R-Type architecture

Generated by Doxygen 1.13.2

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

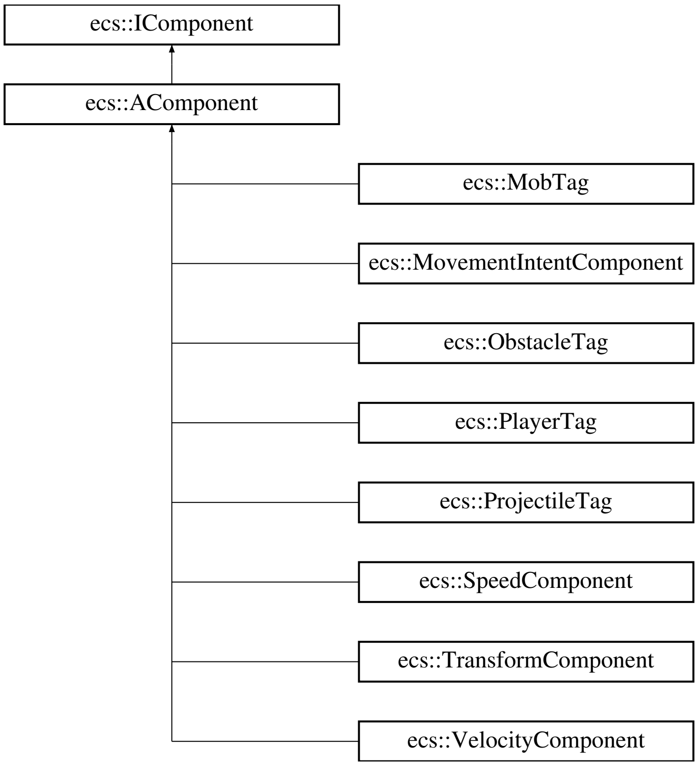
Here is a list of all documented files with brief descriptions:

Chapter 4

Class Documentation

4.1 ecs::AComponent Class Reference

Inheritance diagram for ecs::AComponent:



Public Member Functions

- ComponentState getState () const override
- void setState (ComponentState newState) override

Protected Attributes

- ComponentState **_state** = Permanent

4.1.1 Member Function Documentation

4.1.1.1 getState()

```
ComponentState ecs::AComponent::getState () const  [override], [virtual]
```

Implements ecs::IComponent.

4.1.1.2 setState()

```
void ecs::AComponent::setState (
             ComponentState newState)  [override], [virtual]
```

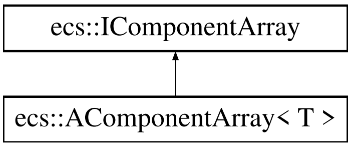
Implements ecs::IComponent.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/base/AComponent.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/base/AComponent.cpp

4.2 ecs::AComponentArray< T > Class Template Reference

Inheritance diagram for ecs::AComponentArray< T >:



Public Member Functions

- void **add** (int entityId, std::shared_ptr< T > component)
- std::shared_ptr< T > **get** (int entityId) const
- void **remove** (int entityId)
- bool **has** (int entityId) const
- void removeAllComponentsWithState (ComponentState state) override
- size_t getMaxEntityId () const override

Private Attributes

- std::vector< std::shared_ptr< T > > **_components**

4.2.1 Member Function Documentation

4.2.1.1 getMaxEntityId()

```
template<typename T>
size_t ecs::AComponentArray< T >::getMaxEntityId () const  [override], [virtual]
```

Implements ecs::IComponentArray.

4.2.1.2 removeAllComponentsWithState()

```
template<typename T>
void ecs::AComponentArray< T >::removeAllComponentsWithState (
             ComponentState state)  [override], [virtual]
```

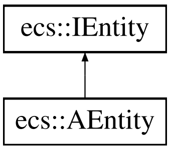
Implements ecs::IComponentArray.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/componentArray/AComponentArray.hpp

4.3 ecs::AEntity Class Reference

Inheritance diagram for ecs::AEntity:



Public Member Functions

- operator size_t () const override

Private Member Functions

- **AEntity** (size_t id)

Private Attributes

- size_t **_id**

4.3.1 Member Function Documentation

4.3.1.1 operator size_t()

```
ecs::AEntity::operator size_t () const  [override], [virtual]
```

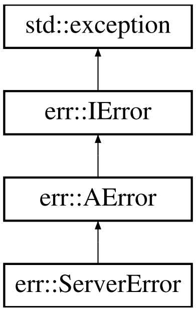
Implements ecs::IEntity.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/AEntity.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/AEntity.cpp

4.4 err::AError Class Reference

Inheritance diagram for err::AError:



Public Member Functions

- **AError** (const std::string &message, int code=0)
- const char ∗ what () const noexcept override
- int getCode () const noexcept
- std::string getDetails () const noexcept
- virtual std::string getType () const noexcept=0

Protected Attributes

- std::string **m_message**
- int **m_code**

4.4.1 Member Function Documentation

4.4.1.1 getCode()

```
int err::AError::getCode () const  [virtual], [noexcept]
```

Implements err::IError.

4.4.1.2 getDetails()

```
std::string err::AError::getDetails () const  [virtual], [noexcept]
```

Implements err::IError.

4.4.1.3 getType()

```
virtual std::string err::AError::getType () const  [pure virtual], [noexcept]
```

Implements err::IError.

4.4.1.4 what()

```
const char * err::AError::what () const  [override], [virtual], [noexcept]
```

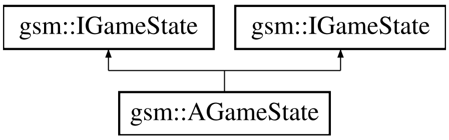
Implements err::IError.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/AError.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/AError.cpp

4.5 gsm::AGameState Class Reference

Inheritance diagram for gsm::AGameState:



Public Member Functions

- **AGameState** (std::shared_ptr< IGameStateMachine > gsm)
- void enter () override
- void update (float deltaTime) override
- void render () override
- void exit () override
- **AGameState** (std::shared_ptr< IGameStateMachine > gsm)
- void enter () override
- void update (float deltaTime) override
- void exit () override

Protected Attributes

- std::shared_ptr< IGameStateMachine > **_gsm**

4.5.1 Member Function Documentation

4.5.1.1 enter() [1/2]

```
void gsm::AGameState::enter ()  [override], [virtual]
```

Implements gsm::IGameState.

4.5.1.2 enter() [2/2]

```
void gsm::AGameState::enter ()  [override], [virtual]
```

Implements gsm::IGameState.

4.5.1.3 exit() [1/2]

```
void gsm::AGameState::exit ()  [override], [virtual]
```

Implements gsm::IGameState.

4.5.1.4 exit() [2/2]

```
void gsm::AGameState::exit ()  [override], [virtual]
```

Implements gsm::IGameState.

4.5.1.5 render()

```
void gsm::AGameState::render ()  [override], [virtual]
```

Implements gsm::IGameState.

4.5.1.6 update() [1/2]

```
void gsm::AGameState::update (
             float deltaTime)  [override], [virtual]
```

Implements gsm::IGameState.

4.5.1.7 update() [2/2]

```
void gsm::AGameState::update (
             float deltaTime)  [override], [virtual]
```

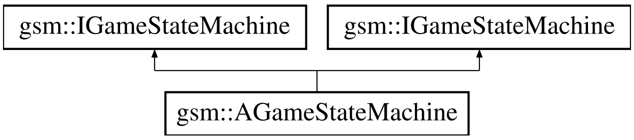
Implements gsm::IGameState.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/AGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/AGameState.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/AGameState.cpp

4.6 gsm::AGameStateMachine Class Reference

Inheritance diagram for gsm::AGameStateMachine:



Public Member Functions

- void changeState (std::shared_ptr< IGameState > newState) override
- void pushState (std::shared_ptr< IGameState > newState) override
- void popState () override
- void update (float deltaTime) override
- void render () override
- void changeState (std::shared_ptr< IGameState > newState) override
- void pushState (std::shared_ptr< IGameState > newState) override
- void popState () override
- void update (float deltaTime) override

Protected Attributes

- std::stack< std::shared_ptr< IGameState > > **_states**

4.6.1 Member Function Documentation

4.6.1.1 changeState() [1/2]

```
void gsm::AGameStateMachine::changeState (
            std::shared_ptr< IGameState > newState)  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.2 changeState() [2/2]

```
void gsm::AGameStateMachine::changeState (
            std::shared_ptr< IGameState > newState)  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.3 popState() [1/2]

```
void gsm::AGameStateMachine::popState ()  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.4 popState() [2/2]

```
void gsm::AGameStateMachine::popState ()  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.5 pushState() [1/2]

```
void gsm::AGameStateMachine::pushState (
            std::shared_ptr< IGameState > newState)  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.6 pushState() [2/2]

```
void gsm::AGameStateMachine::pushState (
            std::shared_ptr< IGameState > newState)  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.7 render()

```
void gsm::AGameStateMachine::render ()  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.8 update() [1/2]

```
void gsm::AGameStateMachine::update (
            float deltaTime)  [override], [virtual]
```

Implements gsm::IGameStateMachine.

4.6.1.9 update() [2/2]

```
void gsm::AGameStateMachine::update (
            float deltaTime)  [override], [virtual]
```

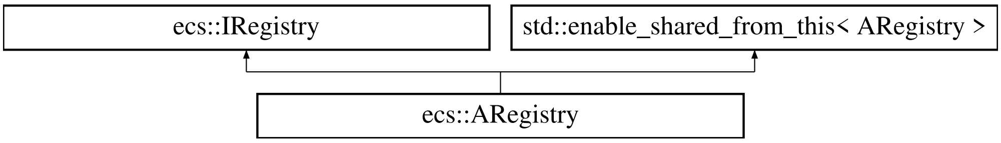
Implements gsm::IGameStateMachine.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/AGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/AGameStateMachine.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/AGameStateMachine.cpp

4.7 ecs::ARegistry Class Reference

Inheritance diagram for ecs::ARegistry:



Public Member Functions

- template< typename T>
  void **registerComponent** ()
- template< typename T>
  void **addComponent** (int entityId, std::shared_ptr< T > component)
- template< typename T>
  std::shared_ptr< T > **getComponent** (int entityId) const
- template< typename T>
  void **removeComponent** (int entityId)
- template< typename T>
  bool **hasComponent** (int entityId) const
- template< typename... Components>
  View< Components... > **view** ()
- template< typename... Components>
  Group< Components... > **group** ()
- size_t **getMaxEntityId** () const
- void removeAllComponentsWithState (ComponentState state) override

Public Member Functions inherited from ecs::IRegistry

- template< typename T>
  void **registerComponent** ()
- template< typename T>
  void **addComponent** (int entityId, std::shared_ptr< T > component)
- template< typename T>
  std::shared_ptr< T > **getComponent** (int entityId)
- template< typename T>
  void **removeComponent** (int entityId)
- template< typename T>
  bool **hasComponent** (int entityId)
- template< typename... Components>
  View< Components... > **view** ()
- template< typename... Components>
  Group< Components... > **group** ()
- size_t **getMaxEntityId** () const

Private Attributes

- std::unordered_map< const char *, std::shared_ptr< IComponentArray > > **_components**

4.7.1 Member Function Documentation

4.7.1.1 removeAllComponentsWithState()

