

Moon

1.0.0

Generado por Doxygen 1.8.17

1 Índice de namespaces

1.1 Lista de 'namespaces'

Lista de toda la documentación de los 'namespaces', con una breve descripción:

Moon::Alias	
Namespace de los alias del motor	??
Moon::Core	
Namespace del core del Motor	??

2 Índice jerárquico

2.1 Jerarquía de la clase

Esta lista de herencias esta ordenada aproximadamente por orden alfabético:

Moon::Core::ComponentBase_t	??
Moon::Core::Component_t< Type >	??
Moon::Core::ComponentBaseVect_t	??
Moon::Core::ComponentVect_t< CMP_t >	??
Moon::Core::ComponentStorage_t	??
Moon::Core::EntityBase_t	??
Moon::Core::Entity_t< Type >	??
Moon::Core::GameContextBase_t	??
Moon::Core::GameContext_t< Type >	??
Moon::Core::SystemBase_t	??
Moon::Core::System_t< Type >	??

3 Índice de clases

3.1 Lista de clases

Lista de las clases, estructuras, uniones e interfaces con una breve descripción:

Moon::Core::Component_t< Type >	
Clase de la que heredan todos los Components del Motor	??
Moon::Core::ComponentBase_t	
Clase de la que heredan todos los components del Motor	??

Moon::Core::ComponentBaseVect_t	
Wrapper de un Vector de Components	??
Moon::Core::ComponentStorage_t	
Es un almacen de Vectores de Components	??
Moon::Core::ComponentVect_t< CMP_t >	
Wrapper de un Vector de Components especificos	??
Moon::Core::Entity_t< Type >	
Clase de la que hereda cada entity del Motor	??
Moon::Core::EntityBase_t	
Clase de la que hereda cualquier entity del Motor	??
Moon::Core::GameContext_t< Type >	
Clase GameContext !!! Es la encargada de almacenar Los datos de las Entity y los Components	??
Moon::Core::GameContextBase_t	
Clase GameContext !!! Es la encargada de almacenar Los datos de una App	??
Moon::Core::System_t< Type >	
Clase de la que heredan todos los Systems del Motor	??
Moon::Core::SystemBase_t	
Clase de la que heredan todos los sistemas del Motor	??

4 Documentación de namespaces

4.1 Referencia del Namespace Moon::Alias

Namespace de los alias del motor.

typedefs

- using **EntityId** = std::size_t
Tipo definido para dar un ID a cada entidad.
- using **EntityType** = std::uint16_t
Tipos definidos para identificar cada tipo distinto de Objeto en el Motor.
- using **SystemType** = std::uint16_t
- using **ComponentType** = std::uint16_t
- using **GameContextType** = std::uint16_t
- using **GameContextVoidType** = std::uint16_t
- using **StateType** = std::uint16_t
- using **EventType** = std::uint16_t
- using **ManagerType** = std::uint16_t
- using **AplicationType** = std::uint16_t
- using **WindowType** = std::uint16_t
- using **EventCode** = std::uint16_t
Tipo para identificar los eventos en el Motor.

4.1.1 Descripción detallada

Namespace de los alias del motor.

4.1.2 Documentación de los 'typedefs'

4.1.2.1 EntityId `using Moon::Alias::EntityId = typedef std::size_t`

Tipo definido para dar un ID a cada entidad.

Definición en la línea 16 del archivo [alias.hpp](#).

4.1.2.2 EntityType `using Moon::Alias::EntityType = typedef std::uint16_t`

Tipos definidos para identificar cada tipo distinto de Objeto en el Motor.

Definición en la línea 23 del archivo [alias.hpp](#).

4.1.2.3 EventCode `using Moon::Alias::EventCode = typedef std::uint16_t`

Tipo para identificar los eventos en el Motor.

Definición en la línea 38 del archivo [alias.hpp](#).

4.2 Referencia del Namespace Moon::Core

Namespace del core del Motor.

