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

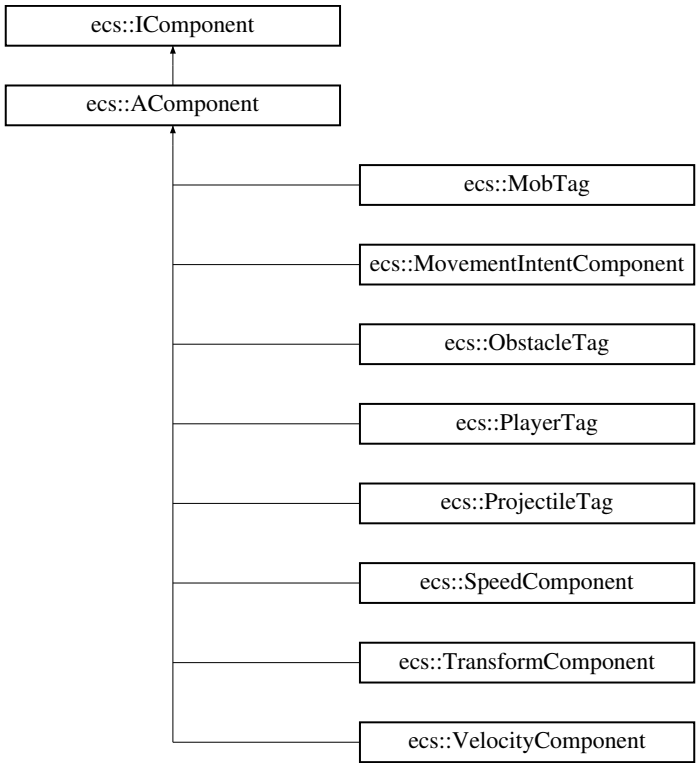
Here is a list of all documented files with brief descriptions:

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]
```

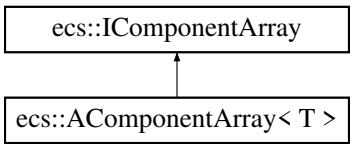
Implements ecs::IComponent.

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]
```

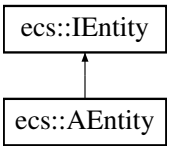
Implements ecs::IComponentArray.

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]
```

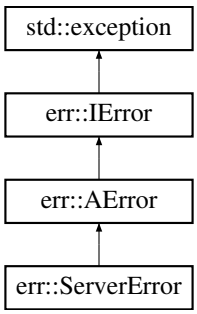
Implements ecs::IEntity.

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]
```

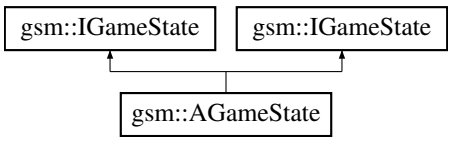
Implements err::IError.

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]
```

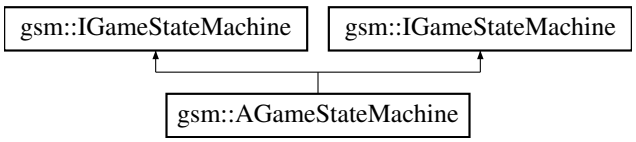
Implements gsm::IGameState.

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]
```

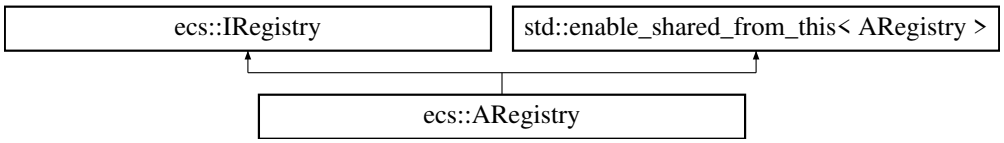
Implements gsm::IGameStateMachine.

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]
```

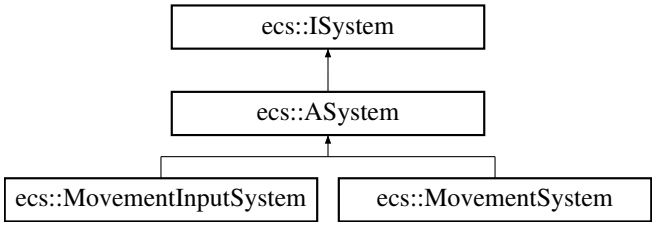
Implements ecs::IRegistry.

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]
```

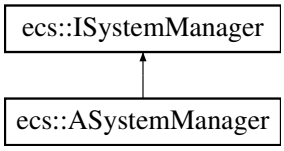
Implements ecs::ISystem.

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

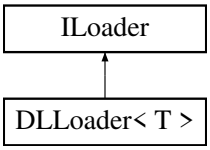
- int **r**
- int **g**
- int **b**

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

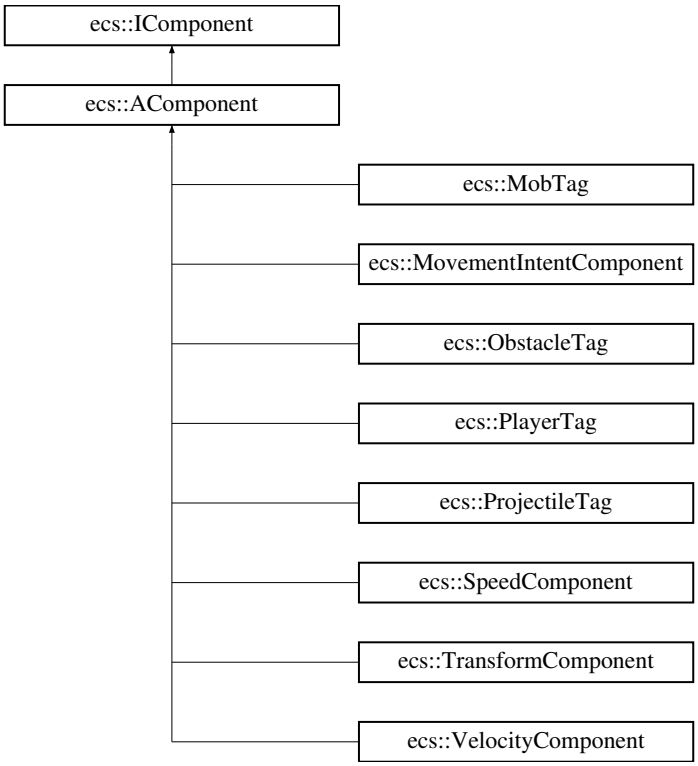
- std::shared_ptr< class ARegistry > **_registry**

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

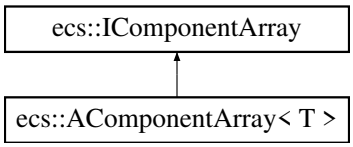
- virtual ComponentState **getState** () const =0
- virtual void **setState** (ComponentState newState)=0

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

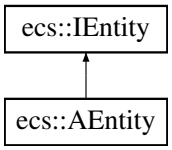
- virtual void **removeAllComponentsWithState** (ComponentState state)=0
- virtual size_t **getMaxEntityId** () const =0

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

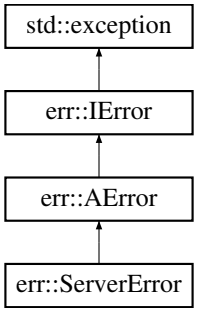
- **IEntity** (size_t id)

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

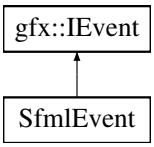
- virtual const char * **what** () const noexcept override=0
- virtual int **getCode** () const noexcept=0
- virtual std::string **getType** () const noexcept=0
- virtual std::string **getDetails** () const noexcept=0

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

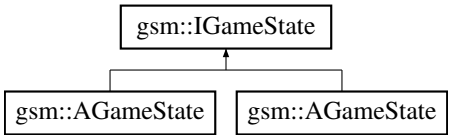
- virtual void **init** ()=0
- virtual event_t **pollEvents** ()=0
- virtual void **cleanup** ()=0
- virtual std::pair< int, int > **getMousePos** ()=0
- virtual bool **isKeyPressed** (event_t key)=0
- virtual bool **isMouseButtonPressed** (int button)=0