```
void ecs::ARegistry::removeAllComponentsWithState (
             ComponentState state)  [override], [virtual]
```

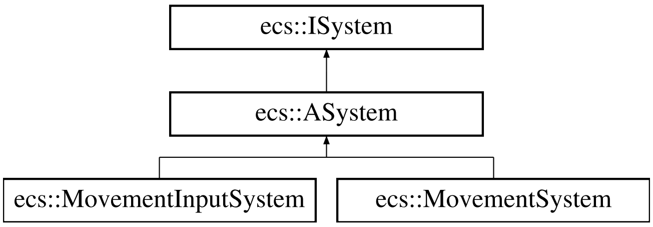
Implements ecs::IRegistry.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/registry/ARegistry.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/registry/ARegistry.cpp

4.8 ecs::ASystem Class Reference

Inheritance diagram for ecs::ASystem:



Public Member Functions

- void updateSystem (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime) override

Protected Member Functions

- virtual void **update** (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime)=0

4.8.1 Member Function Documentation

4.8.1.1 updateSystem()

```
void ecs::ASystem::updateSystem (
             std::shared_ptr< ResourceManager > resourceManager,
             std::shared_ptr< ARegistry > registry,
             float deltaTime)  [override], [virtual]
```

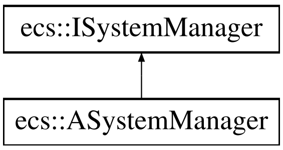
Implements ecs::ISystem.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/base/ASystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/base/ASystem.cpp

4.9 ecs::ASystemManager Class Reference

Inheritance diagram for ecs::ASystemManager:



Public Member Functions

- void updateAllSystems (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime) override
- void addSystem (std::shared_ptr< ISystem > system) override
- void removeSystem (std::shared_ptr< ISystem > system) override

Private Attributes

- std::vector< std::shared_ptr< ISystem > > **_systems**

4.9.1 Member Function Documentation

4.9.1.1 addSystem()

```
void ecs::ASystemManager::addSystem (
            std::shared_ptr< ISystem > system)  [override], [virtual]
```

Implements ecs::ISystemManager.

4.9.1.2 removeSystem()

```
void ecs::ASystemManager::removeSystem (
            std::shared_ptr< ISystem > system)  [override], [virtual]
```

Implements ecs::ISystemManager.

4.9.1.3 updateAllSystems()

```
void ecs::ASystemManager::updateAllSystems (
            std::shared_ptr< ResourceManager > resourceManager,
            std::shared_ptr< ARegistry > registry,
            float deltaTime)  [override], [virtual]
```

Implements ecs::ISystemManager.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/systemManager/ASystemManager.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/systemManager/ASystemManager.cpp

4.10 gfx::color_t Struct Reference

Public Attributes

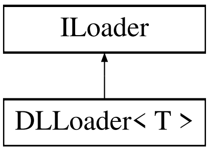
- int **r**
- int **g**
- int **b**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/IWindow.hpp

4.11 DLLoader< T > Class Template Reference

Inheritance diagram for DLLoader< T >:



Public Member Functions

- void ∗ getHandler () const override
- void ∗ Open (const char ∗path, int flag=RTLD_LAZY) override
- void ∗ Symbol (const char ∗symbolName) override
- T **getSymbol** (const char ∗symbolName)
- int Close () override
- const char ∗ Error () override

Private Attributes

- void ∗ **_handler** = nullptr

4.11.1 Member Function Documentation

4.11.1.1 Close()

```
template<typename T>
int DLLoader< T >::Close () [inline], [override], [virtual]
```

Implements ILoader.

4.11.1.2 Error()

```
template<typename T>
const char * DLLoader< T >::Error ()  [inline], [override], [virtual]
```

Implements ILoader.

4.11.1.3 getHandler()

```
template<typename T>
void * DLLoader< T >::getHandler () const  [inline], [override], [virtual]
```

Implements ILoader.

4.11.1.4 Open()

```
template<typename T>
void * DLLoader< T >::Open (
            const char * path,
            int flag = RTLD_LAZY)  [inline], [override], [virtual]
```

Implements ILoader.

4.11.1.5 Symbol()

```
template<typename T>
void * DLLoader< T >::Symbol (
            const char * symbolName)  [inline], [override], [virtual]
```

Implements ILoader.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/DLLoader/DLLoader.hpp

4.12 ecs::Group< Components > Class Template Reference

Classes

- class Iterator

Public Member Functions

- **Group** (std::shared_ptr< class ARegistry > registry)
- Iterator **begin** ()
- Iterator **end** ()

Private Attributes

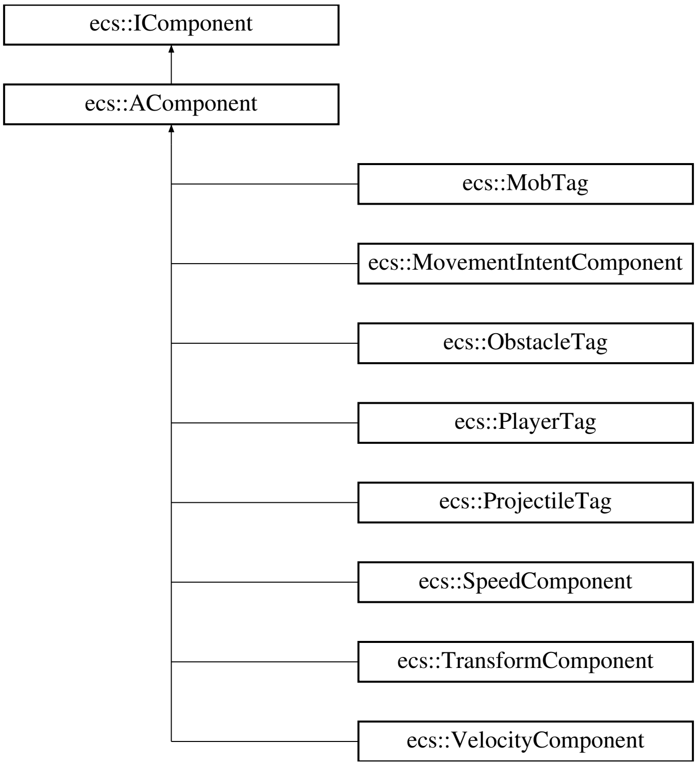
- std::shared_ptr< class ARegistry > **_registry**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.cpp

4.13 ecs::IComponent Class Reference

Inheritance diagram for ecs::IComponent:



Public Member Functions

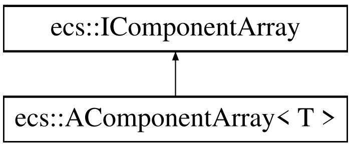
- virtual ComponentState **getState** () const =0
- virtual void **setState** (ComponentState newState)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/base/IComponent.hpp

4.14 ecs::IComponentArray Class Reference

Inheritance diagram for ecs::IComponentArray:



Public Member Functions

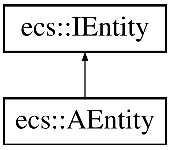
- virtual void **removeAllComponentsWithState** (ComponentState state)=0
- virtual size_t **getMaxEntityId** () const =0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/componentArray/IComponentArray.hpp

4.15 ecs::IEntity Class Reference

Inheritance diagram for ecs::IEntity:



Public Member Functions

- virtual **operator size_t** () const =0

Private Member Functions

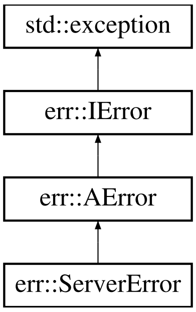
- **IEntity** (size_t id)

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/IEntity.hpp

4.16 err::IError Class Reference

Inheritance diagram for err::IError:



Public Member Functions

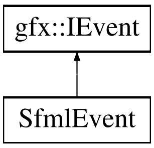
- virtual const char ∗ **what** () const noexcept override=0
- virtual int **getCode** () const noexcept=0
- virtual std::string **getType** () const noexcept=0
- virtual std::string **getDetails** () const noexcept=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/IError.hpp

4.17 gfx::IEvent Class Reference

Inheritance diagram for gfx::IEvent:



Public Types

- using **event_t** = EventType

Public Member Functions

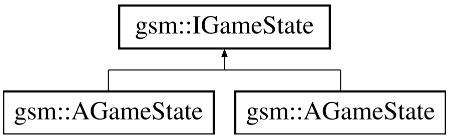
- virtual void **init** ()=0
- virtual event_t **pollEvents** ()=0
- virtual void **cleanup** ()=0
- virtual std::pair< int, int > **getMousePos** ()=0
- virtual bool **isKeyPressed** (event_t key)=0
- virtual bool **isMouseButtonPressed** (int button)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/IEvent.hpp

4.18 gsm::IGameState Class Reference

Inheritance diagram for gsm::IGameState:



Public Member Functions

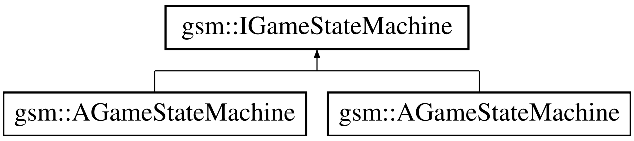
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **exit** ()=0
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **exit** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/IGameState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/states/IGameState.hpp

4.19 gsm::IGameStateMachine Class Reference

Inheritance diagram for gsm::IGameStateMachine:



Public Member Functions

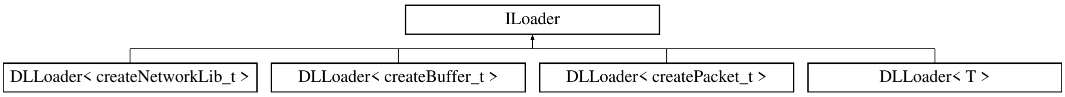
- virtual void **changeState** (std::shared_ptr< IGameState > newState)=0
- virtual void **pushState** (std::shared_ptr< IGameState > newState)=0
- virtual void **popState** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **changeState** (std::shared_ptr< IGameState > newState)=0
- virtual void **pushState** (std::shared_ptr< IGameState > newState)=0
- virtual void **popState** ()=0
- virtual void **update** (float deltaTime)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/machine/IGameStateMachine.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/gsm/machine/IGameStateMachine.hpp

4.20 ILoader Class Reference

Inheritance diagram for ILoader:


ILoader
DLLoader< createNetworkLib_t >
DLLoader< createBuffer_t >
DLLoader< createPacket_t >
DLLoader< T >

Public Member Functions

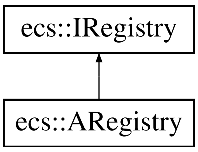
- virtual void ∗ **Open** (const char ∗path, int flag)=0
- virtual void ∗ **Symbol** (const char ∗symbolName)=0
- virtual int **Close** ()=0
- virtual const char ∗ **Error** ()=0
- virtual void ∗ **getHandler** () const =0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/DLLoader/ILoader.hpp

4.21 ecs::IRegistry Class Reference

Inheritance diagram for ecs::IRegistry:



Public Member Functions

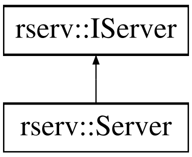
- template<typename T>
  void **registerComponent** ()
- template<typename T>
  void **addComponent** (int entityId, std::shared_ptr< T > component)
- template<typename T>
  std::shared_ptr< T > **getComponent** (int entityId)
- template<typename T>
  void **removeComponent** (int entityId)
- template<typename T>
  bool **hasComponent** (int entityId)
- template<typename... Components>
  View< Components... > **view** ()
- template<typename... Components>
  Group< Components... > **group** ()
- size_t **getMaxEntityId** () const
- virtual void **removeAllComponentsWithState** (ComponentState state)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/entity/registry/IRegistry.hpp

4.22 rserv::IServer Class Reference

Inheritance diagram for rserv::IServer:



Public Member Functions

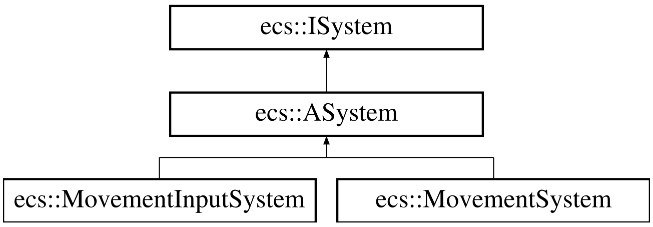
- virtual void **init** ()=0
- virtual void **start** ()=0
- virtual void **stop** ()=0
- virtual void **setConfig** (std::shared_ptr< ServerConfig > config)=0
- virtual std::shared_ptr< ServerConfig > **getConfig** () const =0
- virtual unsigned int **getPort** () const =0
- virtual void **setPort** (unsigned int port)=0
- virtual int **getState** () const =0
- virtual void **setState** (int state)=0
- virtual int **getFd** () const =0
- virtual void **setFd** (int fd)=0
- virtual **operator int** () const noexcept=0
- virtual std::shared_ptr< net::INetwork > **getNetwork** () const =0
- virtual void **setNetwork** (std::shared_ptr< net::INetwork > network)=0
- virtual void **onClientConnected** (int idClient)=0
- virtual void **onClientDisconnected** (int idClient)=0
- virtual void **onPacketReceived** (int idClient, const IPacket &packet)=0
- virtual void **processConnections** ()=0
- virtual void **processIncomingPackets** ()=0
- virtual void **broadcastPacket** (const IPacket &packet)=0
- virtual void **sendToClient** (int idClient, const IPacket &packet)=0
- virtual std::vector< int > **getConnectedClients** () const =0
- virtual int **getClientCount** () const =0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/IServer.hpp

4.23 ecs::ISystem Class Reference

Inheritance diagram for ecs::ISystem:



Public Member Functions

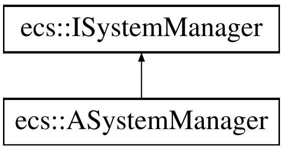
- virtual void **updateSystem** (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/base/ISystem.hpp

4.24 ecs::ISystemManager Class Reference

Inheritance diagram for ecs::ISystemManager:



Public Member Functions

- virtual void **updateAllSystems** (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime)=0
- virtual void **addSystem** (std::shared_ptr< ISystem > system)=0
- virtual void **removeSystem** (std::shared_ptr< ISystem > system)=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/systemManager/ISystemManager.hpp

4.25 ecs::Group< Components >::Iterator Class Reference

Public Member Functions

- **Iterator** (std::shared_ptr< class ARegistry > registry, size_t entityId, size_t maxEntityId)
- bool **operator!=** (const Iterator &other) const
- Iterator & **operator++** ()
- size_t **operator∗** () const

Private Member Functions

- bool **hasAllComponents** () const

Private Attributes

- std::shared_ptr< class ARegistry > **_registry**
- size_t **_entityId**
- size_t **_maxEntityId**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.cpp

4.26 ecs::View< Components >::Iterator Class Reference

Public Member Functions

- **Iterator** (std::shared_ptr< class ARegistry > registry, size_t entityId, size_t maxEntityId)
- bool **operator!=** (const Iterator &other) const
- Iterator & **operator++** ()
- size_t **operator*** () const

Private Member Functions

- bool **hasAllComponents** () const

Private Attributes

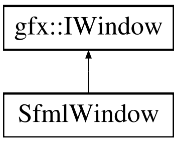
- std::shared_ptr< class ARegistry > **_registry**
- size_t **_entityId**
- size_t **_maxEntityId**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.cpp

4.27 gfx::IWindow Class Reference

Inheritance diagram for gfx::IWindow:



Public Member Functions

- virtual void **init** ()=0
- virtual void **display** ()=0
- virtual void **closeWindow** ()=0
- virtual bool **isOpen** ()=0
- virtual void **clear** ()=0
- virtual void **resizeWindow** (size_t x, size_t y)=0
- virtual void **drawSprite** (std::string asset, color_t color, std::pair< size_t, size_t > position)=0
- virtual void **drawText** (std::string text, color_t color, std::pair< size_t, size_t > position)=0
- virtual void **drawRectangle** (color_t color, std::pair< size_t, size_t > position, std::pair< size_t, size_t > size)=0
- virtual void **setFont** (const std::string &fontPath)=0
- virtual std::string **getFont** () const =0
- virtual bool **isMouseOver** (std::pair< size_t, size_t > position, std::pair< size_t, size_t > size)=0
- virtual std::pair< int, int > **getWindowSize** ()=0

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/IWindow.hpp

4.28 gfx::KeyMappings Class Reference

Static Public Member Functions

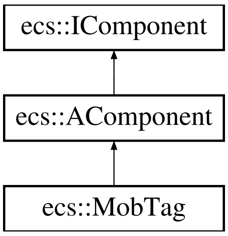
- static std::map< IEvent::event_t, sf::Keyboard::Key > **createKeyboardMapping** ()
- static std::map< IEvent::event_t, sf::Mouse::Button > **createMouseMapping** ()
- static std::map< IEvent::event_t, unsigned int > **createJoystickButtonMapping** ()

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/SfmlKeyMappings.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/SfmlKeyMappings.cpp

4.29 ecs::MobTag Class Reference

Inheritance diagram for ecs::MobTag:



Additional Inherited Members

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Protected Attributes inherited from ecs::AComponent

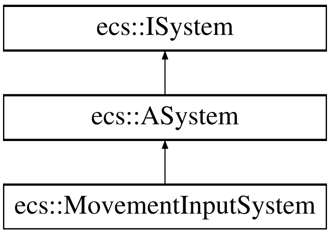
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/tags/MobTag.hpp

4.30 ecs::MovementInputSystem Class Reference

Inheritance diagram for ecs::MovementInputSystem:



Public Member Functions

- void update (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime) override
- void **simulateKeyPress** (MovementKey key, bool pressed)
- void **simulateAxis** (float horizontal, float vertical)

Public Member Functions inherited from ecs::ASystem

- void updateSystem (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime) override

Private Member Functions

- math::Vector2f **getMovementDirection** () const
- void **updateMovementIntent** (std::shared_ptr< ARegistry > registry, int entityId, const math::Vector2f &direction)

Private Attributes

- std::unordered_map< MovementKey, bool > **_movementKeyStates**
- math::Vector2f **_axisInput**

4.30.1 Member Function Documentation

4.30.1.1 update()