Clases

- struct [Component_t](#)
Clase de la que heredan todos los Components del Motor.
- struct [ComponentBase_t](#)
Clase de la que heredan todos los components del Motor.
- struct [ComponentBaseVect_t](#)
Wrapper de un Vector de Components.
- struct [ComponentStorage_t](#)
Es un almacen de Vectores de Components.
- struct [ComponentVect_t](#)
Wrapper de un Vector de Components específicos.
- struct [Entity_t](#)
Clase de la que hereda cada entity del Motor.
- struct [EntityBase_t](#)
Clase de la que hereda cualquier entity del Motor.
- struct [GameContext_t](#)
Clase GameContext !!! Es la encargada de almacenar Los datos de las ENTity y los Components.
- struct [GameContextBase_t](#)
Clase GameContext !!! Es la encargada de almacenar Los datos de una App.
- struct [System_t](#)
Clase de la que heredan todos los Systems del Motor.
- struct [SystemBase_t](#)
Clase de la que heredan todos los sistemas del Motor.

typedefs

- using **MapCmps_t** = std::unordered_map< Moon::Alias::ComponentType, [ComponentBase_t](#) * >

4.2.1 Descripción detallada

Namespace del core del Motor.

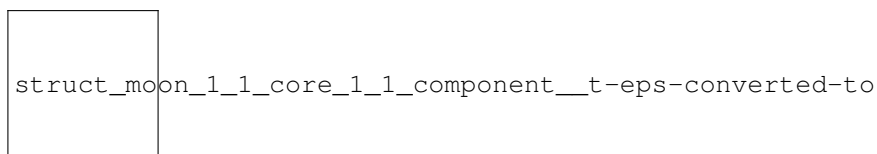
5 Documentación de las clases

5.1 Referencia de la plantilla de la Estructura Moon::Core::Component_t< Type >

Clase de la que heredan todos los Components del Motor.

```
#include <cmp.hpp>
```

Diagrama de herencias de Moon::Core::Component_t< Type >



Métodos públicos

- [Component_t](#) ()
Construct a new [Component_t](#) object.
- [Component_t](#) (Moon::Alias::EntityId eid)
Construct a new [Component_t](#) object.
- virtual ~[Component_t](#) ()=0
Destroy the [Component_t](#) object.

Métodos públicos estáticos

- static Moon::Alias::ComponentType [GetComponentType](#) () noexcept
Get the Component Type object.

Atributos públicos

- [Moon::Alias::EntityId eid](#) = 0

Otros miembros heredados

5.1.1 Descripción detallada

```
template<class Type>
struct Moon::Core::Component_t< Type >
```

Clase de la que heredan todos los Components del Motor.

Parámetros del template

Type	
------	--

Definición en la línea 31 del archivo [cmp.hpp](#).

5.1.2 Documentación del constructor y destructor

5.1.2.1 Component_t() [1/2] `template<class Type >`
`Moon::Core::Component_t< Type >::Component_t ()`

Construct a new [Component_t](#) object.

5.1.2.2 Component_t() [2/2] `template<class Type >`
`Moon::Core::Component_t< Type >::Component_t (`
`Moon::Alias::EntityId eid)`

Construct a new [Component_t](#) object.

Parámetros

eid	Id de la ENtity a la que pertenece el Component
-----	---

5.1.2.3 ~Component_t() `template<class Type >`
`virtual Moon::Core::Component_t< Type >::~~Component_t () [pure virtual]`

Destroy the [Component_t](#) object.

5.1.3 Documentación de las funciones miembro

5.1.3.1 GetComponentType() `template<class Type >`
`static Moon::Alias::ComponentType Moon::Core::Component_t< Type >::GetComponentType () [static],`
`[noexcept]`

Get the Component Type object.

Devuelve

Moon::Alias::ComponentType

5.1.4 Documentación de los datos miembro

5.1.4.1 eid `template<class Type >`
`Moon::Alias::EntityId Moon::Core::Component_t< Type >::eid = 0`

ID de la Entity a la que pertenece el Component

Definición en la línea 34 del archivo `cmp.hpp`.

La documentación para esta estructura fue generada a partir del siguiente fichero:

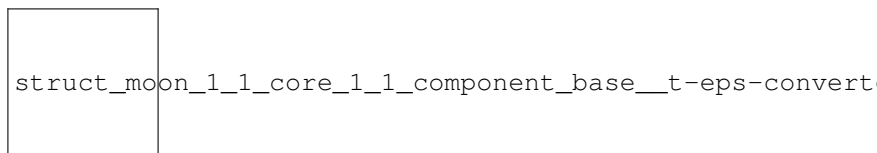
- `/home/rafa/git/Moon/include/cmp/cmp.hpp`

5.2 Referencia de la Estructura Moon::Core::ComponentBase_t

Clase de la que heredan todos los components del Motor.

```
#include <cmp_base.hpp>
```

Diagrama de herencias de Moon::Core::ComponentBase_t



Métodos públicos

- `ComponentBase_t()`
Construct a new `ComponentBase_t` object.
- `virtual ~ComponentBase_t()=0`
Destroy the `ComponentBase_t` object.