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

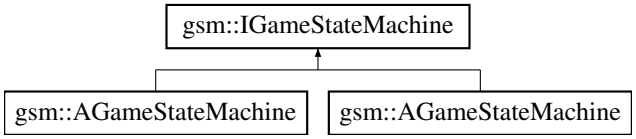
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **exit** ()=0
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **exit** ()=0

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

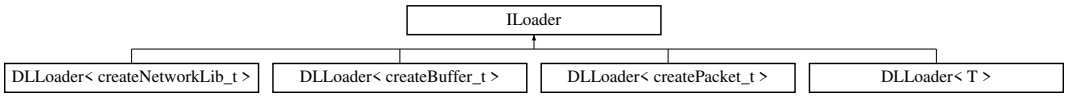
- virtual void **changeState** (std::shared_ptr< IGameState > newState)=0
- virtual void **pushState** (std::shared_ptr< IGameState > newState)=0
- virtual void **popState** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **changeState** (std::shared_ptr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



Public Member Functions

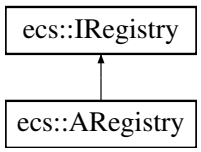
- virtual void * **Open** (const char *path, int flag)=0
- virtual void * **Symbol** (const char *symbolName)=0
- virtual int **Close** ()=0
- virtual const char * **Error** ()=0
- virtual void * **getHandler** () const =0

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

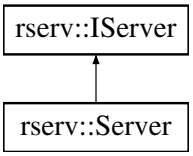
- template<typename T>
  void **registerComponent** ()
- template<typename T>
  void **addComponent** (int entityId, std::shared_ptr< T > component)
- template<typename T>
  std::shared_ptr< T > **getComponent** (int entityId)
- template<typename T>
  void **removeComponent** (int entityId)
- template<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

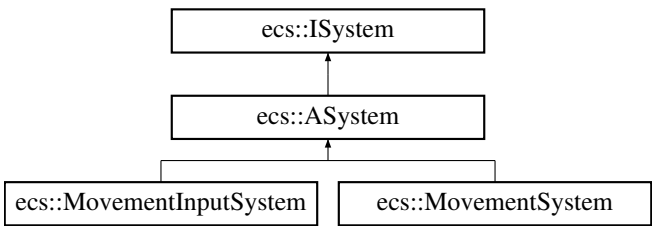
- virtual void **init** ()=0
- virtual void **start** ()=0
- virtual void **stop** ()=0
- virtual void **setConfig** (std::shared_ptr< ServerConfig > config)=0
- virtual std::shared_ptr< ServerConfig > **getConfig** () const =0
- virtual unsigned int **getPort** () const =0
- virtual void **setPort** (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

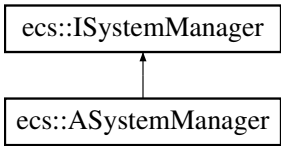
- virtual void **updateSystem** (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime)=0

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 AResgistry > **_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 AResgistry > 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

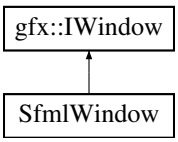
- std::shared_ptr< class AResgistry > **_registry**
- size_t **_entityId**
- size_t **_maxEntityId**

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

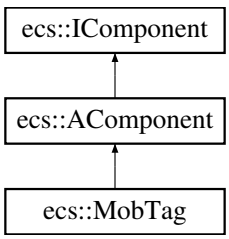
- static std::map< IEvent::event_t, sf::Keyboard::Key > **createKeyboardMapping** ()
- static std::map< IEvent::event_t, sf::Mouse::Button > **createMouseMapping** ()
- static std::map< IEvent::event_t, unsigned int > **createJoystickButtonMapping** ()

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

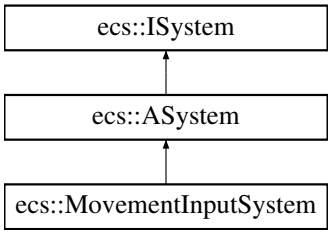
- ComponentState **_state** = Permanent

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]
```

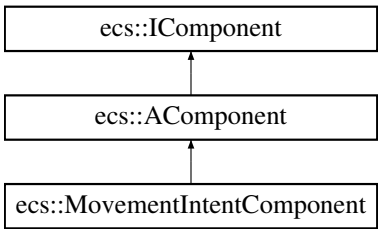
Implements ecs::ASystem.

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

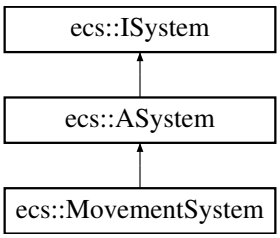
- ComponentState **_state** = Permanent

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]
```

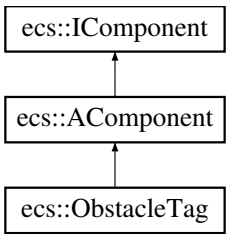
Implements ecs::ASystem.

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

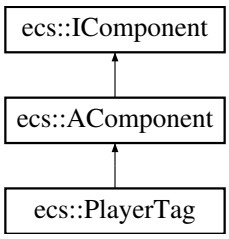
- ComponentState **_state** = Permanent

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

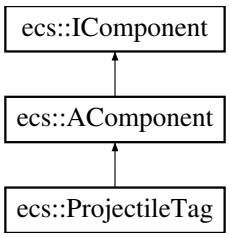
- ComponentState **_state** = Permanent

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

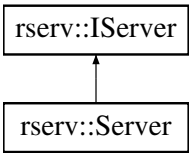
- std::unordered_map< size_t, std::shared_ptr< void > > **resources**

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

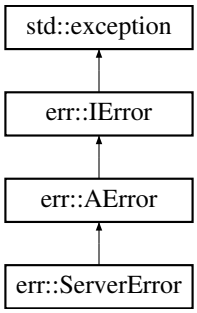
- int **_state**
- int **_fd**
- unsigned int **_port**
- int **_nbClients**
- uint32_t **_ip**

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]
```

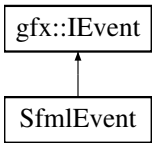
Implements err::AError.

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]
```

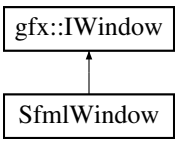
Implements gfx::IEvent.