```
void ecs::MovementInputSystem::update (
            std::shared_ptr< ResourceManager > resourceManager,
            std::shared_ptr< ARegistry > registry,
            float deltaTime)  [override], [virtual]
```

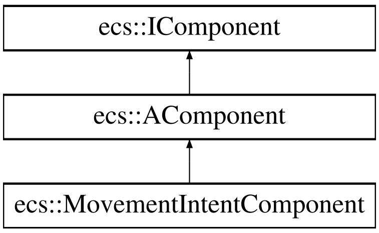
Implements ecs::ASystem.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/input/MovementInputSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/input/MovementInputSystem.cpp

4.31 ecs::MovementIntentComponent Class Reference

Inheritance diagram for ecs::MovementIntentComponent:



Public Member Functions

- **MovementIntentComponent** (const math::Vector2f &direction=math::Vector2f(0.0f, 0.0f), bool active=false)
- math::Vector2f **getDirection** () const
- void **setDirection** (const math::Vector2f &direction)
- bool **isActive** () const
- void **setActive** (bool active)

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Private Attributes

- math::Vector2f **_direction**
- bool **_active**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

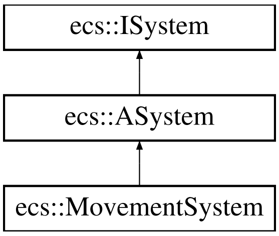
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/temporary/MovementIntent↵ Component.hpp

4.32 ecs::MovementSystem Class Reference

Inheritance diagram for ecs::MovementSystem:



Public Member Functions

- void update (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime) override

Public Member Functions inherited from ecs::ASystem

- void updateSystem (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime) override

4.32.1 Member Function Documentation

4.32.1.1 update()

```
void ecs::MovementSystem::update (
            std::shared_ptr< ResourceManager > resourceManager,
            std::shared_ptr< ARegistry > registry,
            float deltaTime) [override], [virtual]
```

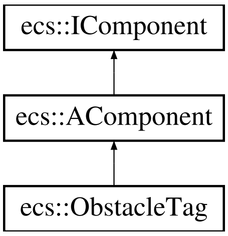
Implements ecs::ASystem.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/movement/MovementSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/movement/MovementSystem.cpp

4.33 ecs::ObstacleTag Class Reference

Inheritance diagram for ecs::ObstacleTag:



Additional Inherited Members

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Protected Attributes inherited from ecs::AComponent

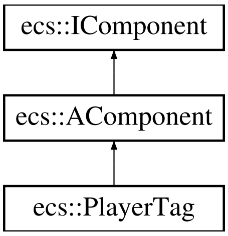
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/tags/ObstacleTag.hpp

4.34 ecs::PlayerTag Class Reference

Inheritance diagram for ecs::PlayerTag:



Additional Inherited Members

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Protected Attributes inherited from ecs::AComponent

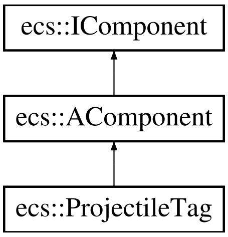
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/tags/PlayerTag.hpp

4.35 ecs::ProjectileTag Class Reference

Inheritance diagram for ecs::ProjectileTag:



Additional Inherited Members

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Protected Attributes inherited from ecs::AComponent

- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/tags/ProjectileTag.hpp

4.36 ecs::ResourceManager Class Reference

Public Member Functions

- template<typename T>
  void **add** (std::shared_ptr< T > resource)
- template<typename T>
  std::shared_ptr< T > **get** ()
- template<typename T>
  bool **has** ()

Private Attributes

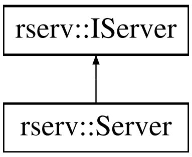
- std::unordered_map< size_t, std::shared_ptr< void > > **resources**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/resourceManager/ResourceManager.hpp

4.37 rserv::Server Class Reference

Inheritance diagram for rserv::Server:



Public Member Functions

- void init () override
- void start () override
- void stop () override
- void setConfig (std::shared_ptr< ServerConfig > config) override
- std::shared_ptr< ServerConfig > getConfig () const override
- unsigned int getPort () const override
- void setPort (unsigned int port) override
- int getState () const override
- void setState (int state) override
- int getFd () const override
- void setFd (int fd) override
- operator int () const noexcept override
- std::shared_ptr< net::INetwork > getNetwork () const override
- void setNetwork (std::shared_ptr< net::INetwork > network) override
- void onClientConnected (int idClient) override
- void onClientDisconnected (int idClient) override
- void onPacketReceived (int idClient, const IPacket &packet) override
- void processConnections () override
- void processIncomingPackets () override
- void broadcastPacket (const IPacket &packet) override
- void sendToClient (int idClient, const IPacket &packet) override
- std::vector< int > getConnectedClients () const override
- int getClientCount () const override

Private Member Functions

- void **loadNetworkLibrary** ()
- void **loadBufferLibrary** ()
- void **loadPacketLibrary** ()

Private Attributes

- DLLoader< createNetworkLib_t > **_networloader**
- DLLoader< createBuffer_t > **_bufferloader**
- DLLoader< createPacket_t > **_packetloader**
- std::shared_ptr< ServerConfig > **_config**
- std::shared_ptr< net::INetwork > **_network**
- std::shared_ptr< IBuffer > **_buffer**
- std::shared_ptr< IPacket > **_packet**

4.37.1 Member Function Documentation

4.37.1.1 broadcastPacket()