Atributos públicos estáticos

- `static Moon::Alias::ComponentType nextType = 0`

5.2.1 Descripción detallada

Clase de la que heredan todos los components del Motor.

Definición en la línea 25 del archivo `cmp_base.hpp`.

5.2.2 Documentación del constructor y destructor

5.2.2.1 ComponentBase_t() Moon::Core::ComponentBase_t::ComponentBase_t ()

Construct a new [ComponentBase_t](#) object.

5.2.2.2 ~ComponentBase_t() virtual Moon::Core::ComponentBase_t::~~ComponentBase_t () [pure virtual]

Destroy the [ComponentBase_t](#) object.

5.2.3 Documentación de los datos miembro

5.2.3.1 nextType Moon::Alias::ComponentType Moon::Core::ComponentBase_t::nextType = 0 [inline], [static]

Identificador del siguiente tipo de Component

Definición en la línea 28 del archivo [cmp_base.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

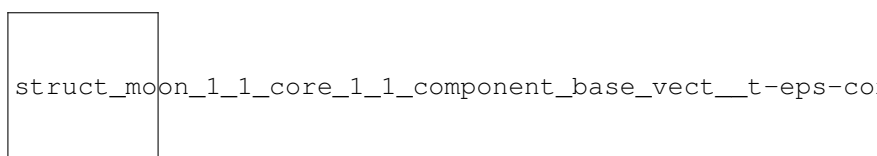
- [/home/rafa/git/Moon/include/cmp/cmp_base.hpp](#)

5.3 Referencia de la Estructura Moon::Core::ComponentBaseVect_t

Wrapper de un Vector de Components.

```
#include <cmp_vect.hpp>
```

Diagrama de herencias de Moon::Core::ComponentBaseVect_t



Métodos públicos

- virtual [~ComponentBaseVect_t](#) ()=default
Destructor de [ComponentBaseVect_t](#) object.

5.3.1 Descripción detallada

Wrapper de un Vector de Components.

Definición en la línea 27 del archivo [cmp_vect.hpp](#).

5.3.2 Documentación del constructor y destructor

5.3.2.1 ~ComponentBaseVect_t() `virtual Moon::Core::ComponentBaseVect_t::~~ComponentBaseVect_t
() [virtual], [default]`

Destructor de [ComponentBaseVect_t](#) object.

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `/home/rafa/git/Moon/include/cmp/cmp_vect.hpp`

5.4 Referencia de la Estructura Moon::Core::ComponentStorage_t

Es un almacen de Vectores de Components.

```
#include <cmp_storage.hpp>
```

Métodos públicos

- [ComponentStorage_t](#) ()
Contructor de [ComponentStorage_t](#) object.
- `template<Moon::Concepts::Cmp_t CMP_t, typename... Ts>
CMP_t & createComponent (Moon::Alias::EntityId eid, Ts &&... args)`
Create a Component object.
- `template<Moon::Concepts::Cmp_t CMP_t>
std::vector< CMP_t > & getComponents ()`
Get the Components object.
- `template<Moon::Concepts::Cmp_t CMP_t>
const std::vector< CMP_t > & getComponents () const`
Get the Components object.
- `template<Moon::Concepts::Cmp_t CMP_t>
std::vector< CMP_t > & createComponentVector ()`
Create a Component Vector object.

Atributos públicos

- `std::unordered_map< Moon::Alias::ComponentType, std::unique_ptr< ComponentBaseVect_t > > storage`

5.4.1 Descripción detallada

Es un almacen de Vectores de Components.

Definición en la línea 39 del archivo [cmp_storage.hpp](#).

5.4.2 Documentación del constructor y destructor

5.4.2.1 ComponentStorage_t() Moon::Core::ComponentStorage_t::ComponentStorage_t ()

Contructor de [ComponentStorage_t](#) object.

5.4.3 Documentación de las funciones miembro

5.4.3.1 createComponent() template<Moon::Concepts::Cmp_t CMP_t, typename... Ts>

```
CMP_t& Moon::Core::ComponentStorage_t::createComponent (
    Moon::Alias::EntityId eid,
    Ts &&... args )
```

Create a Component object.

