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

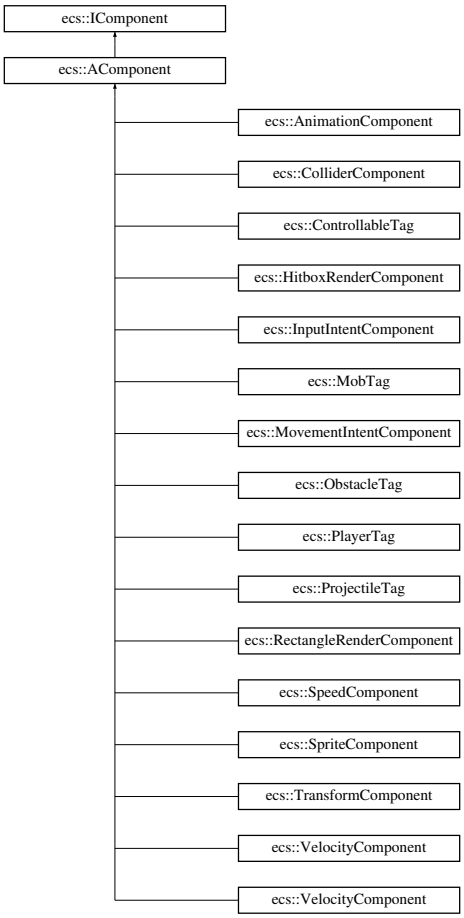
Here is a list of all documented files with brief descriptions:

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]
```

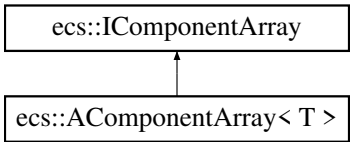
Implements ecs::IComponent.

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

Private Attributes

- std::vector< 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]
```

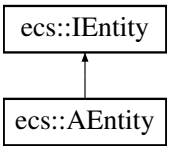
Implements ecs::IComponentArray.

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]
```

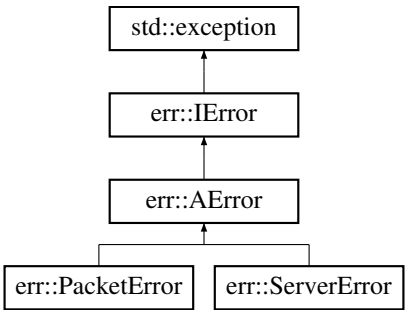
Implements ecs::IEntity.

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 override
- std::string getDetails () const noexcept override
- virtual std::string getType () const noexcept override=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  [override], [virtual], [noexcept]
```

Implements err::IError.

4.4.1.2 getDetails()

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

Implements err::IError.

4.4.1.3 getType()

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

Implements err::IError.

4.4.1.4 what()

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

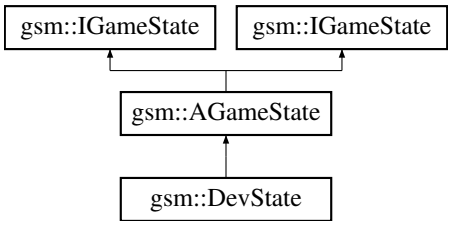
Implements err::IError.

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]
```

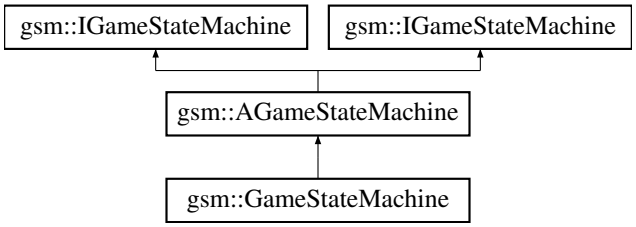
Implements gsm::IGameState.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/base/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/base/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]
```

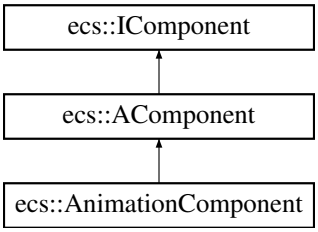
Implements gsm::IGameStateMachine.

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::AnimationComponent Class Reference

Inheritance diagram for ecs::AnimationComponent:



Public Member Functions

- **AnimationComponent** (const std::string &texturePath, float frameWidth, float frameHeight, int frameCount, float startWidth, float startHeight, float speed=0.1f)
- const math::FRect & **getFrameRect** () const
- void **setFrameRect** (const math::FRect &rect)
- float **getFrameWidth** () const
- float **getFrameHeight** () const
- int **getFrameCount** () const
- int **getCurrentFrame** () const
- void **setCurrentFrame** (int frame)
- float **getAnimationSpeed** () const
- void **setAnimationSpeed** (float speed)
- math::Chrono & **getChrono** ()
- const math::Chrono & **getChrono** () const
- bool **isValid** () const
- const std::string & **getTexturePath** () const
- float **getStartWidth** () const
- void **setStartWidth** (float startWidth)
- float **getStartHeight** () const
- void **setStartHeight** (float startHeight)

Public Member Functions inherited from ecs::AComponent

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

Private Attributes

- std::string **_texturePath**
- math::FRect **_frameRect**
- int **_frameCount**
- int **_currentFrame**
- float **_animationSpeed**
- math::Chrono **_chrono**
- float **_startHeight**
- float **_startWidth**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

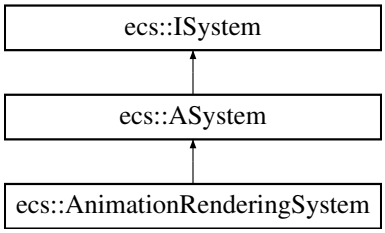
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/rendering/AnimationComponent.↩ hpp

4.8 ecs::AnimationRenderingSystem Class Reference

Inheritance diagram for ecs::AnimationRenderingSystem:



Protected Member Functions

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

Additional Inherited Members

Public Member Functions inherited from ecs::ASystem

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

4.8.1 Member Function Documentation

4.8.1.1 update()

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

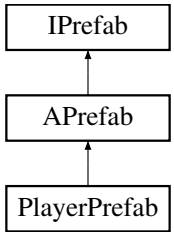
Implements ecs::ASystem.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/AnimationRendering↩ System.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/AnimationRendering↩ System.cpp

4.9 APrefab Class Reference

Inheritance diagram for APrefab:



Public Member Functions

- size_t instantiate (const std::shared_ptr< ecs::ARegistry > &registry) override

4.9.1 Member Function Documentation

4.9.1.1 instantiate()

```
size_t APrefab::instantiate (
            const std::shared_ptr< ecs::ARegistry > & registry)  [override], [virtual]
```

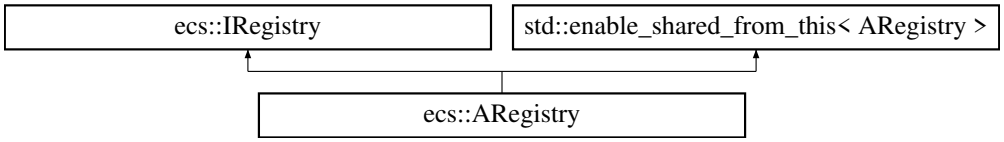
Implements IPrefab.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/APrefab.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/APrefab.cpp

4.10 ecs::ARegistry Class Reference

Inheritance diagram for ecs::ARegistry:



Public Member Functions

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

Public Member Functions inherited from ecs::IRegistry

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

Private Attributes

- size_t **_nextEntityId**
- std::unordered_map< std::string, std::shared_ptr< IComponentArray > > **_components**

4.10.1 Member Function Documentation

4.10.1.1 createEntity()

```
size_t ecs::ARegistry::createEntity ()  [override], [virtual]
```

Implements ecs::IRegistry.

4.10.1.2 removeAllComponentsWithState()

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

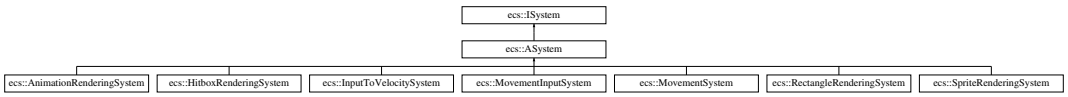
Implements ecs::IRegistry.

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

4.11 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

4.11.1 Member Function Documentation

4.11.1.1 updateSystem()

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

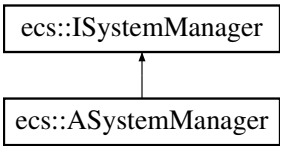
Implements ecs::ISystem.

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

4.12 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**

4.12.1 Member Function Documentation

4.12.1.1 addSystem()

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

Implements ecs::ISystemManager.

4.12.1.2 removeSystem()

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

Implements ecs::ISystemManager.

4.12.1.3 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

4.13 math::Chrono Class Reference

Public Member Functions

- void **start** ()
- void **stop** ()
- void **reset** ()
- float **getElapsedSeconds** () const
- float **getElapsedMilliseconds** () const
- bool **isRunning** () const

Private Attributes

- std::chrono::high_resolution_clock::time_point **_startTime**
- std::chrono::high_resolution_clock::time_point **_stopTime**
- bool **_isRunning**

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

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

4.14 ClientNetwork Class Reference

Public Member Functions

- void **init** ()
- void **start** ()
- void **stop** ()
- int **getPort** () const
- void **setPort** (int port)
- uint32_t **getIp** () const
- void **setIp** (uint32_t ip)
- void **sendData** (const IPacketManager &data, size_t size)
- IPacketManager & **receiveData** (const IBuffer &buffer, size_t size) const
- void **loadNetworkLibrary** ()
- void **loadBufferLibrary** ()
- void **loadPacketLibrary** ()

Private Attributes

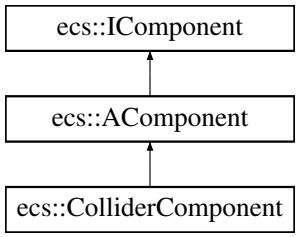
- DLLoader< createNetworkLib_t > **_networloader**
- DLLoader< createBuffer_t > **_bufferloader**
- DLLoader< createPacket_t > **_packetloader**
- std::shared_ptr< net::INetwork > **_network**
- std::shared_ptr< IBuffer > **_buffer**
- std::shared_ptr< IPacketManager > **_packet**
- int **_port**
- uint32_t **_ip**

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

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

4.15 ecs::ColliderComponent Class Reference

Inheritance diagram for ecs::ColliderComponent:



Public Member Functions

- **ColliderComponent** (math::Vector2f offset=math::Vector2f(0.0f, 0.0f), math::Vector2f size=math::Vector2f(0.↩ 0f, 0.0f), CollisionType type=CollisionType::Solid)
- math::Vector2f **getOffset** () const
- void **setOffset** (math::Vector2f offset)
- math::Vector2f **getSize** () const
- void **setSize** (math::Vector2f size)
- CollisionType **getType** () const
- void **setType** (CollisionType type)
- math::FRect **getHitbox** (math::Vector2f entityPosition) const

Public Member Functions inherited from ecs::AComponent

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

Private Attributes

- math::Vector2f **_offset**
- math::Vector2f **_size**
- CollisionType **_type**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