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]
```

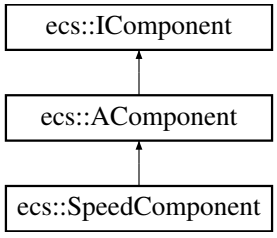
Implements gfx::IWindow.

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

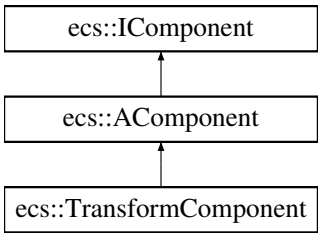
- ComponentState **_state** = Permanent

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

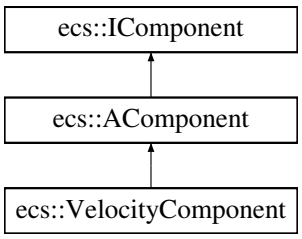
- float **_x**
- float **_y**

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

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** EventTypes
00006 */
00007
00008 #ifndef EVENTTYPES_HPP_
00009 #define EVENTTYPES_HPP_
00010
00011 namespace gfx {
00012
00013 enum class EventType {
00014     // Directional keys
00015     UP,
00016     DOWN,
00017     LEFT,
00018     RIGHT,
00019
00020     // Alphanumeric keys
00021     A, B, C, D, E, F, G, H, I, J, K, L, M,
00022     N, O, P, Q, R, S, T, U, V, W, X, Y, Z,
00023     NUM0, NUM1, NUM2, NUM3, NUM4, NUM5, NUM6, NUM7, NUM8, NUM9,
00024
00025     // Special keys
00026     SPACE,
00027     ENTER,
00028     ESCAPE,
00029     TAB,
00030     BACKSPACE,
00031     DELETE_KEY,
00032     INSERT,
00033     HOME,
00034     END,
00035     PAGEUP,
00036     PAGEDOWN,
00037
00038     // Function keys
00039     F1, F2, F3, F4, F5, F6, F7, F8, F9, F10, F11, F12,
00040
00041     // Modifiers
00042     LSHIFT,
00043     RSHIFT,
00044     LCTRL,
00045     RCTRL,
00046     LALT,
00047     RALT,
00048
00049     // Symbols
00050     MINUS,
00051     PLUS,
00052     EQUALS,
00053     LBRACKET,
00054     RBRACKET,
00055     SEMICOLON,
00056     QUOTE,
00057     COMMA,
```

```
00058     PERIOD,
00059     SLASH,
00060     BACKSLASH,
00061     TILDE,
00062
00063     // Numpad
00064     NUMPAD0, NUMPAD1, NUMPAD2, NUMPAD3, NUMPAD4,
00065     NUMPAD5, NUMPAD6, NUMPAD7, NUMPAD8, NUMPAD9,
00066     NUMPAD_ADD,
00067     NUMPAD_SUBTRACT,
00068     NUMPAD_MULTIPLY,
00069     NUMPAD_DIVIDE,
00070     NUMPAD_ENTER,
00071
00072     // Mouse buttons
00073     MOUSECLICK,
00074     MOUSERIGHTCLICK,
00075     MOUSELEFTCLICK,
00076     MOUSEMIDDLECLICK,
00077     MOUSEX1,
00078     MOUSEX2,
00079     MOUSEWHEELUP,
00080     MOUSEWHEELDOWN,
00081
00082     // Gamepad/Controller buttons
00083     GAMEPAD_A,
00084     GAMEPAD_B,
00085     GAMEPAD_X,
00086     GAMEPAD_Y,
00087     GAMEPAD_LEFT_BUMPER,
00088     GAMEPAD_RIGHT_BUMPER,
00089     GAMEPAD_BACK,
00090     GAMEPAD_START,
00091     GAMEPAD_GUIDE,
00092     GAMEPAD_LEFT_THUMB,
00093     GAMEPAD_RIGHT_THUMB,
00094
00095     // Gamepad D-Pad
00096     GAMEPAD_DPAD_UP,
00097     GAMEPAD_DPAD_DOWN,
00098     GAMEPAD_DPAD_LEFT,
00099     GAMEPAD_DPAD_RIGHT,
00100
00101     // Gamepad triggers
00102     GAMEPAD_LEFT_TRIGGER,
00103     GAMEPAD_RIGHT_TRIGGER,
00104
00105     // Gamepad analog sticks
00106     GAMEPAD_LEFT_STICK_UP,
00107     GAMEPAD_LEFT_STICK_DOWN,
00108     GAMEPAD_LEFT_STICK_LEFT,
00109     GAMEPAD_LEFT_STICK_RIGHT,
00110     GAMEPAD_RIGHT_STICK_UP,
00111     GAMEPAD_RIGHT_STICK_DOWN,
00112     GAMEPAD_RIGHT_STICK_LEFT,
00113     GAMEPAD_RIGHT_STICK_RIGHT,
00114
00115     // Window events
00116     CLOSE,
00117     REFRESH,
00118
00119     // Default
00120     NOTHING
00121 };
00122
00123 } // namespace gfx
00124
00125 #endif /* !EVENTTYPES_HPP_ */
```

5.2 IEvent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** IEvent
00006 */
00007
00008 #ifndef IEVENT_HPP_
00009 #define IEVENT_HPP_
00010
00011 #include <utility>
00012 #include <memory>
```

```
00013 #include "EventTypes.hpp"
00014
00015 namespace gfx {
00016
00017 class IEvent {
00018     public:
00019         using event_t = EventType;
00020         virtual ~IEvent() = default;
00021         virtual void init() = 0;
00022         virtual event_t pollEvents() = 0;
00023         virtual void cleanup() = 0;
00024         virtual std::pair<int, int> getMousePos() = 0;
00025         virtual bool isKeyPressed(event_t key) = 0;
00026         virtual bool isMouseButtonPressed(int button) = 0;
00027
00028 };
00029
00030 } // namespace gfx
00031
00032 #endif /* !IEVENT_HPP_ */
```

5.3 IWindow.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** IWindow
00006 */
00007
00008
00009 #ifndef IWINDOW_HPP_
00010 #define IWINDOW_HPP_
00011
00012 #include <string>
00013 #include <utility>
00014 #include <memory>
00015
00016 namespace gfx {
00017
00018 struct color_t {
00019     int r;
00020     int g;
00021     int b;
00022 };
00023
00024 class IWindow {
00025     public:
00026         virtual ~IWindow() = default;
00027         virtual void init() = 0;
00028         virtual void display() = 0;
00029         virtual void closeWindow() = 0;
00030         virtual bool isOpen() = 0;
00031         virtual void clear() = 0;
00032
00033         virtual void resizeWindow(size_t x, size_t y) = 0;
00034
00035         virtual void drawSprite(std::string asset, color_t color, std::pair<size_t, size_t> position)
      = 0;
00036         virtual void drawText(std::string text, color_t color, std::pair<size_t, size_t> position) =
      0;
00037         virtual void drawRectangle(color_t color, std::pair<size_t, size_t> position,
      std::pair<size_t, size_t> size) = 0;
00038
00039         virtual void setFont(const std::string& fontPath) = 0;
00040         virtual std::string getFont() const = 0;
00041
00042         virtual bool isMouseOver(std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) =
      0;
00043         virtual std::pair<int, int> getWindowSize() = 0;
00044 };
00045
00046 } // namespace gfx
00047
00048 #endif /* !IWINDOW_HPP_ */
```

5.4 SfmlEvent.hpp

```
00001 /*
```

```
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SfmlEvent
00006 */
00007
00008 #ifndef SFMLEVENT_HPP_
00009 #define SFMLEVENT_HPP_
00010
00011 #include <SFML/Graphics.hpp>
00012 #include <map>
00013 #include "./IEvent.hpp"
00014 #include "./IWindow.hpp"
00015 #include "./SfmlKeyMappings.hpp"
00016 #include "../../common/ECS/resourceManager/ResourceManager.hpp"
00017
00018 class SfmlEvent : public gfx::IEvent {
00019     public:
00020         SfmlEvent(std::shared_ptr<ecs::ResourceManager> resourceManager, std::shared_ptr<gfx::IWindow>
      window);
00021         ~SfmlEvent() override;
00022         void init() override;
00023         event_t pollEvents() override;
00024         void cleanup() override;
00025         std::pair<int, int> getMousePos() override;
00026         bool isKeyPressed(event_t key) override;
00027         bool isMouseButtonPressed(int button) override;
00028
00029     private:
00030         void initializeMappings();
00031         void createReverseKeyboardMapping();
00032         event_t processKeyboardEvent(const sf::Event::KeyPressed& keyPressed);
00033         event_t processMouseEvent(const sf::Event::MouseButtonPressed& mousePressed);
00034         event_t processJoystickButtonEvent(const sf::Event::JoystickButtonPressed& joystickPressed);
00035         event_t processJoystickAxisEvent(const sf::Event::JoystickMoved& joystickMoved);
00036
00037         // Member variables
00038         std::shared_ptr<ecs::ResourceManager> _resourceManager;
00039         std::shared_ptr<gfx::IWindow> _window;
00040         std::map<event_t, sf::Keyboard::Key> _keyMap;
00041         std::map<sf::Keyboard::Key, event_t> _reverseKeyMap;
00042         std::map<event_t, sf::Mouse::Button> _mouseMap;
00043         std::map<event_t, unsigned int> _joystickButtonMap;
00044 };
00045
00046 #endif /* !SFMLEVENT_HPP_ */
```

5.5 SfmlKeyMappings.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** KeyMappings
00006 */
00007
00008 #ifndef KEYMAPPINGS_HPP_
00009 #define KEYMAPPINGS_HPP_
00010
00011 #include <SFML/Graphics.hpp>
00012 #include <map>
00013 #include "IEvent.hpp"
00014 #include "EventTypes.hpp"
00015
00016 namespace gfx {
00017
00018 class KeyMappings {
00019     public:
00020         static std::map<IEvent::event_t, sf::Keyboard::Key> createKeyboardMapping();
00021         static std::map<IEvent::event_t, sf::Mouse::Button> createMouseMapping();
00022         static std::map<IEvent::event_t, unsigned int> createJoystickButtonMapping();
00023 };
00024
00025 } // namespace gfx
00026
00027 #endif /* !KEYMAPPINGS_HPP_ */
```

5.6 SfmlWindow.hpp