Parámetros del template

<i>CMP_t</i>	
<i>P_t</i>	

Parámetros

<i>eid</i>	
------------	--

Devuelve

CMP_t&

5.4.3.2 createComponentVector() template<Moon::Concepts::Cmp_t CMP_t>

```
std::vector<CMP_t>& Moon::Core::ComponentStorage_t::createComponentVector ( )
```

Create a Component Vector object.

Parámetros del template

$CM \leftrightarrow$ P_t	
--------------------------------	--

Devuelve

`std::vector<CMP_t>&`

5.4.3.3 `getComponents()` [1/2] `template<Moon::Concepts::Cmp_t CMP_t>`
`std::vector<CMP_t>& Moon::Core::ComponentStorage_t::getComponents ()`

Get the Components object.

Parámetros del template

$CM \leftrightarrow$ P_t	
--------------------------------	--

Devuelve

`std::vector<CMP_t>&`

5.4.3.4 `getComponents()` [2/2] `template<Moon::Concepts::Cmp_t CMP_t>`
`const std::vector<CMP_t>& Moon::Core::ComponentStorage_t::getComponents () const`

Get the Components object.

Parámetros del template

$CM \leftrightarrow$ P_t	
--------------------------------	--

Devuelve

`const std::vector<CMP_t>&`

5.4.4 Documentación de los datos miembro

5.4.4.1 storage `std::unordered_map<Moon::Alias::ComponentType, std::unique_ptr<ComponentBaseVect_t>
> Moon::Core::ComponentStorage_t::storage`

Mapa de tipos y Vector<Components>

Definición en la línea 78 del archivo [cmp_storage.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

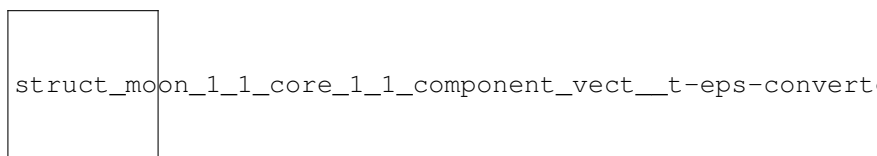
- /home/rafa/git/Moon/include/cmp/cmp_storage.hpp

5.5 Referencia de la plantilla de la Estructura Moon::Core::ComponentVect_t< CMP_t >

Wrapper de un Vector de Components específicos.

```
#include <cmp_vect.hpp>
```

Diagrama de herencias de Moon::Core::ComponentVect_t< CMP_t >



Atributos públicos

- `std::vector< CMP_t > components`

Otros miembros heredados

5.5.1 Descripción detallada

```
template<Moon::Concepts::Cmp_t CMP_t>  
struct Moon::Core::ComponentVect_t< CMP_t >
```

Wrapper de un Vector de Components específicos.

Parámetros del template

$CM \leftrightarrow P_t$	
--------------------------	--

Definición en la línea 42 del archivo [cmp_vect.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

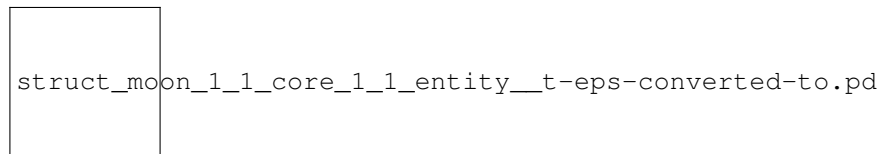
- /home/rafa/git/Moon/include/cmp/cmp_vect.hpp

5.6 Referencia de la plantilla de la Estructura Moon::Core::Entity_t< Type >

Clase de la que hereda cada entity del Motor.

```
#include <ent.hpp>
```

Diagrama de herencias de Moon::Core::Entity_t< Type >



Métodos públicos

- [Entity_t \(\)](#)
Construct a new [Entity_t](#) object.
- [Entity_t \(Moon::Alias::EntityId eid\)](#)
Construct a new [Entity_t](#) object.
- virtual [~Entity_t \(\)=0](#)
Destroy the [Entity_t](#) object.

Métodos públicos estáticos

- static [Moon::Alias::EntityType getEntityType \(\)](#) noexcept
Get the Entity Type object.

Atributos públicos

- [Moon::Alias::EntityId eid](#) = 0

Otros miembros heredados

5.6.1 Descripción detallada

```
template<class Type>
struct Moon::Core::Entity_t< Type >
```

Clase de la que hereda cada entity del Motor.

Parámetros del template

<i>Type</i>	
-------------	--

Definición en la línea [28](#) del archivo [ent.hpp](#).