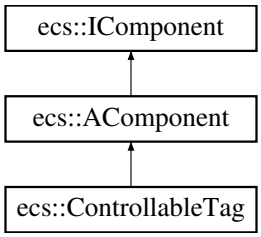
- ComponentState **_state** = Permanent

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

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

4.16 ecs::ControllableTag Class Reference

Inheritance diagram for ecs::ControllableTag:



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/ControllableTag.hpp

4.17 Core Class Reference

Public Member Functions

- void **run** ()
- std::shared_ptr< ClientNetwork > **getNetwork** ()
- void **init** ()
- void **loop** ()
- std::shared_ptr< rserv::ServerConfig > **getConfig** () const
- std::shared_ptr< rserv::Server > **getServer** () const
- std::shared_ptr< ecs::ResourceManager > **getResourceManager** () const

Private Member Functions

- void **initNetwork** ()
- void **initLibraries** ()
- void **networkLoop** ()

Private Attributes

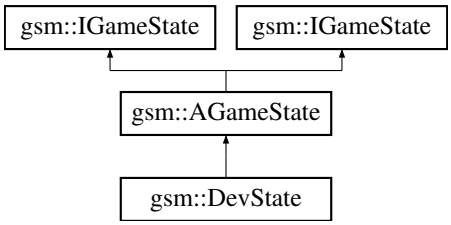
- std::shared_ptr< DLLoader< gfx::createWindow_t > > **_windowLoader**
- std::shared_ptr< DLLoader< gfx::createEvent_t > > **_eventLoader**
- std::shared_ptr< ecs::ResourceManager > **_resourceManager**
- std::shared_ptr< gsm::GameStateMachine > **_gsm**
- std::shared_ptr< ClientNetwork > **_clientNetwork**
- std::thread **_networkThread**
- std::shared_ptr< Utils > **_utils**
- std::shared_ptr< rserv::ServerConfig > **_config**
- std::shared_ptr< rserv::Server > **_server**

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

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

4.18 gsm::DevState Class Reference

Inheritance diagram for gsm::DevState:



Public Member Functions

- **DevState** (std::shared_ptr< IGameStateMachine > gsm, std::shared_ptr< ecs::ResourceManager > resourceManager)
- void enter () override
- void update (float deltaTime) override
- void render () override
- void exit () override

Public Member Functions inherited from gsm::AGameState

- **AGameState** (std::shared_ptr< IGameStateMachine > gsm)
- **AGameState** (std::shared_ptr< IGameStateMachine > gsm)

Private Attributes

- std::shared_ptr< ecs::ResourceManager > **_resourceManager**
- std::shared_ptr< ecs::ARegistry > **_registry**
- std::shared_ptr< ecs::ASystemManager > **_systemManager**
- std::shared_ptr< ecs::MovementSystem > **_movementSystem**
- std::shared_ptr< ecs::InputToVelocitySystem > **_inputToVelocitySystem**
- std::shared_ptr< ecs::MovementInputSystem > **_inputSystem**
- std::shared_ptr< ecs::SpriteRenderingSystem > **_spriteRenderingSystem**
- std::shared_ptr< EntityPrefabManager > **_prefabManager**

Additional Inherited Members

Protected Attributes inherited from gsm::AGameState

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

4.18.1 Member Function Documentation

4.18.1.1 enter()

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

Reimplemented from gsm::AGameState.

4.18.1.2 exit()

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

Reimplemented from gsm::AGameState.

4.18.1.3 render()

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

Reimplemented from gsm::AGameState.

4.18.1.4 update()

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

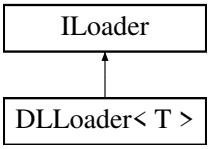
Reimplemented from gsm::AGameState.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/DevState.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/gsm/states/scenes/DevState.cpp

4.19 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

4.19.1 Member Function Documentation

4.19.1.1 Close()

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

Implements ILoader.

4.19.1.2 Error()

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

Implements ILoader.

4.19.1.3 getHandler()

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

Implements ILoader.

4.19.1.4 Open()

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

Implements ILoader.

4.19.1.5 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

4.20 EntityPrefabManager Class Reference

Public Member Functions

- void **registerPrefab** (const std::string &name, const std::shared_ptr< IPrefab > &prefab)
- std::shared_ptr< IPrefab > **getPrefab** (const std::string &name) const
- size_t **createEntityFromPrefab** (const std::string &prefabName, const std::shared_ptr< ecs::ARegistry > &registry)
- bool **hasPrefab** (const std::string &name) const
- void **deletePrefab** (const std::string &name)
- void **clearPrefabs** ()

Private Attributes

- std::map< std::string, std::shared_ptr< IPrefab > > **_prefabs**

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/Prefab/entityPrefabManager/EntityPrefabManager.cpp

4.21 math::FRect Class Reference

Public Member Functions

- **FRect** (float left, float top, float width, float height)
- **FRect** (FRect const &other)
- float **getLeft** () const
- void **setLeft** (float left)
- float **getTop** () const
- void **setTop** (float top)
- float **getWidth** () const
- void **setWidth** (float width)
- float **getHeight** () const
- void **setHeight** (float height)
- bool **contains** (float x, float y) const
- bool **intersects** (FRect const &other) const
- bool **intersects** (FRect const &other, FRect &intersection) const
- FRect & **operator=** (FRect const &other)
- bool **operator==** (FRect const &other) const
- bool **operator!=** (FRect const &other) const

Private Attributes

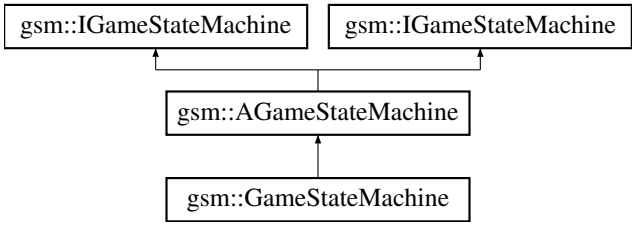
- float **left**
- float **top**
- float **width**
- float **height**

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

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

4.22 gsm::GameStateMachine Class Reference

Inheritance diagram for gsm::GameStateMachine:



Additional Inherited Members

Public Member Functions inherited from gsm::AGameStateMachine

- 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 inherited from gsm::AGameStateMachine

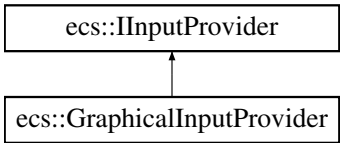
- std::stack< std::shared_ptr< IGameState > > **_states**

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

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

4.23 ecs::GraphicalInputProvider Class Reference

Inheritance diagram for ecs::GraphicalInputProvider:



Public Member Functions

- **GraphicalInputProvider** (std::shared_ptr< gfx::IEvent > eventSystem)
- bool isKeyPressed (event_t key) override
- float getAxisValue (event_t axis) override
- std::pair< int, int > getMousePos () override
- bool isMouseButtonPressed (int button) override

Private Attributes

- std::shared_ptr< gfx::IEvent > **_eventSystem**

Additional Inherited Members

Public Types inherited from ecs::IInputProvider

- using **event_t** = gfx::EventType

4.23.1 Member Function Documentation

4.23.1.1 getAxisValue()

```
float ecs::GraphicalInputProvider::getAxisValue (
            event_t axis)  [override], [virtual]
```

Implements ecs::IInputProvider.

4.23.1.2 getMousePos()

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

Implements ecs::IInputProvider.

4.23.1.3 isKeyPressed()

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

Implements ecs::IInputProvider.

4.23.1.4 isMouseButtonPressed()

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

Implements ecs::IInputProvider.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/client/initRessourcesManager/GraphicalInputProvider.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/initRessourcesManager/GraphicalInputProvider.cpp

4.24 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

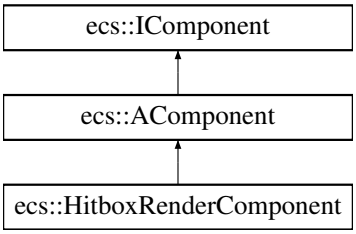
- std::shared_ptr< class ARegistry > **_registry**

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

4.25 ecs::HitboxRenderComponent Class Reference

Inheritance diagram for ecs::HitboxRenderComponent:



Public Member Functions

- **HitboxRenderComponent** (gfx::color_t color, float outlineThickness=1.0f)
- const gfx::color_t & **getColor** () const
- void **setColor** (const gfx::color_t &color)
- float **getOutlineThickness** () const
- void **setOutlineThickness** (float thickness)

Public Member Functions inherited from ecs::AComponent

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

Private Attributes

- gfx::color_t **_color**
- float **_outlineThickness**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

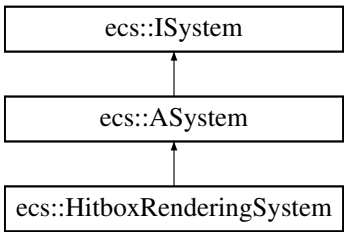
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/rendering/HitboxRenderComponent.↩ hpp

4.26 ecs::HitboxRenderingSystem Class Reference

Inheritance diagram for ecs::HitboxRenderingSystem:



Protected Member Functions

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

Additional Inherited Members

Public Member Functions inherited from ecs::ASystem

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

4.26.1 Member Function Documentation

4.26.1.1 update()

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

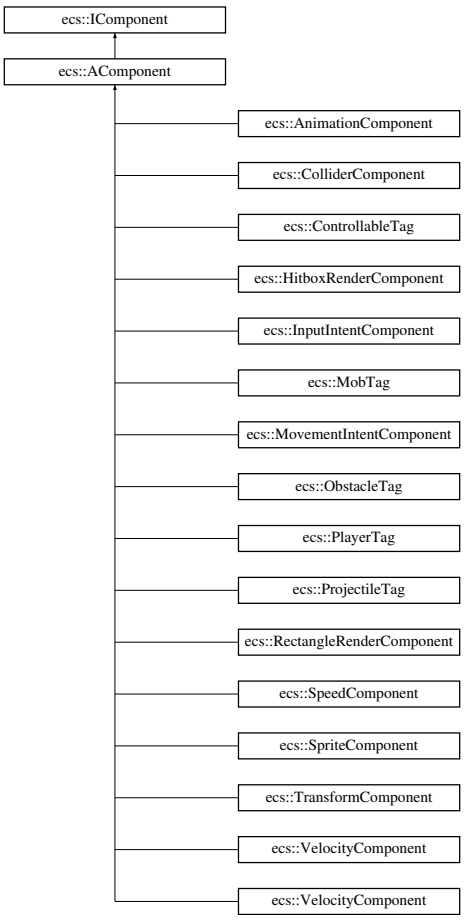
Implements ecs::ASystem.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/HitboxRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/HitboxRenderingSystem.cpp

4.27 ecs::IComponent Class Reference

Inheritance diagram for ecs::IComponent:



Public Member Functions

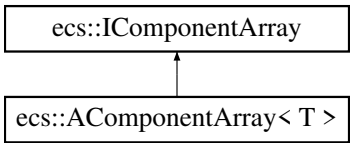
- virtual ComponentState **getState** () const =0
- virtual void **setState** (ComponentState newState)=0

