Space Object Simulation

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

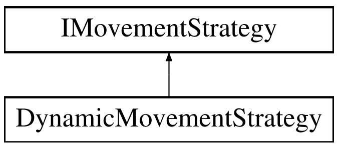
Class Documentation

4.1 DynamicMovementStrategy Class Reference

Movement strategy for dynamic movement based on gravitational forces.

```
#include <dynamic.hpp>
```

Inheritance diagram for DynamicMovementStrategy:



Public Member Functions

- DynamicMovementStrategy (double ∗mass_ptr, double ∗radius_ptr, sf::Vector2f ∗position_ptr, sf::Vector2f ∗velocity_ptr)

  Constructor that initializes the dynamic movement strategy.
- void update_velocity (const std::vector< const SpaceObject ∗ > &others, const float gravitational_constant, const float delta_time) override

  Updates the velocity based on gravitational interactions with other objects.
- void update_position (const float delta_time) override

  Updates the position based on the object's velocity.

Public Member Functions inherited from IMovementStrategy

- IMovementStrategy (double ∗mass_ptr, double ∗radius_ptr, sf::Vector2f ∗position_ptr, sf::Vector2f ∗velocity←_ptr)

  Constructor to initialize the movement strategy with required pointers.

Additional Inherited Members

Protected Attributes inherited from IMovementStrategy

- double ∗ **mass**

  Pointer to the object's mass.
- double ∗ **radius**

  Pointer to the object's radius.
- sf::Vector2f ∗ **position**

  Pointer to the object's position.
- sf::Vector2f ∗ **velocity**

  Pointer to the object's velocity.

4.1.1 Detailed Description

Movement strategy for dynamic movement based on gravitational forces.

Updates velocity and position of a space object considering gravitational forces from other objects.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 DynamicMovementStrategy()

```
DynamicMovementStrategy::DynamicMovementStrategy (
            double * mass_ptr,
            double * radius_ptr,
            sf::Vector2f * position_ptr,
            sf::Vector2f * velocity_ptr)  [inline]
```

Constructor that initializes the dynamic movement strategy.

Parameters

mass_ptr	Pointer to the object's mass.
radius_ptr	Pointer to the object's radius.
position_ptr	Pointer to the object's position.
velocity_ptr	Pointer to the object's velocity.

4.1.3 Member Function Documentation

4.1.3.1 update_position()

```
void DynamicMovementStrategy::update_position (
            const float delta_time)  [override], [virtual]
```

Updates the position based on the object's velocity.

Parameters

delta_time	Time step for the update.

Implements IMovementStrategy.

4.1.3.2 update_velocity()

```
void DynamicMovementStrategy::update_velocity (
            const std::vector< const SpaceObject * > & others,
            const float gravitational_constant,
            const float delta_time)  [override], [virtual]
```

Updates the velocity based on gravitational interactions with other objects.

Parameters

others	List of other space objects in the universe for gravitational calculation.
gravitational_constant	The gravitational constant.
delta_time	Time step for the update.

Implements IMovementStrategy.

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

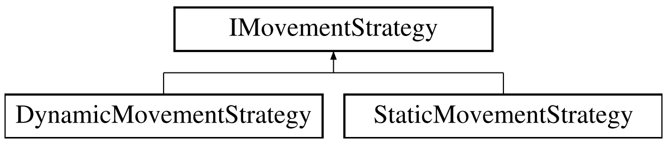
- movement/dynamic.hpp
- C:/Users/ugniu/code/cpp-2025/src/core/movement_dynamic.cpp

4.2 IMovementStrategy Class Reference

Abstract base class for movement strategies of space objects.

`#include <i_movement_strategy.hpp>`

Inheritance diagram for IMovementStrategy:



Public Member Functions

- IMovementStrategy (double *mass_ptr, double *radius_ptr, sf::Vector2f *position_ptr, sf::Vector2f *velocity↩ _ptr)

    Constructor to initialize the movement strategy with required pointers.
- virtual void update_velocity (const std::vector< const SpaceObject * > &others, const float gravitational_↩ constant, const float delta_time)=0

    Updates the velocity based on the movement strategy.
- virtual void update_position (const float delta_time)=0

    Updates the position based on the movement strategy.