```
void rserv::Server::broadcastPacket (
             const IPacket & packet)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.2 getClientCount()

```
int rserv::Server::getClientCount () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.3 getConfig()

```
std::shared_ptr< rserv::ServerConfig > rserv::Server::getConfig () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.4 getConnectedClients()

```
std::vector< int > rserv::Server::getConnectedClients () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.5 getFd()

```
int rserv::Server::getFd () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.6 getNetwork()

```
std::shared_ptr< net::INetwork > rserv::Server::getNetwork () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.7 getPort()

```
unsigned int rserv::Server::getPort () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.8 getState()

```
int rserv::Server::getState () const  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.9 init()

```
void rserv::Server::init ()  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.10 onClientConnected()

```
void rserv::Server::onClientConnected (
            int idClient)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.11 onClientDisconnected()

```
void rserv::Server::onClientDisconnected (
            int idClient)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.12 onPacketReceived()

```
void rserv::Server::onPacketReceived (
            int idClient,
            const IPacket & packet)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.13 operator int()

```
rserv::Server::operator int () const  [override], [virtual], [noexcept]
```

Implements rserv::IServer.

4.37.1.14 processConnections()

```
void rserv::Server::processConnections ()  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.15 processIncomingPackets()

```
void rserv::Server::processIncomingPackets ()  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.16 sendToClient()

```
void rserv::Server::sendToClient (
             int idClient,
             const IPacket & packet)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.17 setConfig()

```
void rserv::Server::setConfig (
             std::shared_ptr< ServerConfig > config)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.18 setFd()

```
void rserv::Server::setFd (
             int fd)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.19 setNetwork()

```
void rserv::Server::setNetwork (
             std::shared_ptr< net::INetwork > network)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.20 setPort()

```
void rserv::Server::setPort (
             unsigned int port)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.21 setState()

```
void rserv::Server::setState (
            int state)  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.22 start()

```
void rserv::Server::start ()  [override], [virtual]
```

Implements rserv::IServer.

4.37.1.23 stop()

```
void rserv::Server::stop ()  [override], [virtual]
```

Implements rserv::IServer.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/Server.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Server.cpp

4.38 rserv::ServerConfig Class Reference

Public Member Functions

- int **getState** () const
- int **getFd** () const
- void **setPort** (unsigned int port)
- unsigned int **getPort** () const
- void **setState** (int state)
- void **setFd** (int fd)
- void **setNbClients** (int nbClients)
- int **getNbClients** () const
- uint32_t **getIp** () const
- void **setIp** (uint32_t ip)

Private Attributes

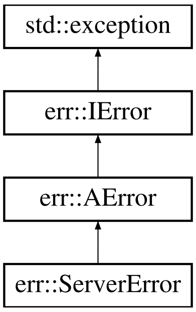
- int **_state**
- int **_fd**
- unsigned int **_port**
- int **_nbClients**
- uint32_t **_ip**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/ServerConfig.cpp

4.39 err::ServerError Class Reference

Inheritance diagram for err::ServerError:



Public Types

- enum **ErrorCode** {
  UNKNOWN = 1000 , **CONNECTION_FAILED** = 1001 , **TIMEOUT** = 1002 , **INVALID_REQUEST** = 1003 ,
  INTERNAL_ERROR = 1004 , **LIBRARY_LOAD_FAILED** = 1005 , **CONFIG_ERROR** = 1006 }

Public Member Functions

- **ServerError** (const std::string &message, ErrorCode code=UNKNOWN)
- std::string getType () const noexcept override

Public Member Functions inherited from err::AError

- **AError** (const std::string &message, int code=0)
- const char ∗ what () const noexcept override
- int getCode () const noexcept
- std::string getDetails () const noexcept

Additional Inherited Members

Protected Attributes inherited from err::AError

- std::string **m_message**
- int **m_code**

4.39.1 Member Function Documentation

4.39.1.1 getType()

```
std::string err::ServerError::getType () const  [override], [virtual], [noexcept]
```

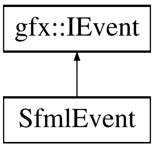
Implements err::AError.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ServerErrror.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Error/ServerErrror.cpp

4.40 SfmlEvent Class Reference

Inheritance diagram for SfmlEvent:



Public Member Functions

- **SfmlEvent** (std::shared_ptr< ecs::ResourceManager > resourceManager, std::shared_ptr< gfx::IWindow > window)
- void init () override
- event_t pollEvents () override
- void cleanup () override
- std::pair< int, int > getMousePos () override
- bool isKeyPressed (event_t key) override
- bool isMouseButtonPressed (int button) override

Private Member Functions

- void **initializeMappings** ()
- void **createReverseKeyboardMapping** ()
- event_t **processKeyboardEvent** (const sf::Event::KeyPressed &keyPressed)
- event_t **processMouseEvent** (const sf::Event::MouseButtonPressed &mousePressed)
- event_t **processJoystickButtonEvent** (const sf::Event::JoystickButtonPressed &joystickPressed)
- event_t **processJoystickAxisEvent** (const sf::Event::JoystickMoved &joystickMoved)

Private Attributes

- std::shared_ptr< ecs::ResourceManager > **_resourceManager**
- std::shared_ptr< gfx::IWindow > **_window**
- std::map< event_t, sf::Keyboard::Key > **_keyMap**
- std::map< sf::Keyboard::Key, event_t > **_reverseKeyMap**
- std::map< event_t, sf::Mouse::Button > **_mouseMap**
- std::map< event_t, unsigned int > **_joystickButtonMap**

Additional Inherited Members

Public Types inherited from gfx::IEvent

- using **event_t** = EventType

4.40.1 Member Function Documentation

4.40.1.1 cleanup()

```
void SfmlEvent::cleanup ()  [override], [virtual]
```

Implements gfx::IEvent.

4.40.1.2 getMousePos()

```
std::pair< int, int > SfmlEvent::getMousePos ()  [override], [virtual]
```

Implements gfx::IEvent.

4.40.1.3 init()

```
void SfmlEvent::init ()  [override], [virtual]
```

Implements gfx::IEvent.

4.40.1.4 isKeyPressed()

```
bool SfmlEvent::isKeyPressed (
            event_t key)  [override], [virtual]
```

Implements gfx::IEvent.

4.40.1.5 isMouseButtonPressed()

```
bool SfmlEvent::isMouseButtonPressed (
            int button)  [override], [virtual]
```

Implements gfx::IEvent.

4.40.1.6 pollEvents()

```
gfx::IEvent::event_t SfmlEvent::pollEvents ()  [override], [virtual]
```

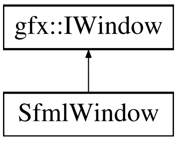
Implements gfx::IEvent.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/SfmlEvent.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/SfmlEvent.cpp

4.41 SfmlWindow Class Reference

Inheritance diagram for SfmlWindow:



Public Member Functions

- **SfmlWindow** (std::string title="R-Type", size_t width=800, size_t height=600)
- void init () override
- void display () override
- void closeWindow () override
- bool isOpen () override
- void clear () override
- void resizeWindow (size_t x, size_t y) override
- void drawSprite (std::string asset, gfx::color_t color, std::pair< size_t, size_t > position) override
- void drawText (std::string text, gfx::color_t color, std::pair< size_t, size_t > position) override
- void drawRectangle (gfx::color_t color, std::pair< size_t, size_t > position, std::pair< size_t, size_t > size) override
- void setFont (const std::string &fontPath) override
- std::string getFont () const override
- std::shared_ptr< sf::RenderWindow > **getSfmlWindow** ()
- bool isMouseOver (std::pair< size_t, size_t > position, std::pair< size_t, size_t > size) override
- std::pair< int, int > getWindowSize () override

Private Attributes

- std::shared_ptr< sf::RenderWindow > **_window**
- sf::Font **_font**
- std::string **_fontPath**

4.41.1 Member Function Documentation

4.41.1.1 clear()

```
void SfmlWindow::clear ()  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.2 closeWindow()

```
void SfmlWindow::closeWindow ()  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.3 display()

```
void SfmlWindow::display ()  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.4 drawRectangle()

```
void SfmlWindow::drawRectangle (
            gfx::color_t color,
            std::pair< size_t, size_t > position,
            std::pair< size_t, size_t > size)  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.5 drawSprite()

```
void SfmlWindow::drawSprite (
            std::string asset,
            gfx::color_t color,
            std::pair< size_t, size_t > position)  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.6 drawText()

```
void SfmlWindow::drawText (
            std::string text,
            gfx::color_t color,
            std::pair< size_t, size_t > position)  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.7 getFont()

```
std::string SfmlWindow::getFont () const  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.8 getWindowSize()

```
std::pair< int, int > SfmlWindow::getWindowSize ()  [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.9 init()

```
void SfmlWindow::init () [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.10 isMouseOver()

```
bool SfmlWindow::isMouseOver (
             std::pair< size_t, size_t > position,
             std::pair< size_t, size_t > size) [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.11 isOpen()

```
bool SfmlWindow::isOpen () [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.12 resizeWindow()

```
void SfmlWindow::resizeWindow (
             size_t x,
             size_t y) [override], [virtual]
```

Implements gfx::IWindow.