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

4.28 ecs::IComponentArray Class Reference

Inheritance diagram for ecs::IComponentArray:



Public Member Functions

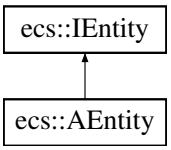
- virtual void **removeAllComponentsWithState** (ComponentState state)=0
- virtual size_t **getMaxEntityId** () const =0

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

4.29 ecs::IEntity Class Reference

Inheritance diagram for ecs::IEntity:



Public Member Functions

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

Private Member Functions

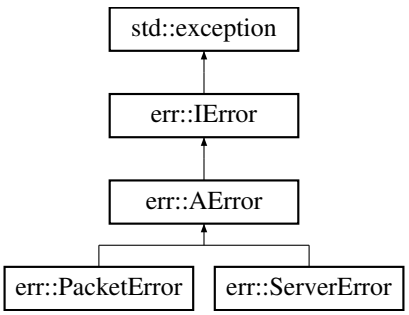
- **IEntity** (size_t id)

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

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

4.30 err::IError Class Reference

Inheritance diagram for err::IError:



Public Member Functions

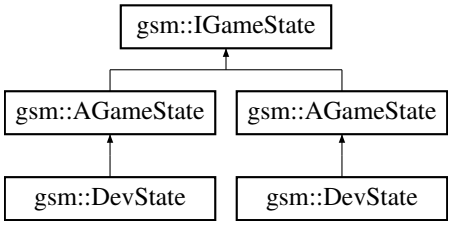
- virtual const char * **what** () const noexcept override=0
- virtual int **getCode** () const noexcept=0
- virtual std::string **getType** () const noexcept=0
- virtual std::string **getDetails** () const noexcept=0

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

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

4.31 gsm::IGameState Class Reference

Inheritance diagram for gsm::IGameState:



Public Member Functions

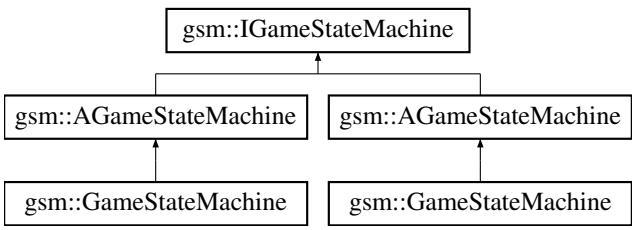
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **exit** ()=0
- virtual void **enter** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **exit** ()=0

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

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

4.32 gsm::IGameStateMachine Class Reference

Inheritance diagram for gsm::IGameStateMachine:



Public Member Functions

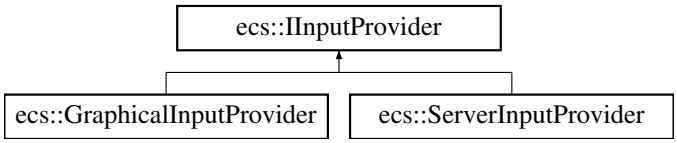
- virtual void **changeState** (std::shared_ptr< IGameState > newState)=0
- virtual void **pushState** (std::shared_ptr< IGameState > newState)=0
- virtual void **popState** ()=0
- virtual void **update** (float deltaTime)=0
- virtual void **render** ()=0
- virtual void **changeState** (std::shared_ptr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

4.33 ecs::IInputProvider Class Reference

Inheritance diagram for ecs::IInputProvider:



Public Types

- using **event_t** = gfx::EventType

Public Member Functions

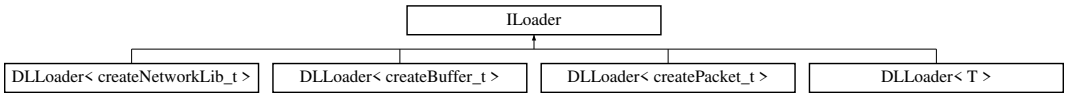
- virtual bool **isKeyPressed** (event_t key)=0
- virtual float **getAxisValue** (event_t axis)=0
- virtual std::pair< int, int > **getMousePos** ()=0
- virtual bool **isMouseButtonPressed** (int button)=0

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

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

4.34 ILoader Class Reference

Inheritance diagram for ILoader:



Public Member Functions

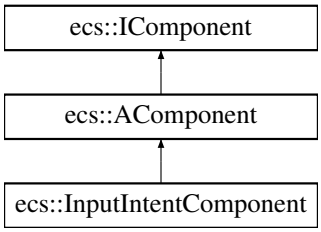
- virtual void * **Open** (const char *path, int flag)=0
- virtual void * **Symbol** (const char *symbolName)=0
- virtual int **Close** ()=0
- virtual const char * **Error** ()=0
- virtual void * **getHandler** () const =0

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

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

4.35 ecs::InputIntentComponent Class Reference

Inheritance diagram for ecs::InputIntentComponent:



Public Member Functions

- **InputIntentComponent** (const math::Vector2f &direction=math::Vector2f(0.0f, 0.0f))
- math::Vector2f **getDirection** () const
- void **setDirection** (const math::Vector2f &direction)

Public Member Functions inherited from ecs::AComponent

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

Private Attributes

- math::Vector2f **_direction**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

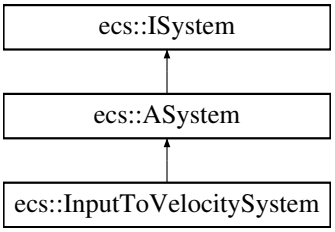
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/temporary/InputIntentComponent.↩
hpp

4.36 ecs::InputToVelocitySystem Class Reference

Inheritance diagram for ecs::InputToVelocitySystem:



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

4.36.1 Member Function Documentation

4.36.1.1 update()

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

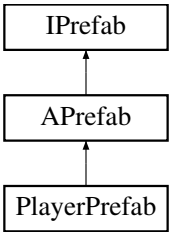
Implements ecs::ASystem.

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

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

4.37 IPrefab Class Reference

Inheritance diagram for IPrefab:



Public Member Functions

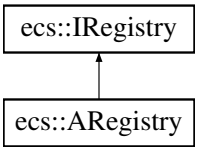
- virtual size_t **instantiate** (const std::shared_ptr< ecs::ARegistry > &registry)=0

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

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

4.38 ecs::IRegistry Class Reference

Inheritance diagram for ecs::IRegistry:



Public Member Functions

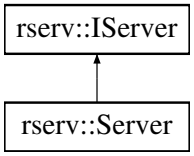
- template<typename T>
  void **registerComponent** ()
- template<typename T>
  void **addComponent** (size_t entityId, std::shared_ptr< T > component)
- template<typename T>
  std::shared_ptr< T > **getComponent** (size_t entityId)
- template<typename T>
  void **removeComponent** (size_t entityId)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

4.39 rserv::IServer Class Reference

Inheritance diagram for rserv::IServer:



Public Member Functions

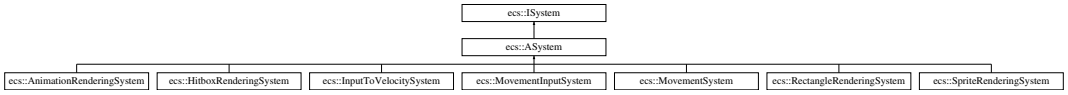
- virtual void **init** ()=0
- virtual void **start** ()=0
- virtual void **stop** ()=0
- virtual void **setConfig** (std::shared_ptr< ServerConfig > config)=0
- virtual std::shared_ptr< ServerConfig > **getConfig** () const =0
- virtual unsigned int **getPort** () const =0
- virtual void **setPort** (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 IPacketManager &packet)=0
- virtual void **processConnections** ()=0
- virtual void **processIncomingPackets** ()=0
- virtual void **broadcastPacket** ()=0
- virtual void **sendToClient** (int idClient)=0
- virtual std::vector< int > **getConnectedClients** () const =0
- virtual size_t **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

4.40 ecs::ISystem Class Reference

Inheritance diagram for ecs::ISystem:



Public Member Functions

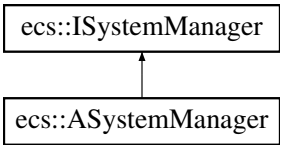
- virtual void **updateSystem** (std::shared_ptr< ResourceManager > resourceManager, std::shared_ptr< ARegistry > registry, float deltaTime)=0

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

4.41 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

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

Public Member Functions

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

Private Member Functions

- bool **hasAllComponents** () const

Private Attributes

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

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

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

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

Public Member Functions

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

Private Member Functions

- bool **hasAllComponents** () const

Private Attributes

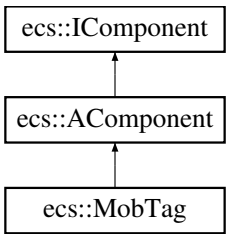
- std::shared_ptr< class ARegistry > **_registry**
- size_t **_entityId**
- size_t **_maxEntityId**

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

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

4.44 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

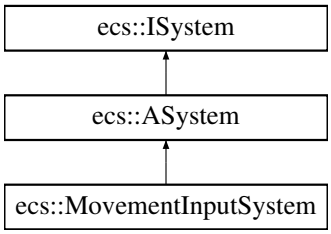
- ComponentState **_state** = Permanent

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

4.45 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

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** (std::shared_ptr< ResourceManager > resourceManager) const
- void **updateInputIntent** (std::shared_ptr< ARegistry > registry, size_t entityId, const math::Vector2f &direction)
- math::Vector2f **getAnalogStickInput** (std::shared_ptr< IInputProvider > inputProvider) const

4.45.1 Member Function Documentation

4.45.1.1 update()

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

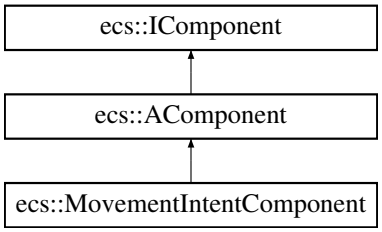
Implements ecs::ASystem.

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

4.46 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

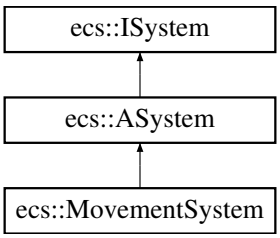
- ComponentState **_state** = Permanent

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

4.47 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

Private Member Functions

- bool **checkCollision** (std::shared_ptr< ARegistry > registry, size_t entityId, math::Vector2f newPos)
- math::Vector2f **calculateSmoothMovement** (std::shared_ptr< ARegistry > registry, size_t entityId, math::Vector2f startPos, math::Vector2f desiredPos)
- math::Vector2f **calculateSlidingMovement** (std::shared_ptr< ARegistry > registry, size_t entityId, math::Vector2f basePos, math::Vector2f desiredPos)
- math::Vector2f **calculateSmoothSlidingPosition** (std::shared_ptr< ARegistry > registry, size_t entityId, math::Vector2f startPos, math::Vector2f desiredPos)
- math::Vector2f **handleBounceCollision** (std::shared_ptr< ARegistry > registry, size_t entityId, math::Vector2f startPos, math::Vector2f desiredPos, std::shared_ptr< ecs::VelocityComponent > velocityComp)

4.47.1 Member Function Documentation

4.47.1.1 update()

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

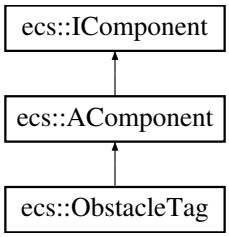
Implements ecs::ASystem.