```
00001 /*
```

```
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SfmlWindow
00006 */
00007
00008 #ifndef SFMLWINDOW_HPP_
00009 #define SFMLWINDOW_HPP_
00010
00011 #include <SFML/Graphics.hpp>
00012 #include "./IWindow.hpp"
00013
00014 class SfmlWindow : public gfx::IWindow {
00015     public:
00016         SfmlWindow(std::string title = "R-Type", size_t width = 800, size_t height = 600);
00017         ~SfmlWindow() override;
00018         void init() override;
00019         void display() override;
00020         void closeWindow() override;
00021         bool isOpen() override;
00022         void clear() override;
00023
00024         void resizeWindow(size_t x, size_t y) override;
00025
00026         void drawSprite(std::string asset, gfx::color_t color, std::pair<size_t, size_t> position)
    override;
00027         void drawText(std::string text, gfx::color_t color, std::pair<size_t, size_t> position)
    override;
00028         void drawRectangle(gfx::color_t color, std::pair<size_t, size_t> position, std::pair<size_t,
    size_t> size) override;
00029
00030         void setFont(const std::string& fontPath) override;
00031         std::string getFont() const override;
00032
00033         std::shared_ptr<sf::RenderWindow> getSfmlWindow();
00034         bool isMouseOver(std::pair<size_t, size_t> position, std::pair<size_t, size_t> size) override;
00035         std::pair<int, int> getWindowSize() override;
00036     private:
00037         std::shared_ptr<sf::RenderWindow> _window;
00038         sf::Font _font;
00039         std::string _fontPath;
00040 };
00041
00042 #endif /* !SFMLWINDOW_HPP_ */
```

5.7 AGameStateMachine.hpp

```
00001 #pragma once
00002
00003 #include "IGameStateMachine.hpp"
00004 #include "../states/IGameState.hpp"
00005
00006 namespace gsm {
00007
00008 class AGameStateMachine : public IGameStateMachine {
00009 public:
00010     AGameStateMachine();
00011     ~AGameStateMachine() override = default;
00012
00013     void changeState(std::shared_ptr<IGameState> newState) override;
00014     void pushState(std::shared_ptr<IGameState> newState) override;
00015     void popState() override;
00016
00017     void update(float deltaTime) override;
00018     void render() override;
00019
00020 protected:
00021     std::stack<std::shared_ptr<IGameState>> _states;
00022 };
00023
00024 } // namespace gsm
```

5.8 AGameStateMachine.hpp

```
00001 #pragma once
00002
00003 #include "IGameStateMachine.hpp"
00004 #include "../states/IGameState.hpp"
```

```
00005
00006 namespace gsm {
00007
00008 class AGameStateMachine : public IGameStateMachine {
00009 public:
00010     AGameStateMachine();
00011     ~AGameStateMachine() override = default;
00012
00013     void changeState(std::shared_ptr<IGameState> newState) override;
00014     void pushState(std::shared_ptr<IGameState> newState) override;
00015     void popState() override;
00016
00017     void update(float deltaTime) override;
00018
00019 protected:
00020     std::stack<std::shared_ptr<IGameState» _states;
00021 };
00022
00023 } // namespace gsm
```

5.9 IGameStateMachine.hpp

```
00001 #pragma once
00002
00003 #include <memory>
00004 #include <stack>
00005
00006 namespace gsm {
00007
00008 class IGameState;
00009
00010 class IGameStateMachine {
00011 public:
00012     virtual ~IGameStateMachine() = default;
00013
00014     virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
00015     virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
00016     virtual void popState() = 0;
00017
00018     virtual void update(float deltaTime) = 0;
00019     virtual void render() = 0;
00020 };
00021
00022 } // namespace gsm
```

5.10 IGameStateMachine.hpp

```
00001 #pragma once
00002
00003 #include <memory>
00004 #include <stack>
00005
00006 namespace gsm {
00007
00008 class IGameState;
00009
00010 class IGameStateMachine {
00011 public:
00012     virtual ~IGameStateMachine() = default;
00013
00014     virtual void changeState(std::shared_ptr<IGameState> newState) = 0;
00015     virtual void pushState(std::shared_ptr<IGameState> newState) = 0;
00016     virtual void popState() = 0;
00017
00018     virtual void update(float deltaTime) = 0;
00019 };
00020
00021 } // namespace gsm
```

5.11 AGameState.hpp

```
00001 #pragma once
00002
```

```
00003 #include "IGameState.hpp"
00004
00005 namespace gsm {
00006
00007 class AGameState : public IGameState {
00008 public:
00009     AGameState(std::shared_ptr<IGameStateMachine> gsm);
00010     ~AGameState() override = default;
00011
00012     void enter() override;
00013     void update(float deltaTime) override;
00014     void render() override;
00015     void exit() override;
00016
00017 protected:
00018     std::shared_ptr<IGameStateMachine> _gsm;
00019 };
00020
00021 } // namespace gsm
```

5.12 AGameState.hpp

```
00001 #pragma once
00002
00003 #include "IGameState.hpp"
00004
00005 namespace gsm {
00006
00007 class AGameState : public IGameState {
00008 public:
00009     AGameState(std::shared_ptr<IGameStateMachine> gsm);
00010     ~AGameState() override = default;
00011
00012     void enter() override;
00013     void update(float deltaTime) override;
00014     void exit() override;
00015
00016 protected:
00017     std::shared_ptr<IGameStateMachine> _gsm;
00018 };
00019
00020 } // namespace gsm
```

5.13 IGameState.hpp

```
00001 #pragma once
00002
00003 #include <memory>
00004 #include "../machine/IGameStateMachine.hpp"
00005
00006 namespace gsm {
00007
00008 class IGameState {
00009 public:
00010     virtual ~IGameState() = default;
00011
00012     virtual void enter() = 0;
00013     virtual void update(float deltaTime) = 0;
00014     virtual void render() = 0;
00015     virtual void exit() = 0;
00016 };
00017
00018 } // namespace gsm
```

5.14 IGameState.hpp

```
00001 #pragma once
00002
00003 #include <memory>
00004 #include "../machine/IGameStateMachine.hpp"
00005
00006 namespace gsm {
00007
```

```
00008 class IGameState {
00009 public:
00010     virtual ~IGameState() = default;
00011
00012     virtual void enter() = 0;
00013     virtual void update(float deltaTime) = 0;
00014     virtual void exit() = 0;
00015 };
00016
00017 } // namespace gsm
```

5.15 initRessourcesManager.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** initRessourcesManager
00006 */
00007
00008 #ifndef INITRESSOURCESMANAGER_HPP_
00009 #define INITRESSOURCESMANAGER_HPP_
00010
00011 #include "../../common/ECS/resourceManager/ResourceManager.hpp"
00012 #include <memory>
00013
00014 std::shared_ptr<ecs::ResourceManager> initRessourcesManager();
00015
00016 #endif /* !INITRESSOURCESMANAGER_HPP_ */
```

5.16 initRessourcesManager.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** initRessourcesManager
00006 */
00007
00008 #ifndef INITRESSOURCESMANAGER_HPP_
00009 #define INITRESSOURCESMANAGER_HPP_
00010
00011 #include "../../common/ECS/resourceManager/ResourceManager.hpp"
00012 #include <memory>
00013
00014 std::shared_ptr<ecs::ResourceManager> initRessourcesManager();
00015
00016 #endif /* !INITRESSOURCESMANAGER_HPP_ */
```

5.17 constants.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Constants
00006 */
00007
00008 #ifndef CONSTANTS_HPP_
00009 #define CONSTANTS_HPP_
00010
00011 namespace constants {
00012     constexpr float BASE_SPEED = 100.0f;
00013 }
00014
00015 #endif /* !CONSTANTS_HPP_ */
```

