56](#)
 - TCP_Server, [56](#)
- test
 - InteractServer, [42](#)
- texture
 - ButtonComp, [12](#)
 - TextureComp, [57](#)
- TextureComp, [57](#)
 - texture, [57](#)
 - TextureComp, [57](#)
 - typeT, [58](#)
- typeT
 - TextureComp, [58](#)
- UDP_Server, [58](#)
 - UDP_Server, [59](#)
- window
 - GameContainer, [33](#)