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

4.48 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

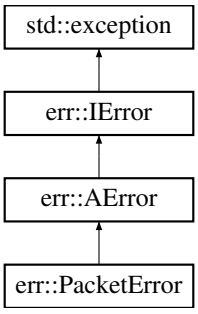
- ComponentState **_state** = Permanent

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

4.49 err::PacketError Class Reference

Inheritance diagram for err::PacketError:



Public Types

- enum **ErrorCode** { **UNKNOWN** = 1000 , **SERIALIZER_ATTRIBUTION_FAILED** = 1001 }

Public Member Functions

- **PacketError** (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 override
- std::string getDetails () const noexcept override

Additional Inherited Members

Protected Attributes inherited from err::AError

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

4.49.1 Member Function Documentation

4.49.1.1 getType()

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

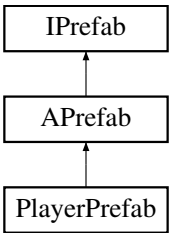
Implements err::AError.

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

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

4.50 PlayerPrefab Class Reference

Inheritance diagram for PlayerPrefab:



Public Member Functions

- **PlayerPrefab** (float x, float y, float scale, const std::string &animationPath, float frameWidth, float frame↩ Height, float startWidth, float startHeight, int frameCount)
- size_t instantiate (const std::shared_ptr< ecs::ARegistry > &registry) override

Private Attributes

- float **_x**
- float **_y**
- float **_scale**
- std::string **_animationPath**
- float **_frameWidth**
- float **_frameHeight**
- float **_startWidth**
- float **_startHeight**
- int **_frameCount**

4.50.1 Member Function Documentation

4.50.1.1 instantiate()

```
size_t PlayerPrefab::instantiate (
            const std::shared_ptr< ecs::ARegistry > & registry)  [inline], [override], [virtual]
```

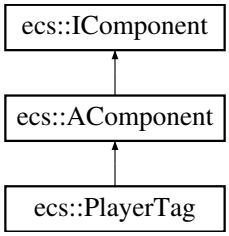
Reimplemented from APrefab.

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

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

4.51 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

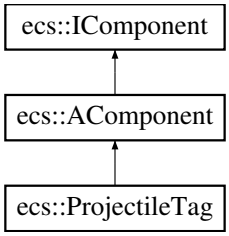
- ComponentState **_state** = Permanent

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

4.52 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

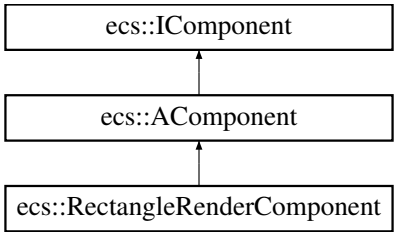
- ComponentState **_state** = Permanent

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

4.53 ecs::RectangleRenderComponent Class Reference

Inheritance diagram for ecs::RectangleRenderComponent:



Public Member Functions

- **RectangleRenderComponent** (gfx::color_t color, float width, float height)
- const gfx::color_t & **getColor** () const
- void **setColor** (const gfx::color_t &color)
- float **getWidth** () const
- float **getHeight** () const
- void **setSize** (float width, float height)

Public Member Functions inherited from ecs::AComponent

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

Private Attributes

- gfx::color_t **_color**
- std::pair< float, float > **_size**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

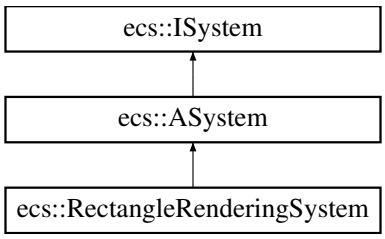
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/rendering/RectangleRender↩ Component.hpp

4.54 ecs::RectangleRenderingSystem Class Reference

Inheritance diagram for ecs::RectangleRenderingSystem:



Protected Member Functions

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

Additional Inherited Members

Public Member Functions inherited from ecs::ASystem

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

4.54.1 Member Function Documentation

4.54.1.1 update()

```
void ecs::RectangleRenderingSystem::update (
            std::shared_ptr< ResourceManager > resourceManager,
            std::shared_ptr< ARegistry > registry,
            float deltaTime)  [override], [protected], [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/rendering/RectangleRendering↩ System.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/RectangleRendering↩ System.cpp

4.55 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

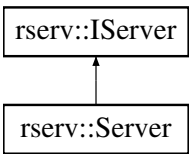
- std::unordered_map< size_t, std::shared_ptr< void > > **resources**

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

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

4.56 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 IPacketManager &packet) override
- void processConnections () override
- void processIncomingPackets () override
- void broadcastPacket () override
- void sendToClient (int idClient) override
- std::vector< int > getConnectedClients () const override
- size_t 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< IPacketManager > **_packet**

4.56.1 Member Function Documentation

4.56.1.1 broadcastPacket()

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

Implements rserv::IServer.

4.56.1.2 getClientCount()

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

Implements rserv::IServer.

4.56.1.3 getConfig()

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

Implements rserv::IServer.

4.56.1.4 getConnectedClients()

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

Implements rserv::IServer.

4.56.1.5 getFd()

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

Implements rserv::IServer.

4.56.1.6 getNetwork()

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

Implements rserv::IServer.

4.56.1.7 getPort()

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

Implements rserv::IServer.

4.56.1.8 getState()

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

Implements rserv::IServer.

4.56.1.9 init()

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

Implements rserv::IServer.

4.56.1.10 onClientConnected()

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

Implements rserv::IServer.

4.56.1.11 onClientDisconnected()

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

Implements rserv::IServer.

4.56.1.12 onPacketReceived()

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

Implements rserv::IServer.

4.56.1.13 operator int()

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

Implements rserv::IServer.

4.56.1.14 processConnections()

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

Implements rserv::IServer.

4.56.1.15 processIncomingPackets()

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

Implements rserv::IServer.

4.56.1.16 sendToClient()

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

Implements rserv::IServer.

4.56.1.17 setConfig()

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

Implements rserv::IServer.

4.56.1.18 setFd()

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

Implements rserv::IServer.

4.56.1.19 setNetwork()

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

Implements rserv::IServer.

4.56.1.20 setPort()

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

Implements rserv::IServer.

4.56.1.21 setState()

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

Implements rserv::IServer.

4.56.1.22 start()

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

Implements rserv::IServer.

4.56.1.23 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

4.57 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

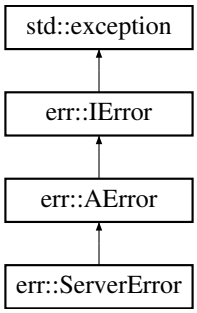
- int **_state**
- int **_fd**
- unsigned int **_port**
- int **_nbClients**
- uint32_t **_ip**

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

4.58 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 override
- std::string getDetails () const noexcept override

Additional Inherited Members

Protected Attributes inherited from err::AError

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

4.58.1 Member Function Documentation

4.58.1.1 getType()

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

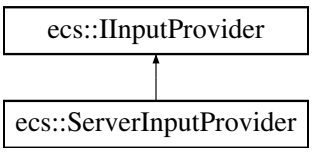
Implements err::AError.

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

4.59 ecs::ServerInputProvider Class Reference

Inheritance diagram for ecs::ServerInputProvider:



Public Member Functions

- bool isKeyPressed (event_t key) override
- float getAxisValue (event_t axis) override
- std::pair< int, int > getMousePos () override
- bool isMouseButtonPressed (int button) override

Additional Inherited Members

Public Types inherited from ecs::IInputProvider

- using **event_t** = gfx::EventType

4.59.1 Member Function Documentation

4.59.1.1 getAxisValue()

```
float ecs::ServerInputProvider::getAxisValue (
            event_t axis)  [override], [virtual]
```

Implements ecs::IInputProvider.

4.59.1.2 getMousePos()

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

Implements ecs::IInputProvider.

4.59.1.3 isKeyPressed()

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

Implements ecs::IInputProvider.

4.59.1.4 isMouseButtonPressed()

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

Implements ecs::IInputProvider.

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

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

4.60 Signal Class Reference

Static Public Member Functions

- static void **signalHandler** (int signum)
- static void **setupSignalHandlers** ()

Static Public Attributes

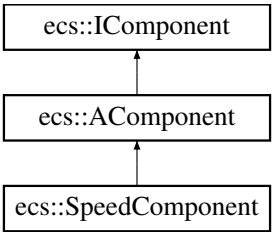
- static volatile sig_atomic_t **stopFlag** = 0

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

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

4.61 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

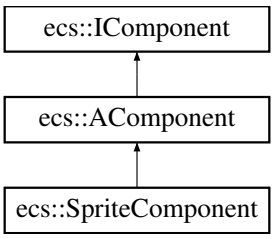
- ComponentState **_state** = Permanent

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

4.62 ecs::SpriteComponent Class Reference

Inheritance diagram for ecs::SpriteComponent:



Public Member Functions

- **SpriteComponent** (const std::string &texturePath)
- const std::string & **getTexturePath** () const
- void **setTexturePath** (const std::string &path)
- bool **isValid** () const

Public Member Functions inherited from ecs::AComponent

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

Private Attributes

- std::string **_texturePath**

Additional Inherited Members

Protected Attributes inherited from ecs::AComponent

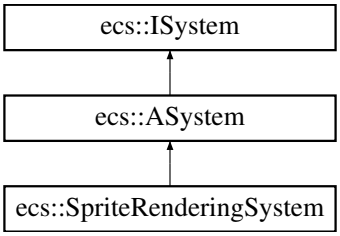
- ComponentState **_state** = Permanent

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/component/rendering/SpriteComponent.hpp

4.63 ecs::SpriteRenderingSystem Class Reference

Inheritance diagram for ecs::SpriteRenderingSystem:



Protected Member Functions

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

Additional Inherited Members

Public Member Functions inherited from ecs::ASystem

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

4.63.1 Member Function Documentation

4.63.1.1 update()

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

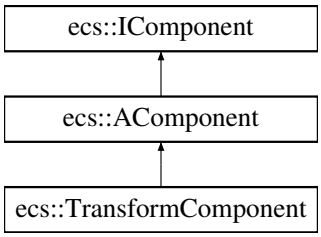
Implements ecs::ASystem.

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

- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/SpriteRenderingSystem.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/common/ECS/system/rendering/SpriteRenderingSystem.cpp

4.64 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

4.65 Utils Class Reference

Public Member Functions

- void **helper** ()
- void **parseCli** (int ac, char **av, std::shared_ptr< ClientNetwork > clientNetwork)
- 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/client/Utils.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Utils.hpp
- /home/albane/epitech/tech3/r-type/ryanR-type/client/Utils.cpp
- /home/albane/epitech/tech3/r-type/ryanR-type/server/Utils.cpp

4.66 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

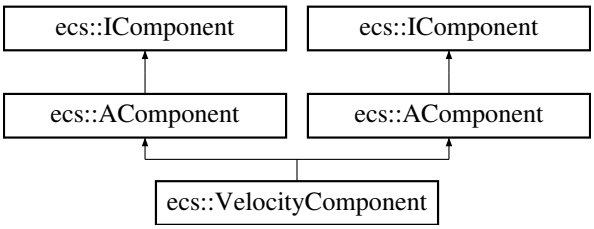
- float **_x**
- float **_y**

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

4.67 ecs::VelocityComponent Class Reference

Inheritance diagram for ecs::VelocityComponent:



Public Member Functions

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

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

4.68 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 ClientNetwork.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ClientNetwork
00006 */
00007
00008 #include <memory>
00009
00010 #include "../common/DLLoader/DLLoader.hpp"
00011 #include "../common/DLLoader/LoaderType.hpp"
00012 #include "../libs/Network/INetwork.hpp"
00013
00014 #ifndef CLIENTNETWORK_HPP_
00015 #define CLIENTNETWORK_HPP_
00016
00017 class ClientNetwork {
00018     public:
00019         ClientNetwork();
00020         ~ClientNetwork();
00021
00022         void init();
00023         void start();
00024         void stop();
00025
00026         int getPort() const;
00027         void setPort(int port);
00028
00029         uint32_t getIp() const;
00030         void setIp(uint32_t ip);
00031
00032         void sendData(const IPacketManager &data, size_t size);
00033         IPacketManager &receiveData(const IBuffer &buffer, size_t size) const;
00034
00035         void loadNetworkLibrary();
00036         void loadBufferLibrary();
00037         void loadPacketLibrary();
00038     protected:
00039     private:
00040         DLLoader<createNetworkLib_t> _networloader;
00041         DLLoader<createBuffer_t> _bufferloader;
00042         DLLoader<createPacket_t> _packetloader;
00043
00044         std::shared_ptr<net::INetwork> _network;
00045         std::shared_ptr<IBuffer> _buffer;
00046         std::shared_ptr<IPacketManager> _packet;
00047
00048         int _port;
00049         uint32_t _ip;
00050 };
00051
00052 #endif /* !CLIENTNETWORK_HPP_ */
```