5.18 DLLoader.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** DLLoader
00006 */
00007
00008 #ifndef DLLOADER_HPP_
00009 #define DLLOADER_HPP_
00010
00011 #ifdef _WIN32
00012     #include <windows.h>
00013     #define RTLD_LAZY 0
00014 #else
00015     #include <dlfcn.h>
00016 #endif
00017
00018 #include <iostream>
00019 #include <ostream>
00020 #include <memory>
00021 #include "ILoader.hpp"
00022
00023 template <typename T>
00024
00025 class DLLoader : public ILoader {
00026     private:
00027 #ifdef _WIN32
00028         HMODULE _handler = nullptr;
00029         mutable std::string _lastError;
00030 #else
00031         void *_handler = nullptr;
00032 #endif
00033
00034     public:
00035         ~DLLoader() = default;
00036
00037         void *getHandler() const override {
00038             return _handler;
00039         };
00040         void *Open(const char *path, int flag = RTLD_LAZY) override {
00041 #ifdef _WIN32
00042             _handler = LoadLibraryA(path);
00043             if (!_handler) {
00044                 _lastError = "Failed to load library: " + std::string(path);
00045             }
00046 #else
00047             _handler = dlopen(path, flag);
00048 #endif
00049             return _handler;
00050         };
00051         void *Symbol(const char *symbolName) override {
00052 #ifdef _WIN32
00053             void *symbol = (void*)GetProcAddress(_handler, symbolName);
00054             if (!symbol) {
00055                 _lastError = "Failed to get symbol: " + std::string(symbolName);
00056                 std::cerr << "GetProcAddress error: " << _lastError << std::endl;
00057                 return nullptr;
00058             }
00059             return symbol;
00060 #else
00061             void *symbol = dlsym(_handler, symbolName);
00062             const char *error = dlerror();
00063             if (error) {
00064                 std::cerr << "dlerror: " << error << std::endl;
00065                 return nullptr;
00066             }
00067             return symbol;
00068 #endif
00069         };
00070         T getSymbol(const char *symbolName) {
00071 #ifdef _WIN32
00072             return reinterpret_cast<T>(GetProcAddress(_handler, symbolName));
00073 #else
00074             return reinterpret_cast<T>(dlsym(_handler, symbolName));
00075 #endif
00076         };
00077         int Close() override{
00078             if (_handler == nullptr)
00079                 return -1;
00080 #ifdef _WIN32
00081             return FreeLibrary(_handler) ? 0 : -1;
00082 #else
00083             return dlclose(_handler);
00084 #endif
```

```
00085         };
00086         const char *Error() override {
00087 #ifdef _WIN32
00088             return _lastError.c_str();
00089 #else
00090             return dlerror();
00091 #endif
00092         };
00093 };
00094
00095 #endif /* !DLLOADER_HPP_ */
```

5.19 ILoader.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** ILoader
00006 */
00007
00008 #ifndef ILoader_HPP_
00009 #define ILoader_HPP_
00010
00011
00012 class ILoader {
00013     public:
00014         ~ILoader() = default;
00015
00016         virtual void *Open(const char *path, int flag) = 0;
00017         virtual void *Symbol(const char *symbolName) = 0;
00018         virtual int Close() = 0;
00019         virtual const char *Error() = 0;
00020         virtual void *getHandler() const = 0;
00021
00022     protected:
00023     private:
00024 };
00025
00026 #endif /* !ILoader_HPP_ */
```

5.20 LoaderType.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** r-type
00004 ** File description:
00005 ** LoaderType
00006 */
00007
00008 #ifndef LOADERTYPE_HPP_
00009 #define LOADERTYPE_HPP_
00010
00011 enum ModuleType_t{
00012     DISPLAY_MODULE = 0,
00013     NETWORK_MODULE = 1,
00014     PACKET_MODULE = 2,
00015     BUFFER_MODULE = 3,
00016     UNKNOWN_MODULE
00017 };
00018
00019 typedef ModuleType_t (*getTypeFunc_t)();
00020
00021 #endif /* !LOADERTYPE_HPP_ */
```

5.21 AComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AComponent
00006 */
```

```
00007
00008 #ifndef ACOMPONENT_HPP_
00009 #define ACOMPONENT_HPP_
00010
00011 #include "IComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class AComponent : public IComponent {
00016     public:
00017         AComponent();
00018         ~AComponent();
00019
00020         ComponentState getState() const override;
00021         void setState(ComponentState newState) override;
00022
00023     protected:
00024         ComponentState _state = Permanent;
00025
00026     private:
00027 };
00028
00029 } // namespace ecs
00030
00031 #endif /* !ACOMPONENT_HPP_ */
```

5.22 IComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IComponent
00006 */
00007
00008 #ifndef ICOMPONENT_HPP_
00009 #define ICOMPONENT_HPP_
00010
00011 namespace ecs {
00012
00013 enum ComponentState {
00014     Permanent = 0,
00015     Temporary = 1,
00016     Processed = 2,
00017 };
00018
00019 class IComponent {
00020     public:
00021         IComponent() = default;
00022         virtual ~IComponent() = default;
00023
00024         virtual ComponentState getState() const = 0;
00025         virtual void setState(ComponentState newState) = 0;
00026
00027     protected:
00028     private:
00029 };
00030
00031 } // namespace ecs
00032
00033 #endif /* !ICOMPONENT_HPP_ */
```

5.23 SpeedComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SpeedComponent
00006 */
00007
00008 #ifndef SPEEDCOMPONENT_HPP_
00009 #define SPEEDCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "constants.hpp"
00013
00014 namespace ecs {
```

```
00015
00016 class SpeedComponent : public AComponent {
00017     public:
00018         SpeedComponent(float speed = constants::BASE_SPEED) : _speed(speed) {};
00019         ~SpeedComponent() = default;
00020
00021         float getSpeed() const { return _speed; };
00022         void setSpeed(float speed) { _speed = speed; };
00023     private:
00024         float _speed;
00025 };
00026
00027 }  // namespace ecs
00028
00029 #endif /* !SPEEDCOMPONENT_HPP_ */
```

5.24 TransformComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** TransformComponent
00006 */
00007
00008 #ifndef TRANSFORMCOMPONENT_HPP_
00009 #define TRANSFORMCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class TransformComponent : public AComponent {
00017     public:
00018         TransformComponent(math::Vector2f position = math::Vector2f(0.0f, 0.0f), float rotation =
      0.0f, math::Vector2f scale = math::Vector2f(1.0f, 1.0f))
00019             : _position(position), _rotation(rotation), _scale(scale) {};
00020         ~TransformComponent() = default;
00021
00022         math::Vector2f getPosition() const { return _position; };
00023         void setPosition(math::Vector2f position) { _position = position; };
00024
00025         float getRotation() const { return _rotation; };
00026         void setRotation(float rotation) { _rotation = rotation; };
00027
00028         math::Vector2f getScale() const { return _scale; };
00029         void setScale(math::Vector2f scale) { _scale = scale; };
00030
00031     private:
00032         math::Vector2f _position;
00033         float _rotation;
00034         math::Vector2f _scale;
00035 };
00036
00037 } // namespace ecs
00038
00039 #endif /* !TRANSFORMCOMPONENT_HPP_ */
```

5.25 VelocityComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** VelocityComponent
00006 */
00007
00008 #ifndef VELOCITYCOMPONENT_HPP_
00009 #define VELOCITYCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class VelocityComponent : public AComponent {
00017     public:
```

```
00018         VelocityComponent(math::Vector2f velocity = math::Vector2f(0.0f, 0.0f)) : _velocity(velocity)
      {};
00019         ~VelocityComponent() = default;
00020
00021         math::Vector2f getVelocity() const { return _velocity; };
00022         void setVelocity(math::Vector2f velocity) { _velocity = velocity; };
00023     private:
00024         math::Vector2f _velocity;
00025 };
00026
00027 } // namespace ecs
00028
00029 #endif /* !VELOCITYCOMPONENT_HPP_ */
```

5.26 MobTag.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MobTag
00006 */
00007
00008 #ifndef MOBTAG_HPP_
00009 #define MOBTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class MobTag : public AComponent {
00016     public:
00017         MobTag() = default;
00018         ~MobTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !MOBTAG_HPP_ */
```

5.27 ObstacleTag.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ObstacleTag
00006 */
00007
00008 #ifndef OBSTACLETAG_HPP_
00009 #define OBSTACLETAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ObstacleTag : public AComponent {
00016     public:
00017         ObstacleTag() = default;
00018         ~ObstacleTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !OBSTACLETAG_HPP_ */
```

5.28 PlayerTag.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** PlayerTag
```

```
00006 */
00007
00008 #ifndef PLAYERTAG_HPP_
00009 #define PLAYERTAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class PlayerTag : public AComponent {
00016     public:
00017         PlayerTag() = default;
00018         ~PlayerTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !PLAYERTAG_HPP_ */
```

5.29 ProjectileTag.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ProjectileTag
00006 */
00007
00008 #ifndef PROJECTILETAG_HPP_
00009 #define PROJECTILETAG_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class ProjectileTag : public AComponent {
00016     public:
00017         ProjectileTag() = default;
00018         ~ProjectileTag() = default;
00019 };
00020
00021 } // namespace ecs
00022
00023 #endif /* !PROJECTILETAG_HPP_ */
```