5.6.2 Documentación del constructor y destructor

5.6.2.1 Entity_t() [1/2] template<class Type >

`Moon::Core::Entity_t< Type >::Entity_t ()`

Construct a new `Entity_t` object.

5.6.2.2 Entity_t() [2/2] template<class Type >

`Moon::Core::Entity_t< Type >::Entity_t (Moon::Alias::EntityId eid)`

Construct a new `Entity_t` object.

Parámetros

<i>eid</i>	
------------	--

5.6.2.3 ~Entity_t() template<class Type >

`virtual Moon::Core::Entity_t< Type >::~~Entity_t () [pure virtual]`

Destroy the `Entity_t` object.

5.6.3 Documentación de las funciones miembro

5.6.3.1 getEntityType() template<class Type >

`static Moon::Alias::EntityType Moon::Core::Entity_t< Type >::getEntityType () [static], [noexcept]`

Get the Entity Type object.

Devuelve

`Moon::Alias::EntityType`

5.6.4 Documentación de los datos miembro

```
5.6.4.1 eid template<class Type >
Moon::Alias::EntityId Moon::Core::Entity_t< Type >::eid = 0
```

ID de la Entity

Definición en la línea 31 del archivo [ent.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

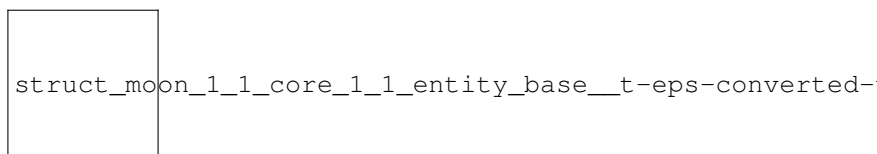
- /home/rafa/git/Moon/include/ent/ent.hpp

5.7 Referencia de la Estructura Moon::Core::EntityBase_t

Clase de la que hereda cualquier entity del Motor.

```
#include <ent_base.hpp>
```

Diagrama de herencias de Moon::Core::EntityBase_t



Métodos públicos

- [EntityBase_t](#) ()=default
Construct a new [EntityBase_t](#) object.
- virtual [~EntityBase_t](#) ()
Destroy the [EntityBase_t](#) object.
- template<class CMP_t >
requires std::is_base_of< [Moon::Core::ComponentBase_t](#), CMP_t >::value void [addComponent](#) (CMP_
t *cmp)
Añade un Component a this.
- template<class CMP_t >
requires std::is_base_of< [Moon::Core::ComponentBase_t](#), CMP_t >::value CMP_t * [getComponent](#) ()
Get the Component object.

Atributos públicos

- MapCmps_t [components](#)

Atributos públicos estáticos

- static [Moon::Alias::EntityType](#) nextType = 0

5.7.1 Descripción detallada

Clase de la que hereda cualquier entity del Motor.

Definición en la línea 34 del archivo [ent_base.hpp](#).

5.7.2 Documentación del constructor y destructor

5.7.2.1 EntityBase_t() `Moon::Core::EntityBase_t::EntityBase_t () [default]`

Construct a new [EntityBase_t](#) object.

5.7.2.2 ~EntityBase_t() `virtual Moon::Core::EntityBase_t::~~EntityBase_t () [inline], [virtual]`

Destroy the [EntityBase_t](#) object.

Definición en la línea 54 del archivo [ent_base.hpp](#).

5.7.3 Documentación de las funciones miembro

5.7.3.1 addComponent() `template<class CMP_t > requires std::is_base_of<Moon::Core::ComponentBase_t, CMP_t>::value void Moon::Core::EntityBase_t::addComponent (CMP_t * cmp)`

Añade un Component a this.

Parámetros del template

<i>CMP_t</i>	
<i>P_t</i>	

Parámetros

<i>cmp</i>	
------------	--

5.7.3.2 getComponent() `template<class CMP_t > requires std::is_base_of<Moon::Core::ComponentBase_t, CMP_t>::value CMP_t* Moon::Core::EntityBase_t::getComponent ()`

Get the Component object.

Parámetros del template

CM_{\leftrightarrow}	
P_t	

Devuelve

CMP_t^*

5.7.4 Documentación de los datos miembro

5.7.4.1 components `MapCmps_t Moon::Core::EntityBase_t::components`

Mapa de Tipos y Components

Definición en la línea 41 del archivo [ent_base.hpp](#).