4.41.1.13 setFont()

```
void SfmlWindow::setFont (
             const std::string & fontPath) [override], [virtual]
```

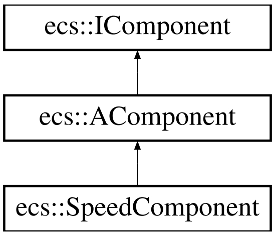
Implements gfx::IWindow.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/SfmlWindow.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/graphicals/SfmlWindow.cpp

4.42 ecs::SpeedComponent Class Reference

Inheritance diagram for ecs::SpeedComponent:



Public Member Functions

- **SpeedComponent** (float speed=constants::BASE_SPEED)
- float **getSpeed** () const
- void **setSpeed** (float speed)

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Private Attributes

- float **_speed**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

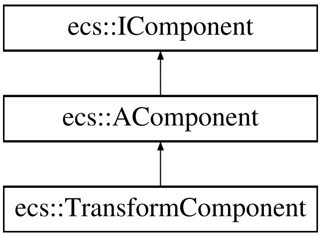
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/permanent/SpeedComponent.hpp

4.43 ecs::TransformComponent Class Reference

Inheritance diagram for ecs::TransformComponent:



Public Member Functions

- **TransformComponent** (math::Vector2f position=math::Vector2f(0.0f, 0.0f), float rotation=0.0f, math::Vector2f scale=math::Vector2f(1.0f, 1.0f))
- math::Vector2f **getPosition** () const
- void **setPosition** (math::Vector2f position)
- float **getRotation** () const
- void **setRotation** (float rotation)
- math::Vector2f **getScale** () const
- void **setScale** (math::Vector2f scale)

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Private Attributes

- math::Vector2f **_position**
- float **_rotation**
- math::Vector2f **_scale**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/permanent/TransformComponent.↩ hpp

4.44 Utils Class Reference

Public Member Functions

- void **helper** ()
- void **parsCli** (int ac, char **av, std::shared_ptr< rserv::ServerConfig > config)

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

- /home/albane/epitech/tech3/r-type/ryanR-type/server/Utils.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Utils.cpp

4.45 math::Vector2f Class Reference

Public Member Functions

- **Vector2f** (float x=0.0f, float y=0.0f)
- **Vector2f** (Vector2f const &other)
- float **getX** () const
- void **setX** (float x)
- float **getY** () const
- void **setY** (float y)
- Vector2f **getVector** () const
- Vector2f **operator*** (float scalar) const
- Vector2f **operator-** (Vector2f const &other) const
- Vector2f **operator+** (Vector2f const &other) const
- void **operator=** (Vector2f const &other)
- void **operator+=** (Vector2f const &other)
- void **operator-=** (Vector2f const &other)
- void **operator*=** (float scalar)
- void **operator/=** (float scalar)

Private Attributes

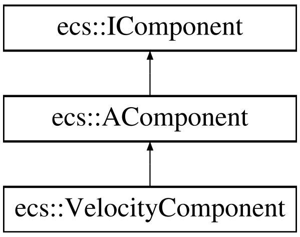
- float **_x**
- float **_y**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/types/Vector2f.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/types/Vector2f.cpp

4.46 ecs::VelocityComponent Class Reference

Inheritance diagram for ecs::VelocityComponent:



Public Member Functions

- **VelocityComponent** (math::Vector2f velocity=math::Vector2f(0.0f, 0.0f))
- math::Vector2f **getVelocity** () const
- void **setVelocity** (math::Vector2f velocity)

Public Member Functions inherited from ecs::AComponent

- ComponentState getState () const override
- void setState (ComponentState newState) override

Private Attributes

- math::Vector2f **_velocity**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/permanent/VelocityComponent.↩ hpp

4.47 ecs::View< Components > Class Template Reference

Classes

- class Iterator

Public Member Functions

- **View** (std::shared_ptr< class ARegistry > registry)
- Iterator **begin** ()
- Iterator **end** ()

Private Attributes

- std::shared_ptr< class ARegistry > **_registry**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/view/View.cpp

Chapter 5

File Documentation

5.1 EventTypes.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** EventTypes
*/

#ifndef EVENTTYPES_HPP_
#define EVENTTYPES_HPP_

namespace gfx {

enum class EventType {
    // Directional keys
    UP,
    DOWN,
    LEFT,
    RIGHT,

    // Alphanumeric keys
    A, B, C, D, E, F, G, H, I, J, K, L, M,
    N, O, P, Q, R, S, T, U, V, W, X, Y, Z,
    NUM0, NUM1, NUM2, NUM3, NUM4, NUM5, NUM6, NUM7, NUM8, NUM9,

    // Special keys
    SPACE,
    ENTER,
    ESCAPE,
    TAB,
    BACKSPACE,
    DELETE_KEY,
    INSERT,
    HOME,
    END,
    PAGEUP,
    PAGEDOWN,

    // Function keys
    F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12,

    // Modifiers
    LSHIFT,
    RSHIFT,
    LCTRL,
    RCTRL,
    LALT,
    RALT,

    // Symbols
    MINUS,
    PLUS,
    EQUALS,
    LBRACKET,
    RBRACKET,
    SEMICOLON,
    QUOTE,
    COMMA,
    PERIOD,
    SLASH,
    BACKSLASH,
    TILDE,

    // Numpad
    NUMPAD0, NUMPAD1, NUMPAD2, NUMPAD3, NUMPAD4,
    NUMPAD5, NUMPAD6, NUMPAD7, NUMPAD8, NUMPAD9,
    NUMPAD_ADD,
    NUMPAD_SUBTRACT,
    NUMPAD_MULTIPLY,
    NUMPAD_DIVIDE,
    NUMPAD_ENTER,

    // Mouse buttons
    MOUSECLICK,
    MOUSERIGHTCLICK,
    MOUSELEFTCLICK,
    MOUSEMIDDLECLICK,
    MOUSEX1,
    MOUSEX2,
    MOUSEWHEELUP,
    MOUSEWHEELDOWN,

    // Gamepad/Controller buttons
    GAMEPAD_A,
    GAMEPAD_B,
    GAMEPAD_X,
    GAMEPAD_Y,
    GAMEPAD_LEFT_BUMPER,
    GAMEPAD_RIGHT_BUMPER,
    GAMEPAD_BACK,
    GAMEPAD_START,
    GAMEPAD_GUIDE,
    GAMEPAD_LEFT_THUMB,
    GAMEPAD_RIGHT_THUMB,

    // Gamepad D-Pad
    GAMEPAD_DPAD_UP,
    GAMEPAD_DPAD_DOWN,
    GAMEPAD_DPAD_LEFT,
    GAMEPAD_DPAD_RIGHT,

    // Gamepad triggers
    GAMEPAD_LEFT_TRIGGER,
    GAMEPAD_RIGHT_TRIGGER,

    // Gamepad analog sticks
    GAMEPAD_LEFT_STICK_UP,
    GAMEPAD_LEFT_STICK_DOWN,
    GAMEPAD_LEFT_STICK_LEFT,
    GAMEPAD_LEFT_STICK_RIGHT,
    GAMEPAD_RIGHT_STICK_UP,
    GAMEPAD_RIGHT_STICK_DOWN,
    GAMEPAD_RIGHT_STICK_LEFT,
    GAMEPAD_RIGHT_STICK_RIGHT,

    // Window events
    CLOSE,
    REFRESH,

    // Default
    NOTHING
};

} // namespace gfx

#endif /* !EVENTTYPES_HPP_ */
```

5.2 IEvent.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** IEvent
*/

#ifndef IEVENT_HPP_
#define IEVENT_HPP_

#include <utility>
#include <memory>
```

```
#include "EventTypes.hpp"

namespace gfx {

class IEvent {
    public:
        using event_t = EventType;
        virtual ~IEvent() = default;
        virtual void init() = 0;
        virtual event_t pollEvents() = 0;
        virtual void cleanup() = 0;
        virtual std::pair<int, int> getMousePos() = 0;
        virtual bool isKeyPressed(event_t key) = 0;
        virtual bool isMouseButtonPressed(int button) = 0;

};

} // namespace gfx

#endif /* !IEVENT_HPP_ */
```

5.3 IWindow.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** IWindow
*/


#ifndef IWINDOW_HPP_
#define IWINDOW_HPP_

#include <string>
#include <utility>
#include <memory>

namespace gfx {

struct color_t {
    int r;
    int g;
    int b;
};

class IWindow {
    public:
        virtual ~IWindow() = default;
        virtual void init() = 0;
        virtual void display() = 0;
        virtual void closeWindow() = 0;
        virtual bool isOpen() = 0;
        virtual void clear() = 0;

        virtual void resizeWindow(size_t x, size_t y) = 0;

        virtual void drawSprite(std::string asset, color_t color, std::pair<size_t, size_t> position) = 0;
        virtual void drawText(std::string text, color_t color, std::pair<size_t, size_t> position) = 0;
        virtual void drawRectangle(color_t color, std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) = 0;

        virtual void setFont(const std::string& fontPath) = 0;
        virtual std::string getFont() const = 0;

        virtual bool isMouseOver(std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) = 0;
        virtual std::pair<int, int> getWindowSize() = 0;
};

} // namespace gfx

#endif /* !IWINDOW_HPP_ */
```

5.4 SfmlEvent.hpp

```
/*
```

```
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** SfmlEvent
*/

#ifndef SFMLEVENT_HPP_
#define SFMLEVENT_HPP_

#include <SFML/Graphics.hpp>
#include <map>
#include "./IEvent.hpp"
#include "./IWindow.hpp"
#include "./SfmlKeyMappings.hpp"
#include "../../common/ECS/resourceManager/ResourceManager.hpp"

class SfmlEvent : public gfx::IEvent {
    public:
        SfmlEvent(std::shared_ptr<ecs::ResourceManager> resourceManager, std::shared_ptr<gfx::IWindow> window);
        ~SfmlEvent() override;
        void init() override;
        event_t pollEvents() override;
        void cleanup() override;
        std::pair<int, int> getMousePos() override;
        bool isKeyPressed(event_t key) override;
        bool isMouseButtonPressed(int button) override;

    private:
        void initializeMappings();
        void createReverseKeyboardMapping();
        event_t processKeyboardEvent(const sf::Event::KeyPressed& keyPressed);
        event_t processMouseEvent(const sf::Event::MouseButtonPressed& mousePressed);
        event_t processJoystickButtonEvent(const sf::Event::JoystickButtonPressed& joystickPressed);
        event_t processJoystickAxisEvent(const sf::Event::JoystickMoved& joystickMoved);

        // Member variables
        std::shared_ptr<ecs::ResourceManager> _resourceManager;
        std::shared_ptr<gfx::IWindow> _window;
        std::map<event_t, sf::Keyboard::Key> _keyMap;
        std::map<sf::Keyboard::Key, event_t> _reverseKeyMap;
        std::map<event_t, sf::Mouse::Button> _mouseMap;
        std::map<event_t, unsigned int> _joystickButtonMap;
};

#endif /* !SFMLEVENT_HPP_ */
```

5.5 SfmlKeyMappings.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** KeyMappings
*/

#ifndef KEYMAPPINGS_HPP_
#define KEYMAPPINGS_HPP_

#include <SFML/Graphics.hpp>
#include <map>
#include "IEvent.hpp"
#include "EventTypes.hpp"

namespace gfx {

class KeyMappings {
    public:
        static std::map<IEvent::event_t, sf::Keyboard::Key> createKeyboardMapping();
        static std::map<IEvent::event_t, sf::Mouse::Button> createMouseMapping();
        static std::map<IEvent::event_t, unsigned int> createJoystickButtonMapping();
};

} // namespace gfx

#endif /* !KEYMAPPINGS_HPP_ */
```