5.30 MovementIntentComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MovementIntentComponent
00006 */
00007
00008 #ifndef MOVEMENTINTENTCOMPONENT_HPP_
00009 #define MOVEMENTINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class MovementIntentComponent : public AComponent {
00017     public:
00018         MovementIntentComponent(const math::Vector2f &direction = math::Vector2f(0.0f, 0.0f), bool
      active = false)
00019             : _direction(direction), _active(active) {
00020             _state = Temporary;
00021         };
00022         ~MovementIntentComponent() = default;
00023
00024         math::Vector2f getDirection() const { return _direction; };
00025         void setDirection(const math::Vector2f &direction) { _direction = direction; };
00026
00027         bool isActive() const { return _active; };
00028         void setActive(bool active) { _active = active; };
00029
00030     private:
```

```
00031         math::Vector2f _direction;
00032         bool _active;
00033 };
00034
00035 }  // namespace ecs
00036
00037 #endif /* !MOVEMENTINTENTCOMPONENT_HPP_ */
```

5.31 AEntity.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IEntity
00006 */
00007
00008 #ifndef AENTITY_HPP_
00009 #define AENTITY_HPP_
00010
00011 #include "IEntity.hpp"
00012
00013 namespace ecs {
00014
00015 class AEntity : public IEntity {
00016     public:
00017         AEntity();
00018         ~AEntity();
00019         operator size_t() const override;
00020
00021     private:
00022         explicit AEntity(size_t id);
00023         size_t _id;
00024 };
00025
00026 }  // namespace ecs
00027
00028 #endif /* !AENTITY_HPP_ */
```

5.32 AComponentArray.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AComposantType
00006 */
00007
00008 #ifndef AComposantType_HPP_
00009 #define AComposantType_HPP_
00010
00011 #include "IComponentArray.hpp"
00012 #include "../../component/base/IComponent.hpp"
00013 #include <vector>
00014 #include <memory>
00015
00016 namespace ecs {
00017
00018 template <typename T>
00019 class AComponentArray : public IComponentArray {
00020     public:
00021         AComponentArray();
00022         ~AComponentArray() override;
00023
00024         void add(int entityId, std::shared_ptr<T> component);
00025         std::shared_ptr<T> get(int entityId) const;
00026         void remove(int entityId);
00027         bool has(int entityId) const;
00028
00029         void removeAllComponentsWithState(ComponentState state) override;
00030         size_t getMaxEntityId() const override;
00031
00032     private:
00033         std::vector<std::shared_ptr<T>> _components;
00034 };
00035
00036 } // namespace ecs
00037
```

```
00038 #include "AComponentArray.tpp"
00039
00040 #endif /* !AComposantType_HPP_ */
```

5.33 IComponentArray.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IComponentArray
00006 */
00007
00008 #ifndef ICOMPONENTARRAY_HPP_
00009 #define ICOMPONENTARRAY_HPP_
00010
00011 #include "../../component/base/IComponent.hpp"
00012
00013 namespace ecs {
00014
00015 class IComponentArray {
00016     public:
00017         virtual ~IComponentArray() = default;
00018         virtual void removeAllComponentsWithState(ComponentState state) = 0;
00019         virtual size_t getMaxEntityId() const = 0;
00020 };
00021
00022 } // namespace ecs
00023
00024 #endif /* !ICOMPONENTARRAY_HPP_ */
```

5.34 IEntity.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IEntity
00006 */
00007
00008 #ifndef IENTITY_HPP_
00009 #define IENTITY_HPP_
00010
00011 #include <cstddef>
00012
00013 namespace ecs {
00014
00015 class IEntity {
00016     public:
00017         IEntity() = default;
00018         virtual ~IEntity() = default;
00019         virtual operator size_t() const = 0;
00020
00021     private:
00022         explicit IEntity(size_t id);
00023 };
00024
00025 } // namespace ecs
00026
00027 #endif /* !IENTITY_HPP_ */
```

5.35 ARegistry.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ARegistry
00006 */
00007
00008 #ifndef AREGISTRY_HPP_
00009 #define AREGISTRY_HPP_
00010
```

```
00011 #include "IRegistry.hpp"
00012 #include "../../component/base/IComponent.hpp"
00013 #include "../componentArray/IComponentArray.hpp"
00014 #include "../componentArray/AComponentArray.hpp"
00015 #include "../../view/View.hpp"
00016 #include <memory>
00017 #include <unordered_map>
00018
00019 namespace ecs {
00020
00021 class ARegistry : public IRegistry, public std::enable_shared_from_this<ARegistry> {
00022     public:
00023         ARegistry();
00024         virtual ~ARegistry();
00025
00026         template <typename T>
00027         void registerComponent();
00028
00029         template <typename T>
00030         void addComponent(int entityId, std::shared_ptr<T> component);
00031         template <typename T>
00032         std::shared_ptr<T> getComponent(int entityId) const;
00033         template <typename T>
00034         void removeComponent(int entityId);
00035         template <typename T>
00036         bool hasComponent(int entityId) const;
00037
00038         template <typename... Components>
00039         View<Components...> view();
00040
00041         template <typename... Components>
00042         Group<Components...> group();
00043
00044         size_t getMaxEntityId() const;
00045
00046         void removeAllComponentsWithState(ComponentState state) override;
00047
00048     protected:
00049     private:
00050         std::unordered_map<const char *, std::shared_ptr<IComponentArray» _components;
00051 };
00052
00053 } // namespace ecs
00054
00055 #include "ARegistry.tpp"
00056
00057 #endif /* !AREGISTRY_HPP_ */
```

5.36 IRegistry.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IRegistry
00006 */
00007
00008 #ifndef IREGISTRY_HPP_
00009 #define IREGISTRY_HPP_
00010
00011 #include <memory>
00012 #include "../../component/base/IComponent.hpp"
00013 #include "../../view/View.hpp"
00014
00015 namespace ecs {
00016
00017 class IRegistry {
00018     public:
00019         virtual ~IRegistry() = default;
00020
00021         template <typename T>
00022         void registerComponent();
00023
00024         template <typename T>
00025         void addComponent(int entityId, std::shared_ptr<T> component);
00026
00027         template <typename T>
00028         std::shared_ptr<T> getComponent(int entityId);
00029
00030         template <typename T>
00031         void removeComponent(int entityId);
00032
00033         template <typename T>
```

```
00034         bool hasComponent(int entityId);
00035
00036         template <typename... Components>
00037         View<Components...> view();
00038
00039         template <typename... Components>
00040         Group<Components...> group();
00041
00042         size_t getMaxEntityId() const;
00043
00044         virtual void removeAllComponentsWithState(ComponentState state) = 0;
00045 };
00046
00047 } // namespace ecs
00048
00049 #endif /* !IREGISTRY_HPP_ */
```

5.37 ResourceManager.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ResourceManager
00006 */
00007
00008 #ifndef RESOURCEMANAGER_HPP_
00009 #define RESOURCEMANAGER_HPP_
00010
00011 #include <unordered_map>
00012 #include <memory>
00013
00014 namespace ecs {
00015
00016 class ResourceManager {
00017     public:
00018         template<typename T>
00019         void add(std::shared_ptr<T> resource);
00020
00021         template<typename T>
00022         std::shared_ptr<T> get();
00023
00024         template<typename T>
00025         bool has();
00026
00027     private:
00028         std::unordered_map<size_t, std::shared_ptr<void» resources;
00029 };
00030
00031 } // namespace ecs
00032
00033 #include "ResourceManager.tpp"
00034
00035 #endif /* !RESOURCEMANAGER_HPP_ */
```

5.38 ASystem.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ASystem
00006 */
00007
00008 #ifndef ASystem_HPP_
00009 #define ASystem_HPP_
00010
00011 #include <memory>
00012
00013 #include "ISystem.hpp"
00014 #include "../../resourceManager/ResourceManager.hpp"
00015 #include "../../entity/registry/ARegistry.hpp"
00016
00017 namespace ecs {
00018
00019 class ASystem : public ISystem {
00020     public:
00021         ASystem();
```