Protected Attributes

- double ∗ **mass**

  Pointer to the object's mass.
- double ∗ **radius**

  Pointer to the object's radius.
- sf::Vector2f ∗ **position**

  Pointer to the object's position.
- sf::Vector2f ∗ **velocity**

  Pointer to the object's velocity.

4.2.1 Detailed Description

Abstract base class for movement strategies of space objects.

Defines the interface for updating velocity and position of space objects.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 IMovementStrategy()

```
IMovementStrategy::IMovementStrategy (
             double * mass_ptr,
             double * radius_ptr,
             sf::Vector2f * position_ptr,
             sf::Vector2f * velocity_ptr)  [inline]
```

Constructor to initialize the movement strategy with required pointers.

Parameters

mass_ptr	Pointer to the object's mass.
radius_ptr	Pointer to the object's radius.
position_ptr	Pointer to the object's position.
velocity_ptr	Pointer to the object's velocity.

4.2.3 Member Function Documentation

4.2.3.1 update_position()

```
virtual void IMovementStrategy::update_position (
             const float delta_time)  [pure virtual]
```

Updates the position based on the movement strategy.

Parameters

delta_time	Time step for the update.

Implemented in DynamicMovementStrategy, and StaticMovementStrategy.

4.2.3.2 update_velocity()

```
virtual void IMovementStrategy::update_velocity (
             const std::vector< const SpaceObject * > & others,
             const float gravitational_constant,
             const float delta_time)  [pure virtual]
```

Updates the velocity based on the movement strategy.

Parameters

others	List of other space objects.
gravitational_constant	Gravitational constant.
delta_time	Time step for the update.

Implemented in DynamicMovementStrategy, and StaticMovementStrategy.

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

- movement/i_movement_strategy.hpp

4.3 SpaceObject Class Reference

Classes

- class SpaceObjectImpl

Public Member Functions

- SpaceObject (std::string name, double mass, double radius, sf::Vector2f position, sf::Vector2f velocity={0, 0}, bool is_movable=true)

  Constructs a SpaceObject with specified properties.
- SpaceObject (const SpaceObject &other)

  Copy constructor for SpaceObject.
- SpaceObject & operator= (const SpaceObject &other)

  Assignment operator for SpaceObject.
- void print_info (std::ostream &output) const

  Prints the information of the space object.
- void update_velocity (std::vector< const SpaceObject * > &others, const float gravitational_constant, const float delta_time)

  Updates the velocity of the space object.
- void update_position (const float delta_time)

  Updates the position of the space object.
- int get_id () const

  Gets the unique ID of the space object.
- std::string get_name () const

  Gets the name of the space object.
- double get_mass () const

  Gets the mass of the space object.

- double get_radius () const

    Gets the radius of the space object.
- sf::Vector2f get_velocity () const

    Gets the current velocity of the space object.
- sf::Vector2f get_position () const

    Gets the current position of the space object.
- bool is_movable () const

    Checks whether the space object is movable.
- void set_name (std::string name)

    Sets the name of the space object.
- void set_mass (double mass)

    Sets the mass of the space object.
- void set_radius (double radius)

    Sets the radius of the space object.
- void set_position (sf::Vector2f position)

    Sets the position of the space object.
- void set_velocity (sf::Vector2f velocity)

    Sets the velocity of the space object.
- void set_movability (bool movable)

    Sets whether the object is movable or static.

Static Public Member Functions

- static int get_object_count ()

    Gets the total number of active SpaceObject instances.

Friends

- std::ofstream & operator<< (std::ofstream &out, const SpaceObject &obj)

    Serialization operator (<<) for saving SpaceObject to a file.
- std::ifstream & operator>> (std::ifstream &in, SpaceObject &obj)

    Deserialization operator (>>) for reading SpaceObject from a file.

4.3.1 Constructor & Destructor Documentation

4.3.1.1 SpaceObject() [1/2]

```
SpaceObject::SpaceObject (
             std::string name,
             double mass,
             double radius,
             sf::Vector2f position,
             sf::Vector2f velocity = {0, 0},
             bool is_movable = true)
```

Constructs a SpaceObject with specified properties.