5.6 SfmlWindow.hpp

```
/*
```

```
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** SfmlWindow
*/

#ifndef SFMLWINDOW_HPP_
#define SFMLWINDOW_HPP_

#include <SFML/Graphics.hpp>
#include "./IWindow.hpp"

class SfmlWindow : public gfx::IWindow {
    public:
        SfmlWindow(std::string title = "R-Type", size_t width = 800, size_t height = 600);
        ~SfmlWindow() override;
        void init() override;
        void display() override;
        void closeWindow() override;
        bool isOpen() override;
        void clear() override;

        void resizeWindow(size_t x, size_t y) override;

        void drawSprite(std::string asset, gfx::color_t color, std::pair<size_t, size_t> position) override;
        void drawText(std::string text, gfx::color_t color, std::pair<size_t, size_t> position) override;
        void drawRectangle(gfx::color_t color, std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) override;

        void setFont(const std::string& fontPath) override;
        std::string getFont() const override;

        std::shared_ptr<sf::RenderWindow> getSfmlWindow();
        bool isMouseOver(std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) override;
        std::pair<int, int> getWindowSize() override;
    private:
        std::shared_ptr<sf::RenderWindow> _window;
        sf::Font _font;
        std::string _fontPath;
};

#endif /* !SFMLWINDOW_HPP_ */
```

5.7 AGameStateMachine.hpp

```
#pragma once

#include "IGameStateMachine.hpp"
#include "../states/IGameState.hpp"

namespace gsm {

class AGameStateMachine : public IGameStateMachine {
public:
    AGameStateMachine();
    ~AGameStateMachine() override = default;

    void changeState(std::shared_ptr<IGameState> newState) override;
    void pushState(std::shared_ptr<IGameState> newState) override;
    void popState() override;

    void update(float deltaTime) override;
    void render() override;

protected:
    std::stack<std::shared_ptr<IGameState» _states;
};

} // namespace gsm
```

5.8 AGameStateMachine.hpp

```
#pragma once

#include "IGameStateMachine.hpp"
#include "../states/IGameState.hpp"
```

```

namespace gsm {

class AGameStateMachine : public IGameStateMachine {
public:
    AGameStateMachine();
    ~AGameStateMachine() override = default;

    void changeState(std::shared_ptr<IGameState> newState) override;
    void pushState(std::shared_ptr<IGameState> newState) override;
    void popState() override;

    void update(float deltaTime) override;

protected:
    std::stack<std::shared_ptr<IGameState» _states;
};

} // namespace gsm
```

5.9 IGameStateMachine.hpp

```
#pragma once

#include <memory>
#include <stack>

namespace gsm {

class IGameState;

class IGameStateMachine {
public:
    virtual ~IGameStateMachine() = default;

    virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
    virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
    virtual void popState() = 0;

    virtual void update(float deltaTime) = 0;
    virtual void render() = 0;
};

} // namespace gsm
```

5.10 IGameStateMachine.hpp

```
#pragma once

#include <memory>
#include <stack>

namespace gsm {

class IGameState;

class IGameStateMachine {
public:
    virtual ~IGameStateMachine() = default;

    virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
    virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
    virtual void popState() = 0;

    virtual void update(float deltaTime) = 0;
};

} // namespace gsm
```

5.11 AGameState.hpp

```
#pragma once

```

```
#include "IGameState.hpp"

namespace gsm {

class AGameState : public IGameState {
public:
    AGameState(std::shared_ptr<IGameStateMachine> gsm);
    ~AGameState() override = default;

    void enter() override;
    void update(float deltaTime) override;
    void render() override;
    void exit() override;

protected:
    std::shared_ptr<IGameStateMachine> _gsm;
};

} // namespace gsm
```

5.12 AGameState.hpp

```
#pragma once

#include "IGameState.hpp"

namespace gsm {

class AGameState : public IGameState {
public:
    AGameState(std::shared_ptr<IGameStateMachine> gsm);
    ~AGameState() override = default;

    void enter() override;
    void update(float deltaTime) override;
    void exit() override;

protected:
    std::shared_ptr<IGameStateMachine> _gsm;
};

} // namespace gsm
```

5.13 IGameState.hpp

```
#pragma once

#include <memory>
#include "../machine/IGameStateMachine.hpp"

namespace gsm {

class IGameState {
public:
    virtual ~IGameState() = default;

    virtual void enter() = 0;
    virtual void update(float deltaTime) = 0;
    virtual void render() = 0;
    virtual void exit() = 0;
};

} // namespace gsm
```

5.14 IGameState.hpp

```
#pragma once

#include <memory>
#include "../machine/IGameStateMachine.hpp"

namespace gsm {

```

```
class IGameState {
public:
    virtual ~IGameState() = default;

    virtual void enter() = 0;
    virtual void update(float deltaTime) = 0;
    virtual void exit() = 0;
};

} // namespace gsm
```

5.15 initRessourcesManager.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** initRessourcesManager
*/

#ifndef INITRESSOURCESMANAGER_HPP_
#define INITRESSOURCESMANAGER_HPP_

#include "../../common/ECS/resourceManager/ResourceManager.hpp"
#include <memory>

std::shared_ptr<ecs::ResourceManager> initRessourcesManager();

#endif /* !INITRESSOURCESMANAGER_HPP_ */
```

5.16 initRessourcesManager.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** initRessourcesManager
*/

#ifndef INITRESSOURCESMANAGER_HPP_
#define INITRESSOURCESMANAGER_HPP_

#include "../../common/ECS/resourceManager/ResourceManager.hpp"
#include <memory>

std::shared_ptr<ecs::ResourceManager> initRessourcesManager();

#endif /* !INITRESSOURCESMANAGER_HPP_ */
```

5.17 constants.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** Constants
*/

#ifndef CONSTANTS_HPP_
#define CONSTANTS_HPP_

namespace constants {
    constexpr float BASE_SPEED = 100.0f;
}

#endif /* !CONSTANTS_HPP_ */
```

5.18 DLLoader.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** DLLoader
*/

#ifndef DLLOADER_HPP_
#define DLLOADER_HPP_

#ifdef _WIN32
    #include <windows.h>
    #define RTLD_LAZY 0
#else
    #include <dlfcn.h>
#endif

#include <iostream>
#include <ostream>
#include <memory>
#include "ILoader.hpp"

template <typename T>

class DLLoader : public ILoader {
    private:
#ifdef _WIN32
        HMODULE _handler = nullptr;
        mutable std::string _lastError;
#else
        void *_handler = nullptr;
#endif

    public:
        ~DLLoader() = default;

        void *getHandler() const override {
            return _handler;
        };
        void *Open(const char *path, int flag = RTLD_LAZY) override {
#ifdef _WIN32
            _handler = LoadLibraryA(path);
            if (!_handler) {
                _lastError = "Failed to load library: " + std::string(path);
            }
#else
            _handler = dlopen(path, flag);
#endif
            return _handler;
        };
        void *Symbol(const char *symbolName) override {
#ifdef _WIN32
            void *symbol = (void*)GetProcAddress(_handler, symbolName);
            if (!symbol) {
                _lastError = "Failed to get symbol: " + std::string(symbolName);
                std::cerr « "GetProcAddress error: " « _lastError « std::endl;
                return nullptr;
            }
            return symbol;
#else
            void *symbol = dlsym(_handler, symbolName);
            const char *error = dlerror();
            if (error) {
                std::cerr « "dlerror: " « error « std::endl;
                return nullptr;
            }
            return symbol;
#endif
        };
        T getSymbol(const char *symbolName) {
#ifdef _WIN32
            return reinterpret_cast<T>(GetProcAddress(_handler, symbolName));
#else
            return reinterpret_cast<T>(dlsym(_handler, symbolName));
#endif
        };
        int Close() override{
            if (_handler == nullptr)
                return -1;
#ifdef _WIN32
            return FreeLibrary(_handler) ? 0 : -1;
#else
            return dlclose(_handler);
#endif
```

```
        };
        const char *Error() override {
#ifdef _WIN32
            return _lastError.c_str();
#else
            return dlerror();
#endif
        };
};

#endif /* !DLLOADER_HPP_ */
```

5.19 ILoader.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** ILoader
*/

#ifndef ILoader_HPP_
#define ILoader_HPP_


class ILoader {
    public:
        ~ILoader() = default;

        virtual void *Open(const char *path, int flag) = 0;
        virtual void *Symbol(const char *symbolName) = 0;
        virtual int Close() = 0;
        virtual const char *Error() = 0;
        virtual void *getHandler() const = 0;

    protected:
    private:
};

#endif /* !ILoader_HPP_ */
```

5.20 LoaderType.hpp

```
/*
** EPITECH PROJECT, 2025
** r-type
** File description:
** LoaderType
*/

#ifndef LOADERTYPE_HPP_
#define LOADERTYPE_HPP_

enum ModuleType_t{
    DISPLAY_MODULE = 0,
    NETWORK_MODULE = 1,
    PACKET_MODULE = 2,
    BUFFER_MODULE = 3,
    UNKNOWN_MODULE
};

typedef ModuleType_t (*getTypeFunc_t)();

#endif /* !LOADERTYPE_HPP_ */
```

5.21 AComponent.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** AComponent
*/
```

```

#ifndef ACOMPONENT_HPP_
#define ACOMPONENT_HPP_

#include "IComponent.hpp"

namespace ecs {

class AComponent : public IComponent {
    public:
        AComponent();
        ~AComponent();

        ComponentState getState() const override;
        void setState(ComponentState newState) override;

    protected:
        ComponentState _state = Permanent;

    private:
};

} // namespace ecs

#endif /* !ACOMPONENT_HPP_ */
```

5.22 IComponent.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** IComponent
*/

#ifndef ICOMPONENT_HPP_
#define ICOMPONENT_HPP_

namespace ecs {

enum ComponentState {
    Permanent = 0,
    Temporary = 1,
    Processed = 2,
};

class IComponent {
    public:
        IComponent() = default;
        virtual ~IComponent() = default;

        virtual ComponentState getState() const = 0;
        virtual void setState(ComponentState newState) = 0;

    protected:
    private:
};

} // namespace ecs

#endif /* !ICOMPONENT_HPP_ */
```

5.23 SpeedComponent.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** SpeedComponent
*/

#ifndef SPEEDCOMPONENT_HPP_
#define SPEEDCOMPONENT_HPP_

#include "../base/AComponent.hpp"
#include "constants.hpp"

namespace ecs {
```

```

class SpeedComponent : public AComponent {
    public:
        SpeedComponent(float speed = constants::BASE_SPEED) : _speed(speed) {};
        ~SpeedComponent() = default;

        float getSpeed() const { return _speed; };
        void setSpeed(float speed) { _speed = speed; };
    private:
        float _speed;
};

}  // namespace ecs

#endif /* !SPEEDCOMPONENT_HPP_ */
```

5.24 TransformComponent.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** TransformComponent
*/

#ifndef TRANSFORMCOMPONENT_HPP_
#define TRANSFORMCOMPONENT_HPP_

#include "../base/AComponent.hpp"
#include "../../../types/Vector2f.hpp"

namespace ecs {

class TransformComponent : public AComponent {
    public:
        TransformComponent(math::Vector2f position = math::Vector2f(0.0f, 0.0f), float rotation = 0.0f, math::Vector2f scale = math::Vector2f(1.0f, 1.0f))
            : _position(position), _rotation(rotation), _scale(scale) {};
        ~TransformComponent() = default;

        math::Vector2f getPosition() const { return _position; };
        void setPosition(math::Vector2f position) { _position = position; };

        float getRotation() const { return _rotation; };
        void setRotation(float rotation) { _rotation = rotation; };

        math::Vector2f getScale() const { return _scale; };
        void setScale(math::Vector2f scale) { _scale = scale; };

    private:
        math::Vector2f _position;
        float _rotation;
        math::Vector2f _scale;
};