```
00022         ~ASystem() = default;
00023         void updateSystem(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry>
      registry, float deltaTime) override;
00024
00025     protected:
00026         virtual void update(std::shared_ptr<ResourceManager> resourceManager,
      std::shared_ptr<ARegistry> registry, float deltaTime) = 0;
00027
00028     private:
00029 };
00030
00031 } // namespace ecs
00032
00033 #endif /* !ASystem_HPP_ */
```

5.39 ISystem.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ISystem
00006 */
00007
00008 #ifndef ISystem_HPP_
00009 #define ISystem_HPP_
00010
00011 #include "../../resourceManager/ResourceManager.hpp"
00012 #include "../../entity/registry/ARegistry.hpp"
00013 #include <memory>
00014
00015 namespace ecs {
00016
00017 class ISystem {
00018     public:
00019         virtual ~ISystem() = default;
00020         virtual void updateSystem(std::shared_ptr<ResourceManager> resourceManager,
      std::shared_ptr<ARegistry> registry, float deltaTime) = 0;
00021 };
00022
00023 } // namespace ecs
00024
00025 #endif /* !ISystem_HPP_ */
```

5.40 MovementInputSystem.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MovementInputSystem
00006 */
00007
00008 #ifndef MOVEMENTINPUTSYSTEM_HPP_
00009 #define MOVEMENTINPUTSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../../component/temporary/MovementIntentComponent.hpp"
00013 #include <unordered_map>
00014
00015 namespace ecs {
00016
00017 enum class MovementKey {
00018     Left,
00019     Right,
00020     Up,
00021     Down
00022 };
00023
00024 class MovementInputSystem : public ASystem {
00025     public:
00026         MovementInputSystem();
00027         ~MovementInputSystem() = default;
00028
00029         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry>
      registry, float deltaTime) override;
00030
00031         // Simulation methods (will be replaced by real input system)
```

```
00032         void simulateKeyPress(MovementKey key, bool pressed);
00033         void simulateAxis(float horizontal, float vertical);
00034
00035     private:
00036         // Movement key states
00037         std::unordered_map<MovementKey, bool> _movementKeyStates;
00038         math::Vector2f _axisInput;
00039
00040         math::Vector2f getMovementDirection() const;
00041         void updateMovementIntent(std::shared_ptr<ARegistry> registry, int entityId, const
      math::Vector2f &direction);
00042 };
00043
00044 } // namespace ecs
00045
00046 #endif /* !MOVEMENTINPUTSYSTEM_HPP_ */
```

5.41 MovementSystem.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** MovementSystem
00006 */
00007
00008 #ifndef MOVEMENTSYSTEM_HPP_
00009 #define MOVEMENTSYSTEM_HPP_
00010
00011 #include "../base/ASystem.hpp"
00012 #include "../../component/base/IComponent.hpp"
00013 #include "../../component/temporary/MovementIntentComponent.hpp"
00014 #include "../../component/permanent/TransformComponent.hpp"
00015 #include "../../component/permanent/SpeedComponent.hpp"
00016 #include "../../../constants.hpp"
00017
00018 namespace ecs {
00019
00020 class MovementSystem : public ASystem {
00021     public:
00022         MovementSystem();
00023         ~MovementSystem() = default;
00024
00025         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry>
      registry, float deltaTime) override;
00026 };
00027
00028 }  // namespace ecs
00029
00030 #endif /* !MOVEMENTSYSTEM_HPP_ */
```

5.42 ASystemManager.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ASystemManager
00006 */
00007
00008 #ifndef ASystemMANAGER_HPP_
00009 #define ASystemMANAGER_HPP_
00010
00011 #include <vector>
00012 #include <memory>
00013
00014 #include "ISystemManager.hpp"
00015 #include "../../resourceManager/ResourceManager.hpp"
00016 #include "../../entity/registry/ARegistry.hpp"
00017 #include "../base/ISystem.hpp"
00018
00019 namespace ecs {
00020
00021 class ASystemManager : public ISystemManager {
00022     public:
00023         ASystemManager();
00024         ~ASystemManager();
```

```
00025         void updateAllSystems(std::shared_ptr<ResourceManager> resourceManager,
      std::shared_ptr<ARegistry> registry, float deltaTime) override;
00026         void addSystem(std::shared_ptr<ISystem> system) override;
00027         void removeSystem(std::shared_ptr<ISystem> system) override;
00028
00029     private:
00030         std::vector<std::shared_ptr<ISystem» _systems;
00031 };
00032
00033 } // namespace ecs
00034
00035 #endif /* !ASystemMANAGER_HPP_ */
```

5.43 ISystemManager.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ISystemManager
00006 */
00007
00008 #ifndef ISystemMANAGER_HPP_
00009 #define ISystemMANAGER_HPP_
00010
00011 #include <memory>
00012
00013 #include "../base/ISystem.hpp"
00014 #include "../../resourceManager/ResourceManager.hpp"
00015 #include "../../entity/registry/ARegistry.hpp"
00016
00017 namespace ecs {
00018
00019 class ISystemManager {
00020     public:
00021         virtual ~ISystemManager() = default;
00022         virtual void updateAllSystems(std::shared_ptr<ResourceManager> resourceManager,
      std::shared_ptr<ARegistry> registry, float deltaTime) = 0;
00023         virtual void addSystem(std::shared_ptr<ISystem> system) = 0;
00024         virtual void removeSystem(std::shared_ptr<ISystem> system) = 0;
00025 };
00026
00027 }  // namespace ecs
00028
00029 #endif /* !ISystemMANAGER_HPP_ */
```

5.44 View.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** View
00006 */
00007
00008 #ifndef VIEW_HPP_
00009 #define VIEW_HPP_
00010
00011 #include <vector>
00012 #include <memory>
00013 #include <type_traits>
00014
00015 namespace ecs {
00016
00017 template <typename... Components>
00018 class View {
00019     public:
00020         View(std::shared_ptr<class ARegistry> registry);
00021
00022         class Iterator;
00023
00024         Iterator begin();
00025         Iterator end();
00026
00027         class Iterator {
00028             public:
00029                 Iterator(std::shared_ptr<class ARegistry> registry, size_t entityId, size_t
      maxEntityId);
```

```
00030                 bool operator!=(const Iterator& other) const;
00031                 Iterator& operator++();
00032                 size_t operator*() const;
00033
00034             private:
00035                 bool hasAllComponents() const;
00036                 std::shared_ptr<class ARegistry> _registry;
00037                 size_t _entityId;
00038                 size_t _maxEntityId;
00039         };
00040
00041     private:
00042         std::shared_ptr<class ARegistry> _registry;
00043 };
00044
00045 template <typename... Components>
00046 class Group {
00047     public:
00048         Group(std::shared_ptr<class ARegistry> registry);
00049
00050         class Iterator;
00051
00052         Iterator begin();
00053         Iterator end();
00054
00055         class Iterator {
00056             public:
00057                 Iterator(std::shared_ptr<class ARegistry> registry, size_t entityId, size_t
      maxEntityId);
00058                 bool operator!=(const Iterator& other) const;
00059                 Iterator& operator++();
00060                 size_t operator*() const;
00061
00062             private:
00063                 bool hasAllComponents() const;
00064                 std::shared_ptr<class ARegistry> _registry;
00065                 size_t _entityId;
00066                 size_t _maxEntityId;
00067         };
00068
00069     private:
00070         std::shared_ptr<class ARegistry> _registry;
00071 };
00072
00073 }  // namespace ecs
00074
00075 #endif /* !VIEW_HPP_ */
```

5.45 AError.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AError
00006 */
00007
00008 #ifndef AERROR_HPP_
00009 #define AERROR_HPP_
00010
00011 #include <string>
00012 #include "IError.hpp"
00013
00014 namespace err {
00015
00016 class AError : public IError {
00017     public:
00018         AError(const std::string &message, int code = 0);
00019
00020         virtual ~AError() noexcept = default;
00021         const char *what() const noexcept override;
00022         int getCode() const noexcept;
00023         std::string getDetails() const noexcept;
00024
00025         virtual std::string getType() const noexcept = 0;
00026
00027     protected:
00028         std::string m_message;
00029         int m_code;
00030 };
00031
00032 }
00033
00034 #endif /* !AERROR_HPP_ */
```