5.7.4.2 nextType `Moon::Alias::EntityType Moon::Core::EntityBase_t::nextType = 0 [inline], [static]`

Identificador del siguiente tipo de Entity

Definición en la línea 38 del archivo [ent_base.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

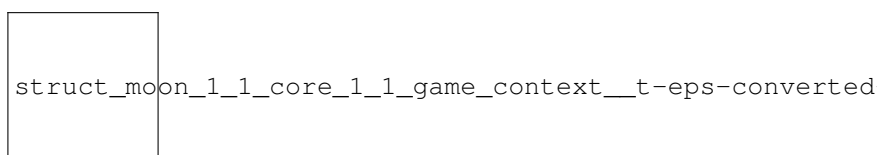
- `/home/rafa/git/Moon/include/ent/ent_base.hpp`

5.8 Referencia de la plantilla de la Estructura `Moon::Core::GameContext_t< Type >`

Clase GameContext !!! Es la encargada de almacenar Los datos de las Entity y los Components.

```
#include <gtx.hpp>
```

Diagrama de herencias de `Moon::Core::GameContext_t< Type >`



Métodos públicos

- [GameContext_t \(\)](#)
Constructor de [GameContext_t](#) object.
- virtual [~GameContext_t \(\)=0](#)
Destructor de [GameContext_t](#) object.
- Type & [addEntity \(\)](#)
Añade una nueva entity.
- Type * [getEntityById \(Moon::Alias::EntityId eid\)](#)
Get the Entity By Id object.
- template<Moon::Concepts::Cmp_t CMP_t, typename... Ts>
CMP_t & [addComponentById \(Moon::Alias::EntityId eid, Ts &&... args\)](#)
Añade un nuevo component a la entity señalada.
- template<Moon::Concepts::Cmp_t CMP_t>
std::vector< CMP_t > & [getComponents \(\)](#)
Get the Components object.

Métodos públicos estáticos

- static Moon::Alias::GameContextType [getGameContextType \(\)](#) noexcept
Get the Game Context Type object.

Atributos públicos

- std::vector< Type > [entities](#)
- [ComponentStorage_t components](#)

Otros miembros heredados

5.8.1 Descripción detallada

```
template<Moon::Concepts::Ent_t Type>
struct Moon::Core::GameContext_t< Type >
```

Clase GameContext !!! Es la encargada de almacenar Los datos de las Entity y los Components.

Parámetros del template

<i>Type</i>	
-------------	--

Definición en la línea [46](#) del archivo [gtx.hpp](#).

5.8.2 Documentación del constructor y destructor

5.8.2.1 GameContext_t() `template<Moon::Concepts::Ent_t Type>`
`Moon::Core::GameContext_t< Type >::GameContext_t ()`

Contructor de `GameContext_t` object.

5.8.2.2 ~GameContext_t() `template<Moon::Concepts::Ent_t Type>`
`virtual Moon::Core::GameContext_t< Type >::~~GameContext_t () [pure virtual]`

Destructor de `GameContext_t` object.

5.8.3 Documentación de las funciones miembro

5.8.3.1 addComponentById() `template<Moon::Concepts::Ent_t Type>`
`template<Moon::Concepts::Cmp_t CMP_t, typename... Ts>`
`CMP_t& Moon::Core::GameContext_t< Type >::addComponentById (`
`Moon::Alias::EntityId eid,`
`Ts &&... args) [inline]`

Añade un nuevo component a la entity señalada.

Parámetros del template

<i>Cmp_t</i>	
--------------	--

Parámetros

<i>eid</i>	
------------	--

Devuelve

`Cmp_t&`

Definición en la línea 96 del archivo `gtx.hpp`.

5.8.3.2 addEntity() `template<Moon::Concepts::Ent_t Type>`
`Type& Moon::Core::GameContext_t< Type >::addEntity ()`

Añade una nueva entity.

Devuelve

`Type&`

```
5.8.3.3 getComponents()  template<Moon::Concepts::Ent_t Type>
template<Moon::Concepts::Cmp_t CMP_t>
std::vector<CMP_t>& Moon::Core::GameContext_t< Type >::getComponents ( ) [inline]
```

Get the Components object.

Parámetros del template

$CM \leftrightarrow P_t$	
---------------------------	--

Devuelve

std::vector<CMP_t>&

Definición en la línea 112 del archivo [gtx.hpp](#).

```
5.8.3.4 getEntityById()  template<Moon::Concepts::Ent_t Type>
Type* Moon::Core::GameContext_t< Type >::getEntityById (
    Moon::Alias::EntityId eid )
```

Get the Entity By Id object.