} // namespace ecs

#endif /* !TRANSFORMCOMPONENT_HPP_ */
```

5.25 VelocityComponent.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** VelocityComponent
*/

#ifndef VELOCITYCOMPONENT_HPP_
#define VELOCITYCOMPONENT_HPP_

#include "../base/AComponent.hpp"
#include "../../../types/Vector2f.hpp"

namespace ecs {

class VelocityComponent : public AComponent {
    public:
        VelocityComponent(math::Vector2f velocity = math::Vector2f(0.0f, 0.0f)) : _velocity(velocity) {};
        ~VelocityComponent() = default;

        math::Vector2f getVelocity() const { return _velocity; };
        void setVelocity(math::Vector2f velocity) { _velocity = velocity; };
    private:
        math::Vector2f _velocity;
};

} // namespace ecs

#endif /* !VELOCITYCOMPONENT_HPP_ */
```

5.26 MobTag.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** MobTag
*/

#ifndef MOBTAG_HPP_
#define MOBTAG_HPP_

#include "../base/AComponent.hpp"

namespace ecs {

class MobTag : public AComponent {
    public:
        MobTag() = default;
        ~MobTag() = default;
};

} // namespace ecs

#endif /* !MOBTAG_HPP_ */
```

5.27 ObstacleTag.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ObstacleTag
*/

#ifndef OBSTACLETAG_HPP_
#define OBSTACLETAG_HPP_

#include "../base/AComponent.hpp"

namespace ecs {

class ObstacleTag : public AComponent {
    public:
        ObstacleTag() = default;
        ~ObstacleTag() = default;
};

} // namespace ecs

#endif /* !OBSTACLETAG_HPP_ */
```

5.28 PlayerTag.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** PlayerTag
```

```
*/

#ifndef PLAYERTAG_HPP_
#define PLAYERTAG_HPP_

#include "../base/AComponent.hpp"

namespace ecs {

class PlayerTag : public AComponent {
    public:
        PlayerTag() = default;
        ~PlayerTag() = default;
};

} // namespace ecs

#endif /* !PLAYERTAG_HPP_ */
```

5.29 ProjectileTag.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ProjectileTag
*/

#ifndef PROJECTILETAG_HPP_
#define PROJECTILETAG_HPP_

#include "../base/AComponent.hpp"

namespace ecs {

class ProjectileTag : public AComponent {
    public:
        ProjectileTag() = default;
        ~ProjectileTag() = default;
};

} // namespace ecs

#endif /* !PROJECTILETAG_HPP_ */
```

5.30 MovementIntentComponent.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** MovementIntentComponent
*/

#ifndef MOVEMENTINTENTCOMPONENT_HPP_
#define MOVEMENTINTENTCOMPONENT_HPP_

#include "../base/AComponent.hpp"
#include "../../../types/Vector2f.hpp"

namespace ecs {

class MovementIntentComponent : public AComponent {
    public:
        MovementIntentComponent(const math::Vector2f &direction = math::Vector2f(0.0f, 0.0f), bool active = false)
            : _direction(direction), _active(active) {
            _state = Temporary;
        };
        ~MovementIntentComponent() = default;

        math::Vector2f getDirection() const { return _direction; };
        void setDirection(const math::Vector2f &direction) { _direction = direction; };

        bool isActive() const { return _active; };
        void setActive(bool active) { _active = active; };

    private:
        math::Vector2f _direction;
        bool _active;
};

}  // namespace ecs

#endif /* !MOVEMENTINTENTCOMPONENT_HPP_ */
```

5.31 AEntity.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** IEntity
*/

#ifndef AENTITY_HPP_
#define AENTITY_HPP_

#include "IEntity.hpp"

namespace ecs {

class AEntity : public IEntity {
    public:
        AEntity();
        ~AEntity();
        operator size_t() const override;

    private:
        explicit AEntity(size_t id);
        size_t _id;
};

}  // namespace ecs

#endif /* !AENTITY_HPP_ */
```

5.32 AComponentArray.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** AComposantType
*/

#ifndef AComposantType_HPP_
#define AComposantType_HPP_

#include "IComponentArray.hpp"
#include "../../component/base/IComponent.hpp"
#include <vector>
#include <memory>

namespace ecs {

template <typename T>
class AComponentArray : public IComponentArray {
    public:
        AComponentArray();
        ~AComponentArray() override;

        void add(int entityId, std::shared_ptr<T> component);
        std::shared_ptr<T> get(int entityId) const;
        void remove(int entityId);
        bool has(int entityId) const;

        void removeAllComponentsWithState(ComponentState state) override;
        size_t getMaxEntityId() const override;

    private:
        std::vector<std::shared_ptr<T>> _components;
};

} // namespace ecs

```

```
#include "AComponentArray.tpp"

#endif /* !AComposantType_HPP_ */
```

5.33 IComponentArray.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** IComponentArray
*/

#ifndef ICOMPONENTARRAY_HPP_
#define ICOMPONENTARRAY_HPP_

#include "../../component/base/IComponent.hpp"

namespace ecs {

class IComponentArray {
    public:
        virtual ~IComponentArray() = default;
        virtual void removeAllComponentsWithState(ComponentState state) = 0;
        virtual size_t getMaxEntityId() const = 0;
};

} // namespace ecs

#endif /* !ICOMPONENTARRAY_HPP_ */
```

5.34 IEntity.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** IEntity
*/

#ifndef IENTITY_HPP_
#define IENTITY_HPP_

#include <cstddef>

namespace ecs {

class IEntity {
    public:
        IEntity() = default;
        virtual ~IEntity() = default;
        virtual operator size_t() const = 0;

    private:
        explicit IEntity(size_t id);
};

} // namespace ecs

#endif /* !IENTITY_HPP_ */
```

5.35 ARegistry.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ARegistry
*/

#ifndef AREGISTRY_HPP_
#define AREGISTRY_HPP_

```

```
#include "IRegistry.hpp"
#include "../../component/base/IComponent.hpp"
#include "../componentArray/IComponentArray.hpp"
#include "../componentArray/AComponentArray.hpp"
#include "../../view/View.hpp"
#include <memory>
#include <unordered_map>

namespace ecs {

class ARegistry : public IRegistry, public std::enable_shared_from_this<ARegistry> {
    public:
        ARegistry();
        virtual ~ARegistry();

        template <typename T>
        void registerComponent();

        template <typename T>
        void addComponent(int entityId, std::shared_ptr<T> component);
        template <typename T>
        std::shared_ptr<T> getComponent(int entityId) const;
        template <typename T>
        void removeComponent(int entityId);
        template <typename T>
        bool hasComponent(int entityId) const;

        template <typename... Components>
        View<Components...> view();

        template <typename... Components>
        Group<Components...> group();

        size_t getMaxEntityId() const;

        void removeAllComponentsWithState(ComponentState state) override;

    protected:
    private:
        std::unordered_map<const char *, std::shared_ptr<IComponentArray» _components;
};

} // namespace ecs

#include "ARegistry.tpp"

#endif /* !AREGISTRY_HPP_ */
```

5.36 IRegistry.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** IRegistry
*/

#ifndef IREGISTRY_HPP_
#define IREGISTRY_HPP_

#include <memory>
#include "../../component/base/IComponent.hpp"
#include "../../view/View.hpp"

namespace ecs {

class IRegistry {
    public:
        virtual ~IRegistry() = default;

        template <typename T>
        void registerComponent();

        template <typename T>
        void addComponent(int entityId, std::shared_ptr<T> component);

        template <typename T>
        std::shared_ptr<T> getComponent(int entityId);

        template <typename T>
        void removeComponent(int entityId);

        template <typename T>
        bool hasComponent(int entityId);

        template <typename... Components>
        View<Components...> view();

        template <typename... Components>
        Group<Components...> group();

        size_t getMaxEntityId() const;

        virtual void removeAllComponentsWithState(ComponentState state) = 0;
};

} // namespace ecs

#endif /* !IREGISTRY_HPP_ */
```

5.37 ResourceManager.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ResourceManager
*/

#ifndef RESOURCEMANAGER_HPP_
#define RESOURCEMANAGER_HPP_

#include <unordered_map>
#include <memory>

namespace ecs {

class ResourceManager {
    public:
        template<typename T>
        void add(std::shared_ptr<T> resource);

        template<typename T>
        std::shared_ptr<T> get();

        template<typename T>
        bool has();

    private:
        std::unordered_map<size_t, std::shared_ptr<void» resources;
};

} // namespace ecs

#include "ResourceManager.tpp"

#endif /* !RESOURCEMANAGER_HPP_ */
```

5.38 ASystem.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ASystem
*/

#ifndef ASystem_HPP_
#define ASystem_HPP_

#include <memory>

#include "ISystem.hpp"
#include "../../resourceManager/ResourceManager.hpp"
#include "../../entity/registry/ARegistry.hpp"

namespace ecs {

class ASystem : public ISystem {
    public:
        ASystem();
        ~ASystem() = default;
        void updateSystem(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) override;

    protected:
        virtual void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) = 0;

    private:
};

} // namespace ecs

#endif /* !ASystem_HPP_ */
```

5.39 ISystem.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ISystem
*/

#ifndef ISystem_HPP_
#define ISystem_HPP_

#include "../../resourceManager/ResourceManager.hpp"
#include "../../entity/registry/ARegistry.hpp"
#include <memory>

namespace ecs {

class ISystem {
    public:
        virtual ~ISystem() = default;
        virtual void updateSystem(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) = 0;
};

} // namespace ecs

#endif /* !ISystem_HPP_ */
```

5.40 MovementInputSystem.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** MovementInputSystem
*/

#ifndef MOVEMENTINPUTSYSTEM_HPP_
#define MOVEMENTINPUTSYSTEM_HPP_

#include "../base/ASystem.hpp"
#include "../../component/temporary/MovementIntentComponent.hpp"
#include <unordered_map>

namespace ecs {

enum class MovementKey {
    Left,
    Right,
    Up,
    Down
};

class MovementInputSystem : public ASystem {
    public:
        MovementInputSystem();
        ~MovementInputSystem() = default;

        void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) override;

        // Simulation methods (will be replaced by real input system)
        void simulateKeyPress(MovementKey key, bool pressed);
        void simulateAxis(float horizontal, float vertical);

    private:
        // Movement key states
        std::unordered_map<MovementKey, bool> _movementKeyStates;
        math::Vector2f _axisInput;

        math::Vector2f getMovementDirection() const;
        void updateMovementIntent(std::shared_ptr<ARegistry> registry, int entityId, const math::Vector2f &direction);
};

} // namespace ecs

#endif /* !MOVEMENTINPUTSYSTEM_HPP_ */
```

5.41 MovementSystem.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** MovementSystem
*/

#ifndef MOVEMENTSYSTEM_HPP_
#define MOVEMENTSYSTEM_HPP_

#include "../base/ASystem.hpp"
#include "../../component/base/IComponent.hpp"
#include "../../component/temporary/MovementIntentComponent.hpp"
#include "../../component/permanent/TransformComponent.hpp"
#include "../../component/permanent/SpeedComponent.hpp"
#include "../../../constants.hpp"

namespace ecs {

class MovementSystem : public ASystem {
    public:
        MovementSystem();
        ~MovementSystem() = default;

        void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) override;
};

}  // namespace ecs

#endif /* !MOVEMENTSYSTEM_HPP_ */
```