5.46 IError.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IError
00006 */
00007
00008 #ifndef IERROR_HPP_
00009 #define IERROR_HPP_
00010
00011 #include <string>
00012 #include <exception>
00013
00014 namespace err {
00015
00016 class IError : public std::exception {
00017     public:
00018
00019         virtual ~IError() noexcept = default;
00020         virtual const char *what() const noexcept override = 0;
00021         virtual int getCode() const noexcept = 0;
00022         virtual std::string getType() const noexcept = 0;
00023         virtual std::string getDetails() const noexcept = 0;
00024
00025     protected:
00026 };
00027
00028 }
00029
00030 #endif /* !IERROR_HPP_ */
```

5.47 ServerErrror.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerErrror
00006 */
00007
00008
00009 #include "AError.hpp"
00010
00011 namespace err {
00012
00013 class ServerError : public AError {
00014     public:
00015         enum ErrorCode {
00016             UNKNOWN = 1000,
00017             CONNECTION_FAILED = 1001,
00018             TIMEOUT = 1002,
00019             INVALID_REQUEST = 1003,
00020             INTERNAL_ERROR = 1004,
00021             LIBRARY_LOAD_FAILED = 1005,
00022             CONFIG_ERROR = 1006
00023         };
00024
00025         ServerError(const std::string &message, ErrorCode code = UNKNOWN);
00026         virtual ~ServerError() noexcept = default;
00027         std::string getType() const noexcept override;
00028
00029     private:
00030 };
00031
00032 }
```

5.48 Vector2f.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Vector2f
00006 */
00007
```

```
00008 #ifndef VECTOR2F_HPP_
00009 #define VECTOR2F_HPP_
00010
00011 namespace math {
00012
00013 class Vector2f {
00014     public:
00015         Vector2f(float x = 0.0f, float y = 0.0f);
00016         Vector2f(Vector2f const &other);
00017         ~Vector2f() = default;
00018
00019         float getX() const;
00020         void setX(float x);
00021         float getY() const;
00022         void setY(float y);
00023
00024         Vector2f getVector() const;
00025         Vector2f operator*(float scalar) const;
00026         Vector2f operator-(Vector2f const &other) const;
00027         Vector2f operator+(Vector2f const &other) const;
00028         void operator=(Vector2f const &other);
00029         void operator+=(Vector2f const &other);
00030         void operator-=(Vector2f const &other);
00031         void operator*=(float scalar);
00032         void operator/=(float scalar);
00033     private:
00034         float _x;
00035         float _y;
00036 };
00037
00038 } // namespace math
00039
00040 #endif /* !VECTOR2F_HPP_ */
```

5.49 IServer.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef ISERVER_HPP_
00009     #define ISERVER_HPP_
00010
00011 #include <memory>
00012 #include <vector>
00013
00014 #include "ServerConfig.hpp"
00015 #include "../libs/Packet/IPacket.hpp"
00016 #include "../libs/Network/INetwork.hpp"
00017
00018 namespace rserv {
00019 class IServer {
00020 public:
00021     virtual ~IServer() = default;
00022
00023     virtual void init() = 0;
00024     virtual void start() = 0;
00025     virtual void stop() = 0;
00026
00027     virtual void setConfig(std::shared_ptr<ServerConfig> config) = 0;
00028     virtual std::shared_ptr<ServerConfig> getConfig() const = 0;
00029     virtual unsigned int getPort() const = 0;
00030     virtual void setPort(unsigned int port) = 0;
00031
00032     virtual int getState() const = 0;
00033     virtual void setState(int state) = 0;
00034
00035     virtual int getFd() const = 0;
00036     virtual void setFd(int fd) = 0;
00037     virtual operator int() const noexcept = 0;
00038
00039     virtual std::shared_ptr<net::INetwork> getNetwork() const = 0;
00040     virtual void setNetwork(std::shared_ptr<net::INetwork> network) = 0;
00041
00042     virtual void onClientConnected(int idClient) = 0;
00043     virtual void onClientDisconnected(int idClient) = 0;
00044     virtual void onPacketReceived(int idClient, const IPacket &packet) = 0;
00045
00046     virtual void processConnections() = 0;
00047     virtual void processIncomingPackets() = 0;
```

```
00048
00049     virtual void broadcastPacket(const IPacket &packet) = 0;
00050     virtual void sendToClient(int idClient, const IPacket &packet) = 0;
00051     virtual std::vector<int> getConnectedClients() const = 0;
00052     virtual int getClientCount() const = 0;
00053 };
00054 } // namespace rserv = r-type server
00055 #endif /* !ISERVER_HPP_ */
```

5.50 Server.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef SERVER_HPP_
00009     #define SERVER_HPP_
00010
00011 #include <memory>
00012 #include "IServer.hpp"
00013 #include "ServerConfig.hpp"
00014 #include "../libs/Network/INetwork.hpp"
00015 #include "../libs/Buffer/IBuffer.hpp"
00016 #include "../common/DLLoader/DLLoader.hpp"
00017
00018 typedef void *(*createNetworkLib_t)();
00019 typedef void *(*createBuffer_t)();
00020 typedef void *(*createPacket_t)();
00021
00022 #define pathLoad "./librairies"
00023 #define networkLib "libNetwork.so"
00024 #define bufferLib "libBuffer.so"
00025 #define packetLib "libPacket.so"
00026
00027 namespace rserv {
00028     class Server : public IServer {
00029         public:
00030             Server();
00031             ~Server();
00032
00033             void init() override;
00034             void start() override;
00035             void stop() override;
00036
00037             void setConfig(std::shared_ptr<ServerConfig> config) override;
00038             std::shared_ptr<ServerConfig> getConfig() const override;
00039             unsigned int getPort() const override;
00040             void setPort(unsigned int port) override;
00041
00042             int getState() const override;
00043             void setState(int state) override;
00044
00045             int getFd() const override;
00046             void setFd(int fd) override;
00047             operator int() const noexcept override;
00048
00049             std::shared_ptr<net::INetwork> getNetwork() const override;
00050             void setNetwork(std::shared_ptr<net::INetwork> network) override;
00051
00052             void onClientConnected(int idClient) override;
00053             void onClientDisconnected(int idClient) override;
00054             void onPacketReceived(int idClient, const IPacket &packet) override;
00055
00056             void processConnections() override;
00057             void processIncomingPackets() override;
00058
00059             void broadcastPacket(const IPacket &packet) override;
00060             void sendToClient(int idClient, const IPacket &packet) override;
00061             std::vector<int> getConnectedClients() const override;
00062             int getClientCount() const override;
00063
00064         private:
00065             void loadNetworkLibrary();
00066             void loadBufferLibrary();
00067             void loadPacketLibrary();
00068             DLLoader<createNetworkLib_t> _networloader;
00069             DLLoader<createBuffer_t> _bufferloader;
00070             DLLoader<createPacket_t> _packetloader;
00071
00072             std::shared_ptr<ServerConfig> _config;
```

```
00073             std::shared_ptr<net::INetwork> _network;
00074             std::shared_ptr<IBuffer> _buffer;
00075             std::shared_ptr<IPacket> _packet;
00076
00077     };
00078 } // namespace rserv = r-type server
00079
00080 #endif /* !SERVER_HPP_ */
```

5.51 ServerConfig.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** Header
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #include <cstdint>
00009
00010 #ifndef SERVER_CONFIG_HPP_
00011     #define SERVER_CONFIG_HPP_
00012
00013 namespace rserv {
00014     class ServerConfig {
00015         public:
00016             ServerConfig();
00017             ~ServerConfig();
00018
00019             int getState() const;
00020             int getFd() const;
00021
00022             void setPort(unsigned int port);
00023             unsigned int getPort() const;
00024
00025             void setState(int state);
00026             void setFd(int fd);
00027
00028             void setNbClients(int nbClients);
00029             int getNbClients() const;
00030
00031             uint32_t getIp() const;
00032             void setIp(uint32_t ip);
00033
00034         private:
00035             int _state;
00036             int _fd;
00037             unsigned int _port;
00038             int _nbClients;
00039             uint32_t _ip;
00040     };
00041 } // namespace rserv = r-type server
00042
00043 #endif /* !SERVER_CONFIG_HPP_ */
```

5.52 Utils.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Utils
00006 */
00007
00008 #include <vector>
00009 #include <memory>
00010
00011 #include "ServerConfig.hpp"
00012
00013 #ifndef UTILS_HPP_
00014 #define UTILS_HPP_
00015
00016 class Utils {
00017     public:
00018         Utils();
00019         ~Utils();
00020
00021         void helper();
```

```
00022         void parsCli(int ac, char **av, std::shared_ptr<rserv::ServerConfig> config);
00023     protected:
00024     private:
00025 };
00026
00027 #endif /* !UTILS_HPP_ */
```

Index