5.2 Core.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Core.hpp
00006 */
00007
00008 #ifndef CORE_HPP_
00009 #define CORE_HPP_
00010
00011 #include <memory>
00012 #include <thread>
00013 #include "../common/ECS/resourceManager/ResourceManager.hpp"
00014 #include "ClientNetwork.hpp"
00015 #include "../libs/Multimedia/IWindow.hpp"
00016 #include "../libs/Multimedia/IEvent.hpp"
00017 #include "gsm/machine/GameStateMachine.hpp"
00018 #include "../common/DLLoader/DLLoader.hpp"
00019
00020 class Core
00021 {
00022     public:
00023         Core();
00024         ~Core();
00025
00026         void run();
00027
00028         std::shared_ptr<ClientNetwork> getNetwork();
00029
00030     private:
00031         std::shared_ptr<DLLoader<gfx::createWindow_t» _windowLoader;
00032         std::shared_ptr<DLLoader<gfx::createEvent_t» _eventLoader;
00033
00034         std::shared_ptr<ecs::ResourceManager> _resourceManager;
00035         std::shared_ptr<gsm::GameStateMachine> _gsm;
00036         std::shared_ptr<ClientNetwork> _clientNetwork;
00037         std::thread _networkThread;
00038
00039         void initNetwork();
00040         void initLibraries();
00041         void networkLoop();
00042 };
00043
00044 #endif /* !CORE_HPP_ */
```

5.3 Core.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Core
00006 */
00007
00008 #include "Server.hpp"
00009 #include "Utils.hpp"
00010 #include "ServerConfig.hpp"
00011 #include "../common/ECS/resourceManager/ResourceManager.hpp"
00012
00013 #ifndef CORE_HPP_
00014 #define CORE_HPP_
00015
00016 class Core {
00017     public:
00018         Core();
00019         ~Core();
00020
00021         void init();
00022         void loop();
00023
00024         std::shared_ptr<rserv::ServerConfig> getConfig() const;
00025         std::shared_ptr<rserv::Server> getServer() const;
00026         std::shared_ptr<ecs::ResourceManager> getResourceManager() const;
00027
00028     protected:
00029     private:
00030         std::shared_ptr<Utils> _utils;
00031         std::shared_ptr<rserv::ServerConfig> _config;
00032         std::shared_ptr<rserv::Server> _server;
00033
```

```
00034         /* Add esc related elem here */
00035         std::shared_ptr<ecs::ResourceManager> _resourceManager;
00036 };
00037
00038 #endif /* !CORE_HPP_ */
```

5.4 AGameStateMachine.hpp

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

5.5 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
```

5.6 GameStateMachine.hpp

```
00001 #pragma once
00002
00003 #include "AGameStateMachine.hpp"
00004
00005 namespace gsm {
00006
00007 class GameStateMachine : public AGameStateMachine {
00008 public:
00009     GameStateMachine();
00010     ~GameStateMachine() override = default;
00011 };
00012
00013 }  // namespace gsm
```

5.7 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
```

5.8 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
```

5.9 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
```

5.10 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
```

5.11 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
```

5.12 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
```

5.13 DevState.hpp

```
00001 #pragma once
00002
00003 #include "../base/AGameState.hpp"
00004 #include "../../../../common/ECS/resourceManager/ResourceManager.hpp"
00005 #include "../../../../common/ECS/entity/registry/ARegistry.hpp"
00006 #include "../../../../common/ECS/system/systemManager/ASystemManager.hpp"
00007 #include "../../../../common/ECS/system/movement/MovementSystem.hpp"
```

```
00008 #include "../../../../common/ECS/system/movement/InputToVelocitySystem.hpp"
00009 #include "../../../../common/ECS/system/input/MovementInputSystem.hpp"
00010 #include "../../../../common/ECS/system/rendering/SpriteRenderingSystem.hpp"
00011 #include "../../../../common/Prefab/entityPrefabManager/EntityPrefabManager.hpp"
00012 namespace gsm {
00013
00014 class DevState : public AGameState {
00015 public:
00016     DevState(std::shared_ptr<IGameStateMachine> gsm, std::shared_ptr<ecs::ResourceManager>
      resourceManager);
00017     ~DevState() override = default;
00018
00019     void enter() override;
00020     void update(float deltaTime) override;
00021     void render() override;
00022     void exit() override;
00023
00024 private:
00025     std::shared_ptr<ecs::ResourceManager> _resourceManager;
00026     std::shared_ptr<ecs::ARegistry> _registry;
00027     std::shared_ptr<ecs::ASystemManager> _systemManager;
00028     std::shared_ptr<ecs::MovementSystem> _movementSystem;
00029     std::shared_ptr<ecs::InputToVelocitySystem> _inputToVelocitySystem;
00030     std::shared_ptr<ecs::MovementInputSystem> _inputSystem;
00031     std::shared_ptr<ecs::SpriteRenderingSystem> _spriteRenderingSystem;
00032     std::shared_ptr<EntityPrefabManager> _prefabManager;
00033 };
00034
00035 }  // namespace gsm
```

5.14 GraphicalInputProvider.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** GraphicalInputProvider
00006 */
00007
00008 #include "../../common/ECS/resourceManager/IInputProvider.hpp"
00009 #include "../../libs/Multimedia/IEvent.hpp"
00010 #include <memory>
00011
00012 namespace ecs {
00013
00014 class GraphicalInputProvider : public IInputProvider {
00015     public:
00016         GraphicalInputProvider(std::shared_ptr<gfx::IEvent> eventSystem);
00017         ~GraphicalInputProvider() override = default;
00018
00019         bool isKeyPressed(event_t key) override;
00020         float getAxisValue(event_t axis) override;
00021         std::pair<int, int> getMousePos() override;
00022         bool isMouseButtonPressed(int button) override;
00023
00024     private:
00025         std::shared_ptr<gfx::IEvent> _eventSystem;
00026 };
00027
00028 } // namespace ecs
```

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 #include "../../common/DLLoader/DLLoader.hpp"
00014 #include "../../libs/Multimedia/IWindow.hpp"
00015 #include "../../libs/Multimedia/IEvent.hpp"
```

```
00016
00017 std::shared_ptr<ecs::ResourceManager> initRessourcesManager(
00018     std::shared_ptr<DLLoader<gfx::createWindow_t>>,
00019     std::shared_ptr<DLLoader<gfx::createEvent_t>>
00020 );
00021
00022 #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 Utils.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Utils
00006 */
00007 #include "ClientNetwork.hpp"
00008
00009 #ifndef UTILS_HPP_
00010 #define UTILS_HPP_
00011
00012 class Utils {
00013     public:
00014         Utils();
00015         ~Utils();
00016
00017         void helper();
00018         void parseCli(int ac, char **av, std::shared_ptr<ClientNetwork> clientNetwork);
00019     protected:
00020     private:
00021 };
00022
00023 #endif /* !UTILS_HPP_ */
```

5.18 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_ */
```

5.19 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 #include <cstdint>
00012
00013 namespace constants {
00014     /* Network Defaults */
00015     constexpr int DEFAULT_SERVER_PORT = 4242;
00016     constexpr uint32_t DEFAULT_SERVER_IP = 0x7F000001;  // 127.0.0.1
00017
00018     /* Game Defaults */
00019     constexpr float BASE_SPEED = 100.0f;
00020     constexpr float EPS = 1e-6f;
00021     constexpr float PLAYER_BASE_SPEED = 300.0f;
00022     constexpr float GAMEPAD_DEADZONE = 0.15f;
00023     constexpr int SMOOTH_MOVEMENT_ITERATIONS = 4;
00024 }
00025
00026 #endif /* !CONSTANTS_HPP_ */
```

5.20 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
00041         void *Open(const char *path, int flag = RTLD_LAZY) override {
00042 #ifdef _WIN32
00043             (void)flag;
00044             _handler = LoadLibraryA(path);
00045             if (!_handler) {
00046                 _lastError = "Failed to load library: " + std::string(path);
00047             }
00048 #else
00049             _handler = dlopen(path, flag);
00050 #endif
00051             return _handler;
00052         };
00053
00054         void *Symbol(const char *symbolName) override {
00055 #ifdef _WIN32
00056             void *symbol = (void*)GetProcAddress(_handler, symbolName);
00057             if (!symbol) {
00058                 _lastError = "Failed to get symbol: " + std::string(symbolName);
00059                 std::cerr << "GetProcAddress error: " << _lastError << std::endl;
00060                 return nullptr;
00061             }
00062             return symbol;
00063 #else
00064             void *symbol = dlsym(_handler, symbolName);
00065             const char *error = dlerror();
00066             if (error) {
00067                 std::cerr << "dlerror: " << error << std::endl;
00068                 return nullptr;
00069             }
00070             return symbol;
00071 #endif
00072         };
00073
00074         T getSymbol(const char *symbolName) {
00075 #ifdef _WIN32
00076             return reinterpret_cast<T>(GetProcAddress(_handler, symbolName));
00077 #else
00078             return reinterpret_cast<T>(dlsym(_handler, symbolName));
00079 #endif
00080         };
00081
00082         int Close() override{
00083             if (_handler == nullptr)
00084                 return -1;
00085 #ifdef _WIN32
00086             return FreeLibrary(_handler) ? 0 : -1;
00087 #else
00088             return dlclose(_handler);
00089 #endif
00090         };
00091
00092         const char *Error() override {
00093 #ifdef _WIN32
00094             return _lastError.c_str();
00095 #else
00096             return dlerror();
00097 #endif
00098         };
00099 };
00100
00101 #endif /* !DLLOADER_HPP_ */
```