5.42 ASystemManager.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ASystemManager
*/

#ifndef ASystemMANAGER_HPP_
#define ASystemMANAGER_HPP_

#include <vector>
#include <memory>

#include "ISystemManager.hpp"
#include "../../resourceManager/ResourceManager.hpp"
#include "../../entity/registry/ARegistry.hpp"
#include "../base/ISystem.hpp"

namespace ecs {

class ASystemManager : public ISystemManager {
    public:
        ASystemManager();
        ~ASystemManager();
        void updateAllSystems(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) override;
        void addSystem(std::shared_ptr<ISystem> system) override;
        void removeSystem(std::shared_ptr<ISystem> system) override;

    private:
        std::vector<std::shared_ptr<ISystem» _systems;
};

} // namespace ecs

#endif /* !ASystemMANAGER_HPP_ */
```

5.43 ISystemManager.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ISystemManager
*/

#ifndef ISystemMANAGER_HPP_
#define ISystemMANAGER_HPP_

#include <memory>

#include "../base/ISystem.hpp"
#include "../../resourceManager/ResourceManager.hpp"
#include "../../entity/registry/ARegistry.hpp"

namespace ecs {

class ISystemManager {
    public:
        virtual ~ISystemManager() = default;
        virtual void updateAllSystems(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry> registry, float deltaTime) = 0;
        virtual void addSystem(std::shared_ptr<ISystem> system) = 0;
        virtual void removeSystem(std::shared_ptr<ISystem> system) = 0;
};

}  // namespace ecs

#endif /* !ISystemMANAGER_HPP_ */
```

5.44 View.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** View
*/

#ifndef VIEW_HPP_
#define VIEW_HPP_

#include <vector>
#include <memory>
#include <type_traits>

namespace ecs {

template <typename... Components>
class View {
    public:
        View(std::shared_ptr<class ARegistry> registry);

        class Iterator;

        Iterator begin();
        Iterator end();

        class Iterator {
            public:
                Iterator(std::shared_ptr<class ARegistry> registry, size_t entityId, size_t maxEntityId);
                bool operator!=(const Iterator& other) const;
                Iterator& operator++();
                size_t operator*() const;

            private:
                bool hasAllComponents() const;
                std::shared_ptr<class ARegistry> _registry;
                size_t _entityId;
                size_t _maxEntityId;
        };

    private:
        std::shared_ptr<class ARegistry> _registry;
};

template <typename... Components>
class Group {
    public:
        Group(std::shared_ptr<class ARegistry> registry);

        class Iterator;

        Iterator begin();
        Iterator end();

        class Iterator {
            public:
                Iterator(std::shared_ptr<class ARegistry> registry, size_t entityId, size_t maxEntityId);
                bool operator!=(const Iterator& other) const;
                Iterator& operator++();
                size_t operator*() const;

            private:
                bool hasAllComponents() const;
                std::shared_ptr<class ARegistry> _registry;
                size_t _entityId;
                size_t _maxEntityId;
        };

    private:
        std::shared_ptr<class ARegistry> _registry;
};

}  // namespace ecs

#endif /* !VIEW_HPP_ */
```

5.45 AError.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** AError
*/

#ifndef AERROR_HPP_
#define AERROR_HPP_

#include <string>
#include "IError.hpp"

namespace err {

class AError : public IError {
    public:
        AError(const std::string &message, int code = 0);

        virtual ~AError() noexcept = default;
        const char *what() const noexcept override;
        int getCode() const noexcept;
        std::string getDetails() const noexcept;

        virtual std::string getType() const noexcept = 0;

    protected:
        std::string m_message;
        int m_code;
};

}

#endif /* !AERROR_HPP_ */
```

5.46 IError.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** IError
*/

#ifndef IERROR_HPP_
#define IERROR_HPP_

#include <string>
#include <exception>

namespace err {

class IError : public std::exception {
    public:

        virtual ~IError() noexcept = default;
        virtual const char *what() const noexcept override = 0;
        virtual int getCode() const noexcept = 0;
        virtual std::string getType() const noexcept = 0;
        virtual std::string getDetails() const noexcept = 0;

    protected:
};

}

#endif /* !IERROR_HPP_ */
```

5.47 ServerErrror.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** ServerErrror
*/


#include "AError.hpp"

namespace err {

class ServerError : public AError {
    public:
        enum ErrorCode {
            UNKNOWN = 1000,
            CONNECTION_FAILED = 1001,
            TIMEOUT = 1002,
            INVALID_REQUEST = 1003,
            INTERNAL_ERROR = 1004,
            LIBRARY_LOAD_FAILED = 1005,
            CONFIG_ERROR = 1006
        };

        ServerError(const std::string &message, ErrorCode code = UNKNOWN);
        virtual ~ServerError() noexcept = default;
        std::string getType() const noexcept override;

    private:
};

}
```

5.48 Vector2f.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** Vector2f
*/

```

```
#ifndef VECTOR2F_HPP_
#define VECTOR2F_HPP_

namespace math {

class Vector2f {
    public:
        Vector2f(float x = 0.0f, float y = 0.0f);
        Vector2f(Vector2f const &other);
        ~Vector2f() = default;

        float getX() const;
        void setX(float x);
        float getY() const;
        void setY(float y);

        Vector2f getVector() const;
        Vector2f operator*(float scalar) const;
        Vector2f operator-(Vector2f const &other) const;
        Vector2f operator+(Vector2f const &other) const;
        void operator=(Vector2f const &other);
        void operator+=(Vector2f const &other);
        void operator-=(Vector2f const &other);
        void operator*=(float scalar);
        void operator/=(float scalar);
    private:
        float _x;
        float _y;
};

} // namespace math

#endif /* !VECTOR2F_HPP_ */
```

5.49 IServer.hpp

```
/*
** EPITECH PROJECT, 2025
** R-Type
** File description:
** Header
*/

#ifndef ISERVER_HPP_
    #define ISERVER_HPP_

#include <memory>
#include <vector>

#include "ServerConfig.hpp"
#include "../libs/Packet/IPacket.hpp"
#include "../libs/Network/INetwork.hpp"

namespace rserv {
class IServer {
public:
    virtual ~IServer() = default;

    virtual void init() = 0;
    virtual void start() = 0;
    virtual void stop() = 0;

    virtual void setConfig(std::shared_ptr<ServerConfig> config) = 0;
    virtual std::shared_ptr<ServerConfig> getConfig() const = 0;
    virtual unsigned int getPort() const = 0;
    virtual void setPort(unsigned int port) = 0;

    virtual int getState() const = 0;
    virtual void setState(int state) = 0;

    virtual int getFd() const = 0;
    virtual void setFd(int fd) = 0;
    virtual operator int() const noexcept = 0;

    virtual std::shared_ptr<net::INetwork> getNetwork() const = 0;
    virtual void setNetwork(std::shared_ptr<net::INetwork> network) = 0;

    virtual void onClientConnected(int idClient) = 0;
    virtual void onClientDisconnected(int idClient) = 0;
    virtual void onPacketReceived(int idClient, const IPacket &packet) = 0;

    virtual void processConnections() = 0;
    virtual void processIncomingPackets() = 0;

    virtual void broadcastPacket(const IPacket &packet) = 0;
    virtual void sendToClient(int idClient, const IPacket &packet) = 0;
    virtual std::vector<int> getConnectedClients() const = 0;
    virtual int getClientCount() const = 0;
};
} // namespace rserv = r-type server
#endif /* !ISERVER_HPP_ */
```

5.50 Server.hpp

```
/*
** EPITECH PROJECT, 2025
** R-Type
** File description:
** Header
*/

#ifndef SERVER_HPP_
    #define SERVER_HPP_

#include <memory>
#include "IServer.hpp"
#include "ServerConfig.hpp"
#include "../libs/Network/INetwork.hpp"
#include "../libs/Buffer/IBuffer.hpp"
#include "../common/DLLoader/DLLoader.hpp"

typedef void *(*createNetworkLib_t)();
typedef void *(*createBuffer_t)();
typedef void *(*createPacket_t)();

#define pathLoad "./librairies"
#define networkLib "libNetwork.so"
#define bufferLib "libBuffer.so"
#define packetLib "libPacket.so"

namespace rserv {
    class Server : public IServer {
        public:
            Server();
            ~Server();

            void init() override;
            void start() override;
            void stop() override;

            void setConfig(std::shared_ptr<ServerConfig> config) override;
            std::shared_ptr<ServerConfig> getConfig() const override;
            unsigned int getPort() const override;
            void setPort(unsigned int port) override;

            int getState() const override;
            void setState(int state) override;

            int getFd() const override;
            void setFd(int fd) override;
            operator int() const noexcept override;

            std::shared_ptr<net::INetwork> getNetwork() const override;
            void setNetwork(std::shared_ptr<net::INetwork> network) override;

            void onClientConnected(int idClient) override;
            void onClientDisconnected(int idClient) override;
            void onPacketReceived(int idClient, const IPacket &packet) override;

            void processConnections() override;
            void processIncomingPackets() override;

            void broadcastPacket(const IPacket &packet) override;
            void sendToClient(int idClient, const IPacket &packet) override;
            std::vector<int> getConnectedClients() const override;
            int getClientCount() const override;

        private:
            void loadNetworkLibrary();
            void loadBufferLibrary();
            void loadPacketLibrary();
            DLLoader<createNetworkLib_t> _networloader;
            DLLoader<createBuffer_t> _bufferloader;
            DLLoader<createPacket_t> _packetloader;

            std::shared_ptr<ServerConfig> _config;
            std::shared_ptr<net::INetwork> _network;
            std::shared_ptr<IBuffer> _buffer;
            std::shared_ptr<IPacket> _packet;

    };
} // namespace rserv = r-type server

#endif /* !SERVER_HPP_ */
```

5.51 ServerConfig.hpp

```
/*
** EPITECH PROJECT, 2025
** Header
** File description:
** Header
*/

#include <cstdint>

#ifndef SERVER_CONFIG_HPP_
    #define SERVER_CONFIG_HPP_

namespace rserv {
    class ServerConfig {
        public:
            ServerConfig();
            ~ServerConfig();

            int getState() const;
            int getFd() const;

            void setPort(unsigned int port);
            unsigned int getPort() const;

            void setState(int state);
            void setFd(int fd);

            void setNbClients(int nbClients);
            int getNbClients() const;

            uint32_t getIp() const;
            void setIp(uint32_t ip);

        private:
            int _state;
            int _fd;
            unsigned int _port;
            int _nbClients;
            uint32_t _ip;
    };
} // namespace rserv = r-type server

#endif /* !SERVER_CONFIG_HPP_ */
```

5.52 Utils.hpp

```
/*
** EPITECH PROJECT, 2025
** ryanR-type
** File description:
** Utils
*/

#include <vector>
#include <memory>

#include "ServerConfig.hpp"

#ifndef UTILS_HPP_
#define UTILS_HPP_

class Utils {
    public:
        Utils();
        ~Utils();

        void helper();
        void parsCli(int ac, char **av, std::shared_ptr<rserv::ServerConfig> config);
    protected:
    private:
};

#endif /* !UTILS_HPP_ */
```

Index