Parámetros

<i>eid</i>	
------------	--

Devuelve

Type*

```
5.8.3.5 getGameContextType()  template<Moon::Concepts::Ent_t Type>
static Moon::Alias::GameContextType Moon::Core::GameContext_t< Type >::getGameContextType ( )
[static], [noexcept]
```

Get the Game Context Type object.

Devuelve

Moon::Alias::GameContextType

5.8.4 Documentación de los datos miembro

5.8.4.1 components `template<Moon::Concepts::Ent_t Type>`
`ComponentStorage_t Moon::Core::GameContext_t< Type >::components`

Almacenamiento de los Compnnets

Definición en la línea 52 del archivo [gtx.hpp](#).

5.8.4.2 entities `template<Moon::Concepts::Ent_t Type>`
`std::vector<Type> Moon::Core::GameContext_t< Type >::entities`

Vector de entities

Definición en la línea 49 del archivo [gtx.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

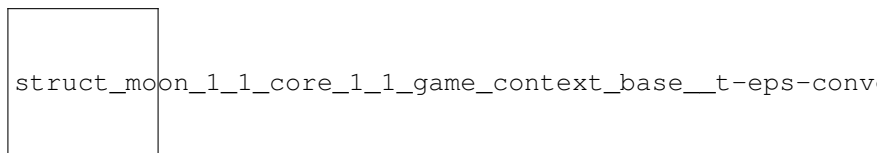
- `/home/rafa/git/Moon/include/gtx/gtx.hpp`

5.9 Referencia de la Estructura Moon::Core::GameContextBase_t

Clase GameContext !!! Es la encargada de almacenar Los datos de una App.

```
#include <gtx_base.hpp>
```

Diagrama de herencias de Moon::Core::GameContextBase_t



Métodos públicos

- [GameContextBase_t\(\)](#)
Constructor de [GameContextBase_t](#) object.
- virtual [~GameContextBase_t\(\)](#)=0
Destructor de [GameContextBase_t](#) object.

Atributos públicos estáticos

- static Moon::Alias::GameContextType [nextType](#) { 0 }
- static [Moon::Alias::EntityId](#) [nextId](#) { 0 }

5.9.1 Descripción detallada

Clase GameContext !!! Es la encargada de almacenar Los datos de una App.

Definición en la línea 27 del archivo [gtx_base.hpp](#).

5.9.2 Documentación del constructor y destructor

5.9.2.1 GameContextBase_t() Moon::Core::GameContextBase_t::GameContextBase_t ()

Constructor de [GameContextBase_t](#) object.

5.9.2.2 ~GameContextBase_t() virtual Moon::Core::GameContextBase_t::~~GameContextBase_t () [pure virtual]

Destructor de [GameContextBase_t](#) object.

5.9.3 Documentación de los datos miembro

5.9.3.1 nextId Moon::Alias::EntityId Moon::Core::GameContextBase_t::nextId { 0 } [inline], [static]

Identificador del siguiente ID de Entity

Definición en la línea 33 del archivo [gtx_base.hpp](#).

5.9.3.2 nextType Moon::Alias::GameContextType Moon::Core::GameContextBase_t::nextType { 0 } [inline], [static]

Identificador del siguiente tipo de GameContext

Definición en la línea 30 del archivo [gtx_base.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

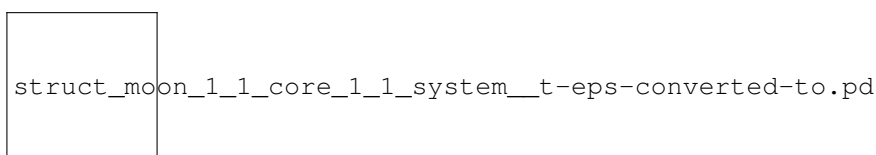
- /home/rafa/git/Moon/include/gtx/gtx_base.hpp

5.10 Referencia de la plantilla de la Estructura Moon::Core::System_t< Type >

Clase de la que heredan todos los Systems del Motor.

```
#include <sys.hpp>
```

Diagrama de herencias de Moon::Core::System_t< Type >



Métodos públicos

- [System_t](#) ()
Constructor de [System_t](#) object.
- virtual [~System_t](#) ()=0
Destructor de [System_t](#) object.
- virtual void [update](#) (Type *... gameContext)=0
Metodo que actualiza al System.
- virtual bool [alive](#) ()=0
Metodo que indica si el system sigue alive.