5.21 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_ */
```

5.22 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     MULTIMEDIA_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 typedef void *(*createNetworkLib_t)();
00022 typedef void *(*createBuffer_t)();
00023 typedef void *(*createPacket_t)();
00024
00025 #define pathLoad "./libraries"
00026
00027 #ifdef _WIN32
00028     #define multimediaLib "libMultimedia.dll"
00029     #define networkLib "libNetwork.dll"
00030     #define bufferLib "libBuffer.dll"
00031     #define packetLib "libPacket.dll"
00032 #else
00033     #define multimediaLib "libMultimedia.so"
00034     #define networkLib "libNetwork.so"
00035     #define bufferLib "libBuffer.so"
00036     #define packetLib "libPacket.so"
00037 #endif
00038
00039 #endif /* !LOADERTYPE_HPP_ */
```

5.23 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_ */
```

5.24 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_ */
```

5.25 ColliderComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ColliderComponent
00006 */
00007
00008 #ifndef COLLIDERCOMPONENT_HPP_
00009 #define COLLIDERCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/FRect.hpp"
00013 #include "../../../types/Vector2f.hpp"
00014
00015 namespace ecs {
00016
00017 enum class CollisionType {
00018     None,
00019     Solid,
00020     Bounce,
00021     Trigger
00022 };
00023
00024 class ColliderComponent : public AComponent {
00025     public:
00026         ColliderComponent(math::Vector2f offset = math::Vector2f(0.0f, 0.0f), math::Vector2f size =
      math::Vector2f(0.0f, 0.0f), CollisionType type = CollisionType::Solid)
00027             : _offset(offset), _size(size), _type(type) {};
00028         ~ColliderComponent() = default;
00029
00030         math::Vector2f getOffset() const { return _offset; };
```

```
00031         void setOffset(math::Vector2f offset) { _offset = offset; };
00032
00033         math::Vector2f getSize() const { return _size; };
00034         void setSize(math::Vector2f size) { _size = size; };
00035
00036         CollisionType getType() const { return _type; };
00037         void setType(CollisionType type) { _type = type; };
00038
00039         math::FRect getHitbox(math::Vector2f entityPosition) const {
00040             return math::FRect(entityPosition.getX() + _offset.getX(), entityPosition.getY() +
      _offset.getY(), _size.getX(), _size.getY());
00041         };
00042
00043     private:
00044         math::Vector2f _offset;
00045         math::Vector2f _size;
00046         CollisionType _type;
00047 };
00048
00049 }  // namespace ecs
00050
00051 #endif /* !COLLIDERCOMPONENT_HPP_ */
```

5.26 DirectionComponent.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(const math::Vector2f &velocity = math::Vector2f(0.0f, 0.0f))
00019             : _velocity(velocity) {};
00020         ~VelocityComponent() = default;
00021
00022         math::Vector2f getVelocity() const { return _velocity; };
00023         void setVelocity(const math::Vector2f &velocity) { _velocity = velocity; };
00024     private:
00025         math::Vector2f _velocity;
00026 };
00027
00028 }  // namespace ecs
00029
00030 #endif /* !VELOCITYCOMPONENT_HPP_ */
```

5.27 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_ */
```

5.28 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_ */
```

5.29 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_ */
```

5.30 AnimationComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AnimationComponent
00006 */
00007
00008 #ifndef ANIMATIONCOMPONENT_HPP_
00009 #define ANIMATIONCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/FRect.hpp"
00013 #include "../../../types/Chrono.hpp"
00014
00015 namespace ecs {
00016
00017 class AnimationComponent : public AComponent {
00018     public:
00019         AnimationComponent(const std::string& texturePath, float frameWidth, float frameHeight, int
      frameCount,
00020             float startWidth, float startHeight, float speed = 0.1f)
00021             : _texturePath(texturePath),
00022               _frameRect(0.0f, 0.0f, frameWidth, frameHeight),
00023               _frameCount(frameCount), _currentFrame(0), _animationSpeed(speed),
00024               _chrono(), _startHeight(startHeight), _startWidth(startWidth) {
00025             _chrono.start();
00026         }
00027
00028         const math::FRect& getFrameRect() const { return _frameRect; }
00029         void setFrameRect(const math::FRect& rect) { _frameRect = rect; }
00030
00031         float getFrameWidth() const { return _frameRect.getWidth(); }
00032         float getFrameHeight() const { return _frameRect.getHeight(); }
00033         int getFrameCount() const { return _frameCount; }
00034         int getCurrentFrame() const { return _currentFrame; }
00035         void setCurrentFrame(int frame) { _currentFrame = frame; }
00036         float getAnimationSpeed() const { return _animationSpeed; }
00037         void setAnimationSpeed(float speed) { _animationSpeed = speed; }
00038         math::Chrono& getChrono() { return _chrono; } // to change
00039         const math::Chrono& getChrono() const { return _chrono; }
00040         bool isValid() const { return _frameCount > 0 && _frameRect.getWidth() > 0 &&
      _frameRect.getHeight() > 0; }
00041         const std::string& getTexturePath() const { return _texturePath; }
00042         float getStartWidth() const { return _startWidth; }
00043         void setStartWidth(float startWidth) { _startWidth = startWidth; }
00044         float getStartHeight() const { return _startHeight; }
00045         void setStartHeight(float startHeight) { _startHeight = startHeight; }
00046
00047     private:
00048         std::string _texturePath;
00049         math::FRect _frameRect;
00050         int _frameCount;
00051         int _currentFrame;
00052         float _animationSpeed;
00053         math::Chrono _chrono;
00054         float _startHeight;
00055         float _startWidth;
00056 };
00057
00058 } // namespace ecs
00059 #endif /* !ANIMATIONCOMPONENT_HPP_ */
```

5.31 HitboxRenderComponent.hpp

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

```
00005 ** HitboxRenderComponent
00006 */
00007
00008 #ifndef HITBOXRENDERCOMPONENT_HPP_
00009 #define HITBOXRENDERCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../../libs/Multimedia/IWindow.hpp"
00013
00014 namespace ecs {
00015
00016 class HitboxRenderComponent : public AComponent {
00017     public:
00018         HitboxRenderComponent() : _color{255, 255, 255}, _outlineThickness(1.0f) {}
00019         HitboxRenderComponent(gfx::color_t color, float outlineThickness = 1.0f)
00020             : _color(color), _outlineThickness(outlineThickness) {}
00021
00022         ~HitboxRenderComponent() = default;
00023
00024         const gfx::color_t& getColor() const { return _color; }
00025         void setColor(const gfx::color_t& color) { _color = color; }
00026
00027         float getOutlineThickness() const { return _outlineThickness; }
00028         void setOutlineThickness(float thickness) { _outlineThickness = thickness; }
00029
00030     private:
00031         gfx::color_t _color;
00032         float _outlineThickness;
00033 };
00034
00035 }  // namespace ecs
00036
00037 #endif /* !HITBOXRENDERCOMPONENT_HPP_ */
```

5.32 RectangleRenderComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** RectangleRenderComponent
00006 */
00007
00008 #ifndef RECTANGLERENDERCOMPONENT_HPP_
00009 #define RECTANGLERENDERCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../../libs/Multimedia/IWindow.hpp"
00013
00014 namespace ecs {
00015
00016 class RectangleRenderComponent : public AComponent {
00017     public:
00018         RectangleRenderComponent() : _color{255, 255, 255}, _size{10.0f, 10.0f} {}
00019         RectangleRenderComponent(gfx::color_t color, float width, float height)
00020             : _color(color), _size{width, height} {}
00021
00022         ~RectangleRenderComponent() = default;
00023
00024         const gfx::color_t& getColor() const { return _color; }
00025         void setColor(const gfx::color_t& color) { _color = color; }
00026
00027         float getWidth() const { return _size.first; }
00028         float getHeight() const { return _size.second; }
00029         void setSize(float width, float height) { _size = {width, height}; }
00030
00031     private:
00032         gfx::color_t _color;
00033         std::pair<float, float> _size;
00034 };
00035
00036 }  // namespace ecs
00037
00038 #endif /* !RECTANGLERENDERCOMPONENT_HPP_ */
```

5.33 SpriteComponent.hpp

```
00001 /*
```

```
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** SpriteComponent
00006 */
00007
00008 #ifndef SPRITECOMPONENT_HPP_
00009 #define SPRITECOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/FRect.hpp"
00013 #include <string>
00014
00015 namespace ecs {
00016
00017 class SpriteComponent : public AComponent {
00018     public:
00019         SpriteComponent() : _texturePath("") {}
00020         SpriteComponent(const std::string& texturePath)
00021             : _texturePath(texturePath) {}
00022
00023         ~SpriteComponent() = default;
00024         const std::string& getTexturePath() const { return _texturePath; }
00025         void setTexturePath(const std::string& path) { _texturePath = path; }
00026         bool isValid() const { return !_texturePath.empty(); }
00027
00028     private:
00029         std::string _texturePath;
00030 };
00031
00032 } // namespace ecs
00033
00034 #endif /* !SPRITECOMPONENT_HPP_ */
```

5.34 ControllableTag.hpp

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

5.35 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_ */
```

5.36 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_ */
```

5.37 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_ */
```

5.38 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_ */
```

5.39 InputIntentComponent.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** InputIntentComponent
00006 */
00007
00008 #ifndef INPUTINTENTCOMPONENT_HPP_
00009 #define INPUTINTENTCOMPONENT_HPP_
00010
00011 #include "../base/AComponent.hpp"
00012 #include "../../../types/Vector2f.hpp"
00013
00014 namespace ecs {
00015
00016 class InputIntentComponent : public AComponent {
00017     public:
00018         InputIntentComponent(const math::Vector2f &direction = math::Vector2f(0.0f, 0.0f))
00019             : _direction(direction) {
00020             _state = Temporary;
00021         };
00022         ~InputIntentComponent() = default;
00023
00024         math::Vector2f getDirection() const { return _direction; };
00025         void setDirection(const math::Vector2f &direction) { _direction = direction; };
00026
00027     private:
00028         math::Vector2f _direction;
00029 };
00030
00031 }  // namespace ecs
00032
00033 #endif /* !INPUTINTENTCOMPONENT_HPP_ */
```

5.40 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_ */
```

5.41 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_ */
```

5.42 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(size_t entityId, std::shared_ptr<T> component);
00025         std::shared_ptr<T> get(size_t entityId) const;
00026         std::vector<std::shared_ptr<T» getAll(size_t entityId) const;
00027         void remove(size_t entityId);
00028         bool has(size_t entityId) const;
00029
00030         void removeAllComponentsWithState(ComponentState state) override;
00031         size_t getMaxEntityId() const override;
00032
00033     private:
```

```
00034         std::vector<std::vector<std::shared_ptr<T>> _components;
00035 };
00036
00037 } // namespace ecs
00038
00039 #include "AComponentArray.tpp"
00040
00041 #endif /* !AComposantType_HPP_ */
```

5.43 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_ */
```

5.44 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_ */
```

5.45 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 #include <string>
00019
00020 namespace ecs {
00021
00022 class ARegistry : public IRegistry, public std::enable_shared_from_this<ARegistry> {
00023     public:
00024         ARegistry();
00025         virtual ~ARegistry();
00026
00027         template <typename T>
00028         void registerComponent();
00029
00030         template <typename T>
00031         void addComponent(size_t entityId, std::shared_ptr<T> component);
00032         template <typename T>
00033         std::shared_ptr<T> getComponent(size_t entityId) const;
00034         template <typename T>
00035         std::vector<std::shared_ptr<T» getComponents(size_t entityId) const;
00036         template <typename T>
00037         void removeComponent(size_t entityId);
00038         template <typename T>
00039         bool hasComponent(size_t entityId) const;
00040
00041         template <typename... Components>
00042         View<Components...> view();
00043
00044         template <typename... Components>
00045         Group<Components...> group();
00046
00047         size_t getMaxEntityId() const;
00048
00049         void removeAllComponentsWithState(ComponentState state) override;
00050         size_t createEntity() override;
00051     protected:
00052     private:
00053         size_t _nextEntityId;
00054         std::unordered_map<std::string, std::shared_ptr<IComponentArray» _components;
00055 };
00056
00057 } // namespace ecs
00058
00059 #include "ARegistry.tpp"
00060
00061 #endif /* !AREGISTRY_HPP_ */
```

5.46 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(size_t entityId, std::shared_ptr<T> component);
00026
00027         template <typename T>
00028         std::shared_ptr<T> getComponent(size_t entityId);
00029
00030         template <typename T>
00031         void removeComponent(size_t entityId);
00032
00033         template <typename T>
00034         bool hasComponent(size_t 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         virtual size_t createEntity() = 0;
00047 };
00048
00049 } // namespace ecs
00050
00051 #endif /* !IREGISTRY_HPP_ */
```

5.47 IInputProvider.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IInputProvider
00006 */
00007
00008 #ifndef IINPUTPROVIDER_HPP_
00009 #define IINPUTPROVIDER_HPP_
00010
00011 #include <utility>
00012 #include "../../libs/Multimedia/EventTypes.hpp"
00013
00014 namespace ecs {
00015
00016 class IInputProvider {
00017     public:
00018         using event_t = gfx::EventType;
00019         virtual ~IInputProvider() = default;
00020         virtual bool isKeyPressed(event_t key) = 0;
00021         virtual float getAxisValue(event_t axis) = 0;
00022         virtual std::pair<int, int> getMousePos() = 0;
00023         virtual bool isMouseButtonPressed(int button) = 0;
00024 };
00025
00026 } // namespace ecs
00027
00028 #endif /* !IINPUTPROVIDER_HPP_ */
```

5.48 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_ */
```

5.49 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_ */
```

5.50 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_ */
```

5.51 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/InputIntentComponent.hpp"
00013 #include "../../resourceManager/IInputProvider.hpp"
00014 #include <memory>
00015
00016 namespace gfx {
00017     class IEvent;
00018 }
00019
00020 namespace ecs {
00021
00022 class MovementInputSystem : public ASystem {
00023     public:
00024         MovementInputSystem();
00025         ~MovementInputSystem() = default;
00026
00027         void update(std::shared_ptr<ResourceManager> resourceManager, std::shared_ptr<ARegistry>
      registry, float deltaTime) override;
00028
00029     private:
00030         math::Vector2f getMovementDirection(std::shared_ptr<ResourceManager> resourceManager) const;
00031         void updateInputIntent(std::shared_ptr<ARegistry> registry, size_t entityId, const
      math::Vector2f &direction);
00032         math::Vector2f getAnalogStickInput(std::shared_ptr<IInputProvider> inputProvider) const;
00033 };
00034
00035 } // namespace ecs
00036
00037 #endif /* !MOVEMENTINPUTSYSTEM_HPP_ */
```

5.52 InputToVelocitySystem.hpp

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

5.53 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 "../../component/permanent/VelocityComponent.hpp"
00017 #include "../../component/permanent/ColliderComponent.hpp"
00018 #include "../../component/tags/ObstacleTag.hpp"
00019 #include "../../../constants.hpp"
00020
00021 namespace ecs {
00022
00023 class MovementSystem : public ASystem {
00024     public:
00025         MovementSystem();
00026         ~MovementSystem() = default;
00027
00028         void update(
00029             std::shared_ptr<ResourceManager> resourceManager,
00030             std::shared_ptr<ARegistry> registry,
00031             float deltaTime
00032         ) override;
00033
00034     private:
00035         bool checkCollision(
00036             std::shared_ptr<ARegistry> registry,
00037             size_t entityId,
00038             math::Vector2f newPos
00039         );
00040         math::Vector2f calculateSmoothMovement(
00041             std::shared_ptr<ARegistry> registry,
00042             size_t entityId,
00043             math::Vector2f startPos,
00044             math::Vector2f desiredPos
00045         );
00046         math::Vector2f calculateSlidingMovement(
00047             std::shared_ptr<ARegistry> registry,
00048             size_t entityId,
00049             math::Vector2f basePos,
00050             math::Vector2f desiredPos
00051         );
00052         math::Vector2f calculateSmoothSlidingPosition(
00053             std::shared_ptr<ARegistry> registry,
00054             size_t entityId,
00055             math::Vector2f startPos,
00056             math::Vector2f desiredPos
00057         );
00058         math::Vector2f handleBounceCollision(
00059             std::shared_ptr<ARegistry> registry,
00060             size_t entityId,
00061             math::Vector2f startPos,
00062             math::Vector2f desiredPos,
00063             std::shared_ptr<ecs::VelocityComponent> velocityComp
00064         );
00065 };
00066
00067 }  // namespace ecs
00068
00069 #endif /* !MOVEMENTSYSTEM_HPP_ */
```

5.54 AnimationRenderingSystem.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** AnimationRenderingSystem
00006 */
00007
00008 #ifndef ANIMATIONRENDERINGSYSTEM_HPP_
```

```
00009 #define ANIMATIONRENDERINGSYSTEM_HPP_
00010
00011
00012 #include "../base/ASystem.hpp"
00013 #include <memory>
00014 namespace ecs {
00015
00016 class AnimationRenderingSystem : public ASystem {
00017     public:
00018         AnimationRenderingSystem();
00019         ~AnimationRenderingSystem() override = default;
00020
00021     protected:
00022         void update(std::shared_ptr<ResourceManager> resourceManager,
00023         std::shared_ptr<ARegistry> registry, float deltaTime) override;
00024 };
00025
00026 }  // namespace ecs
00027
00028 #endif /* !ANIMATIONRENDERINGSYSTEM_HPP_ */
```

5.55 HitboxRenderingSystem.hpp

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

5.56 RectangleRenderingSystem.hpp

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

```
00025
00026 } // namespace ecs
00027
00028 #endif /* !RECTANGLERENDERINGSYSTEM_HPP_ */
```

5.57 SpriteRenderingSystem.hpp

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

5.58 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_ */
```

5.59 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_ */
```

5.60 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_ */
```

5.61 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 override = default;
00021         const char *what() const noexcept override;
00022         int getCode() const noexcept override;
00023         std::string getDetails() const noexcept override;
00024
00025         virtual std::string getType() const noexcept override = 0;
00026
00027     protected:
00028         std::string m_message;
00029         int m_code;
00030 };
00031
00032 }
00033
00034 #endif /* !AERROR_HPP_ */
```

5.62 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     private:
00027 };
00028
00029 }
00030
00031 #endif /* !IERROR_HPP_ */
00032
```

5.63 PacketError.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** R-Type
00004 ** File description:
00005 ** Header
00006 */
00007
00008 #ifndef PACKET_ERROR_HPP
00009     #define PACKET_ERROR_HPP
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015 class PacketError : public AError {
00016     public:
00017         enum ErrorCode {
00018             UNKNOWN = 1000,
00019             SERIALIZER_ATTRIBUTION_FAILED = 1001
00020         };
00021
00022         PacketError(const std::string &message, ErrorCode code = UNKNOWN);
00023         ~PacketError() override;
00024         std::string getType() const noexcept override;
00025
00026     protected:
00027     private:
00028 };
00029
00030 }
00031
00032 #endif /* !PACKET_ERROR_HPP */
```

5.64 ServerErrror.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerErrror
00006 */
00007
00008 #ifndef SERVER_ERROR_HPP
00009     #define SERVER_ERROR_HPP
00010
00011 #include "AError.hpp"
00012
00013 namespace err {
00014
00015 class ServerError : public AError {
00016     public:
```

```
00017         enum ErrorCode {
00018             UNKNOWN = 1000,
00019             CONNECTION_FAILED = 1001,
00020             TIMEOUT = 1002,
00021             INVALID_REQUEST = 1003,
00022             INTERNAL_ERROR = 1004,
00023             LIBRARY_LOAD_FAILED = 1005,
00024             CONFIG_ERROR = 1006
00025         };
00026
00027         ServerError(const std::string &message, ErrorCode code = UNKNOWN);
00028         virtual ~ServerError() noexcept = default;
00029         std::string getType() const noexcept override;
00030
00031     private:
00032 };
00033
00034 }
00035
00036 #endif /* !SERVER_ERROR_HPP */
```

5.65 APrefab.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** APrefab
00006 */
00007
00008 #ifndef APREFAB_HPP_
00009 #define APREFAB_HPP_
00010
00011 #include "IPrefab.hpp"
00012 #include "../ECS/entity/registry/ARegistry.hpp"
00013
00014 class APrefab : public IPrefab {
00015     public:
00016         APrefab() = default;
00017         virtual ~APrefab() = default;
00018         size_t instantiate(const std::shared_ptr<ecs::ARegistry> &registry) override;
00019 };
00020
00021 #endif /* !APREFAB_HPP_ */
```

5.66 EntityPrefabManager.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** EntityPrefabManager
00006 */
00007
00008 #ifndef ENTITYPREFABMANAGER_HPP_
00009 #define ENTITYPREFABMANAGER_HPP_
00010
00011 #include <vector>
00012 #include <string>
00013 #include <map>
00014 #include <memory>
00015 #include "../../ECS/entity/IEntity.hpp"
00016 #include "../IPrefab.hpp"
00017
00018 class EntityPrefabManager
00019 {
00020     public:
00021         EntityPrefabManager();
00022         ~EntityPrefabManager();
00023
00024         void registerPrefab(const std::string &name, const std::shared_ptr<IPrefab> &prefab);
00025         std::shared_ptr<IPrefab> getPrefab(const std::string &name) const;
00026         size_t createEntityFromPrefab(const std::string &prefabName, const
      std::shared_ptr<ecs::ARegistry> &registry);
00027         bool hasPrefab(const std::string &name) const;
00028         void deletePrefab(const std::string &name);
00029         void clearPrefabs();
00030     private:
```