Métodos públicos estáticos

- static Moon::Alias::SystemType [getSystemType](#) () noexcept
Get the System Type object.

Otros miembros heredados

5.10.1 Descripción detallada

```
template<Moon::Concepts::Ctx_t... Type>
struct Moon::Core::System_t< Type >
```

Clase de la que heredan todos los Systems del Motor.

Parámetros del template

<i>Type</i>	
-------------	--

Definición en la línea 32 del archivo [sys.hpp](#).

5.10.2 Documentación del constructor y destructor

5.10.2.1 [System_t](#)() `template<Moon::Concepts::Ctx_t... Type>`
[Moon::Core::System_t](#)< Type >::[System_t](#) ()

Constructor de [System_t](#) object.

5.10.2.2 [~System_t](#)() `template<Moon::Concepts::Ctx_t... Type>`
 virtual [Moon::Core::System_t](#)< Type >::[~System_t](#) () [pure virtual]

Destructor de [System_t](#) object.

5.10.3 Documentación de las funciones miembro

5.10.3.1 alive() `template<Moon::Concepts::Ctx_t... Type>`
`virtual bool Moon::Core::System_t< Type >::alive () [pure virtual]`

Metodo que indica si el system sigue alive.

Devuelve

true
false

Implementa [Moon::Core::SystemBase_t](#).

5.10.3.2 getSystemType() `template<Moon::Concepts::Ctx_t... Type>`
`static Moon::Alias::SystemType Moon::Core::System_t< Type >::getSystemType () [static],`
`[noexcept]`

Get the System Type object.

Devuelve

Moon::Alias::SystemType

5.10.3.3 update() `template<Moon::Concepts::Ctx_t... Type>`
`virtual void Moon::Core::System_t< Type >::update (`
`Type *... gameContext) [pure virtual]`

Metodo que actualiza al System.

Parámetros

<i>gameContext</i>	
--------------------	--

La documentación para esta estructura fue generada a partir del siguiente fichero:

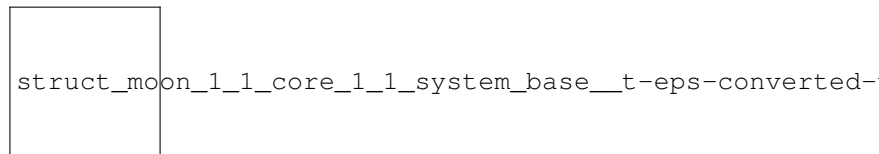
- /home/rafa/git/Moon/include/sys/sys.hpp

5.11 Referencia de la Estructura Moon::Core::SystemBase_t

Clase de la que heredan todos los sistemas del Motor.


```
#include <sys_base.hpp>
```

Diagrama de herencias de Moon::Core::SystemBase_t



Métodos públicos

- [SystemBase_t](#) ()
Contructor de [SystemBase_t](#) object.
- virtual [~SystemBase_t](#) ()=0
Destructor de [SystemBase_t](#) object.
- virtual bool [alive](#) ()=0
Metodo que indica si el sistema sigue alive.

Atributos públicos estáticos

- static Moon::Alias::SystemType [nextType](#) = 0

5.11.1 Descripción detallada

Clase de la que heredan todos los sistemas del Motor.

Definición en la línea [25](#) del archivo [sys_base.hpp](#).

5.11.2 Documentación del constructor y destructor

5.11.2.1 SystemBase_t() Moon::Core::SystemBase_t::SystemBase_t ()

Contructor de [SystemBase_t](#) object.

5.11.2.2 ~SystemBase_t() virtual Moon::Core::SystemBase_t::~~SystemBase_t () [pure virtual]

Destructor de [SystemBase_t](#) object.

5.11.3 Documentación de las funciones miembro

5.11.3.1 alive() `virtual bool Moon::Core::SystemBase_t::alive () [pure virtual]`

Metodo que indica si el sistema sigue alive.

Devuelve

true

false

Implementado en [Moon::Core::System_t< Type >](#).

5.11.4 Documentación de los datos miembro

5.11.4.1 nextType `Moon::Alias::SystemType Moon::Core::SystemBase_t::nextType = 0 [inline], [static]`

ID de la Entity a la que pertenece el Component

Definición en la línea 28 del archivo [sys_base.hpp](#).

La documentación para esta estructura fue generada a partir del siguiente fichero:

- `/home/rafa/git/Moon/include/sys/sys_base.hpp`