```
00031         std::map<std::string, std::shared_ptr<IPrefab» _prefabs;
00032 };
00033
00034 #endif /* !ENTITYPREFABMANAGER_HPP_ */
```

5.67 IPrefab.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** IPrefab
00006 */
00007
00008 #ifndef IPREFAB_HPP_
00009 #define IPREFAB_HPP_
00010 #include <memory>
00011 #include "../ECS/entity/registry/ARegistry.hpp"
00012
00013 class IPrefab {
00014     public:
00015         virtual ~IPrefab() = default;
00016         virtual size_t instantiate(const std::shared_ptr<ecs::ARegistry> &registry) = 0;
00017 };
00018
00019 #endif /* !IPREFAB_HPP_ */
```

5.68 PlayerPrefab.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** PlayerPrefab
00006 */
00007
00008 #ifndef PLAYERPREFAB_HPP_
00009 #define PLAYERPREFAB_HPP_
00010
00011 #include "../APrefab.hpp"
00012 #include "../../ECS/component/permanent/TransformComponent.hpp"
00013 #include "../../ECS/component/permanent/VelocityComponent.hpp"
00014 #include "../../ECS/component/permanent/SpeedComponent.hpp"
00015 #include "../../ECS/component/rendering/SpriteComponent.hpp"
00016 #include "../../ECS/component/rendering/AnimationComponent.hpp"
00017 #include "../../ECS/component/tags/ControllableTag.hpp"
00018 #include "../../ECS/component/tags/PlayerTag.hpp"
00019 #include "../../ECS/component/permanent/ColliderComponent.hpp"
00020 #include "../../types/Vector2f.hpp"
00021 #include <memory>
00022 #include <string>
00023 #include <iostream>
00024
00025
00026 class PlayerPrefab : public APrefab {
00027     public:
00028         PlayerPrefab(float x, float y, float scale,
00029                      const std::string &animationPath,
00030                      float frameWidth, float frameHeight,
00031                      float startWidth, float startHeight,
00032                      int frameCount)
00033             : _x(x), _y(y), _scale(scale),
00034               _animationPath(animationPath),
00035               _frameWidth(frameWidth),
00036               _frameHeight(frameHeight),
00037               _startWidth(startWidth),
00038               _startHeight(startHeight),
00039               _frameCount(frameCount) {}
00040
00041         ~PlayerPrefab() = default;
00042
00043         size_t instantiate(const std::shared_ptr<ecs::ARegistry> &registry) override {
00044             size_t entity = registry->createEntity();
00045             auto transform = std::make_shared<ecs::TransformComponent>(math::Vector2f(_x, _y));
00046             transform->setScale(math::Vector2f(_scale, _scale));
00047             registry->addComponent(entity, transform);
00048             registry->addComponent(entity, std::make_shared<ecs::VelocityComponent>());
00049             registry->addComponent(entity, std::make_shared<ecs::SpeedComponent>());
```

```
00050             registry->addComponent(entity,
00051                 std::make_shared<ecs::AnimationComponent>(_animationPath, _frameWidth, _frameHeight,
    _frameCount, _startWidth, _startHeight));
00052             registry->addComponent(entity, std::make_shared<ecs::PlayerTag>());
00053             registry->addComponent(entity, std::make_shared<ecs::ControllableTag>());
00054             auto collider = std::make_shared<ecs::ColliderComponent>(
00055                 math::Vector2f(0.0f, 0.0f),
00056                 math::Vector2f(100.0f, 100.0f),
00057                 ecs::CollisionType::Solid
00058             );
00059             registry->addComponent(entity, collider);
00060             return entity;
00061         }
00062
00063     private:
00064         float _x;
00065         float _y;
00066         float _scale;
00067         std::string _animationPath;
00068         float _frameWidth, _frameHeight;
00069         float _startWidth;
00070         float _startHeight;
00071         int _frameCount;
00072 };
00073
00074 #endif /* !PLAYERPREFAB_HPP_ */
```

5.69 Signal.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Signal
00006 */
00007
00008 #ifndef SIGNAL_HPP_
00009 #define SIGNAL_HPP_
00010
00011 #ifdef _WIN32
00012     #ifndef _WIN32_WINNT
00013         #define _WIN32_WINNT 0x0A00
00014     #endif
00015
00016     #ifndef WIN32_LEAN_AND_MEAN
00017         #define WIN32_LEAN_AND_MEAN
00018     #endif
00019 #endif
00020
00021 #include <csignal>
00022
00023 class Signal {
00024     public:
00025         Signal();
00026         ~Signal();
00027
00028         static volatile sig_atomic_t stopFlag;
00029         static void signalHandler(int signum);
00030         static void setupSignalHandlers();
00031
00032     protected:
00033     private:
00034 };
00035
00036 #endif /* !SIGNAL_HPP_ */
```

5.70 Chrono.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** Chrono wrapper for std::chrono
00006 */
00007
00008 #ifndef CHRONO_HPP_
00009 #define CHRONO_HPP_
00010
```

```
00011 #include <chrono>
00012
00013 namespace math {
00014
00015 class Chrono {
00016     public:
00017         Chrono();
00018         ~Chrono() = default;
00019
00020         void start();
00021         void stop();
00022         void reset();
00023         float getElapsedSeconds() const;
00024         float getElapsedMilliseconds() const;
00025         bool isRunning() const;
00026
00027     private:
00028         std::chrono::high_resolution_clock::time_point _startTime;
00029         std::chrono::high_resolution_clock::time_point _stopTime;
00030         bool _isRunning;
00031 };
00032
00033 } // namespace math
00034
00035 #endif /* !CHRONO_HPP_ */
```

5.71 FRect.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** FRect
00006 */
00007
00008 #ifndef FRECT_HPP_
00009 #define FRECT_HPP_
00010
00011 namespace math {
00012
00013 class FRect {
00014     public:
00015         FRect();
00016         FRect(float left, float top, float width, float height);
00017         FRect(FRect const &other);
00018         ~FRect() = default;
00019
00020         float getLeft() const;
00021         void setLeft(float left);
00022         float getTop() const;
00023         void setTop(float top);
00024         float getWidth() const;
00025         void setWidth(float width);
00026         float getHeight() const;
00027         void setHeight(float height);
00028
00029         bool contains(float x, float y) const;
00030         bool intersects(FRect const &other) const;
00031         bool intersects(FRect const &other, FRect &intersection) const;
00032
00033         FRect &operator=(FRect const &other);
00034         bool operator==(FRect const &other) const;
00035         bool operator!=(FRect const &other) const;
00036
00037         private:
00038         float left;
00039         float top;
00040         float width;
00041         float height;
00042 };
00043
00044 } // namespace math
00045
00046 #endif /* !FRECT_HPP_ */
```

5.72 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_ */
```

5.73 ServerInputProvider.hpp

```
00001 /*
00002 ** EPITECH PROJECT, 2025
00003 ** ryanR-type
00004 ** File description:
00005 ** ServerInputProvider
00006 */
00007
00008 #include "../../common/ECS/resourceManager/IInputProvider.hpp"
00009
00010 namespace ecs {
00011
00012 class ServerInputProvider : public IInputProvider {
00013     public:
00014         ServerInputProvider();
00015         ~ServerInputProvider() override = default;
00016
00017         bool isKeyPressed(event_t key) override;
00018         float getAxisValue(event_t axis) override;
00019         std::pair<int, int> getMousePos() override;
00020         bool isMouseButtonPressed(int button) override;
00021 };
00022
00023 } // namespace ecs
```

5.74 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/IPacketManager.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 IPacketManager &packet) = 0;
00045
00046     virtual void processConnections() = 0;
00047     virtual void processIncomingPackets() = 0;
00048
00049     virtual void broadcastPacket() = 0;
00050     virtual void sendToClient(int idClient) = 0;
00051     virtual std::vector<int> getConnectedClients() const = 0;
00052     virtual size_t getClientCount() const = 0;
00053 };
00054 } // namespace rserv = r-type server
00055 #endif /* !ISERVER_HPP_ */
```

5.75 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 #ifdef _WIN32
00012     #ifndef _WIN32_WINNT
00013         #define _WIN32_WINNT 0x0A00
00014     #endif
00015
00016     #ifndef WIN32_LEAN_AND_MEAN
00017         #define WIN32_LEAN_AND_MEAN
00018     #endif
00019 #endif
00020
00021 #include <memory>
00022 #include "IServer.hpp"
00023 #include "ServerConfig.hpp"
00024 #include "../libs/Network/INetwork.hpp"
00025 #include "../libs/Buffer/IBuffer.hpp"
00026 #include "../common/DLLoader/DLLoader.hpp"
00027 #include "../common/DLLoader/LoaderType.hpp"
00028 #include "Signal.hpp"
00029
00030
00031
00032 namespace rserv {
00033     class Server : public IServer {
00034         public:
00035             Server();
```

```
00036             ~Server();
00037
00038             void init() override;
00039             void start() override;
00040             void stop() override;
00041
00042             void setConfig(std::shared_ptr<ServerConfig> config) override;
00043             std::shared_ptr<ServerConfig> getConfig() const override;
00044             unsigned int getPort() const override;
00045             void setPort(unsigned int port) override;
00046
00047             int getState() const override;
00048             void setState(int state) override;
00049
00050             int getFd() const override;
00051             void setFd(int fd) override;
00052             operator int() const noexcept override;
00053
00054             std::shared_ptr<net::INetwork> getNetwork() const override;
00055             void setNetwork(std::shared_ptr<net::INetwork> network) override;
00056
00057             void onClientConnected(int idClient) override;
00058             void onClientDisconnected(int idClient) override;
00059             void onPacketReceived(int idClient, const IPacketManager &packet) override;
00060
00061             void processConnections() override;
00062             void processIncomingPackets() override;
00063
00064             void broadcastPacket() override;
00065             void sendToClient(int idClient ) override;
00066             std::vector<int> getConnectedClients() const override;
00067             size_t getClientCount() const override;
00068
00069         private:
00070             void loadNetworkLibrary();
00071             void loadBufferLibrary();
00072             void loadPacketLibrary();
00073             DLLoader<createNetworkLib_t> _networloader;
00074             DLLoader<createBuffer_t> _bufferloader;
00075             DLLoader<createPacket_t> _packetloader;
00076
00077             std::shared_ptr<ServerConfig> _config;
00078             std::shared_ptr<net::INetwork> _network;
00079             std::shared_ptr<IBuffer> _buffer;
00080             std::shared_ptr<IPacketManager> _packet;
00081
00082     };
00083 } // namespace rserv = r-type server
00084
00085 #endif /* !SERVER_HPP_ */
```

5.76 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_ */
```

Index