Parameters

name	Name of the space object.
mass	Mass of the space object.
radius	Radius of the space object.
position	Position of the space object.
velocity	Velocity of the space object (default is {0, 0}).
is_movable	Whether the space object is movable (default is true).

4.3.1.2 SpaceObject() [2/2]

```
SpaceObject::SpaceObject (
             const SpaceObject & other)
```

Copy constructor for SpaceObject.

Parameters

other	Another SpaceObject to copy from.

4.3.2 Member Function Documentation

4.3.2.1 get_id()

```
int SpaceObject::get_id () const
```

Gets the unique ID of the space object.

Returns

The ID of the space object.

4.3.2.2 get_mass()

```
double SpaceObject::get_mass () const
```

Gets the mass of the space object.

Returns

The mass as a double.

4.3.2.3 get_name()

```
std::string SpaceObject::get_name () const
```

Gets the name of the space object.

Returns

The name as a string.

4.3.2.4 get_object_count()

```
int SpaceObject::get_object_count () [static]
```

Gets the total number of active SpaceObject instances.

Returns

The current count of active SpaceObjects.

4.3.2.5 get_position()

```
sf::Vector2f SpaceObject::get_position () const
```

Gets the current position of the space object.

Returns

The position as an sf::Vector2f.

4.3.2.6 get_radius()

```
double SpaceObject::get_radius () const
```

Gets the radius of the space object.

Returns

The radius as a double.

4.3.2.7 get_velocity()

```
sf::Vector2f SpaceObject::get_velocity () const
```

Gets the current velocity of the space object.

Returns

The velocity as an sf::Vector2f.

4.3.2.8 is_movable()

```
bool SpaceObject::is_movable () const
```

Checks whether the space object is movable.

Returns

True if movable, false otherwise.

4.3.2.9 operator=()

```
SpaceObject & SpaceObject::operator= (
             const SpaceObject & other)
```

Assignment operator for SpaceObject.

Parameters

other	Another SpaceObject to assign from.

Returns

A reference to this SpaceObject.

4.3.2.10 print_info()

```
void SpaceObject::print_info (
             std::ostream & output) const
```

Prints the information of the space object.

Parameters

output	Output stream to print information.

4.3.2.11 set_mass()

```
void SpaceObject::set_mass (
             double mass)
```

Sets the mass of the space object.

Parameters

mass	New mass value.

4.3.2.12 set_movability()

```
void SpaceObject::set_movability (
             bool movable)
```

Sets whether the object is movable or static.

Parameters

movable	True to make the object movable, false to make it static.

4.3.2.13 set_name()

```
void SpaceObject::set_name (
             std::string name)
```

Sets the name of the space object.

Parameters

name	New name as a string.

4.3.2.14 set_position()

```
void SpaceObject::set_position (
             sf::Vector2f position)
```

Sets the position of the space object.

Parameters

position	New position as an sf::Vector2f.

4.3.2.15 set_radius()

```
void SpaceObject::set_radius (
             double radius)
```

Sets the radius of the space object.

Parameters

radius	New radius value.

4.3.2.16 set_velocity()

```
void SpaceObject::set_velocity (
             sf::Vector2f velocity)
```

Sets the velocity of the space object.

Parameters

velocity	New velocity as an sf::Vector2f.

4.3.2.17 update_position()

```
void SpaceObject::update_position (
             const float delta_time)
```

Updates the position of the space object.

The position is updated based on the object's velocity and movement strategy (either dynamic or static). If the object is dynamic, its position will be adjusted based on its velocity. If the object is static, its position will remain unchanged.

Parameters

delta_time	Time step for the update.

4.3.2.18 update_velocity()

```
void SpaceObject::update_velocity (
             std::vector< const SpaceObject * > & others,
             const float gravitational_constant,
             const float delta_time)
```

Updates the velocity of the space object.

The velocity is updated based on the object's movement strategy (either dynamic or static). If the object is dynamic, its velocity will be influenced by gravitational forces and other space objects. If the object is static, its velocity will be set to zero.

Parameters

others	List of other space objects for gravitational calculation.
gravitational_constant	Gravitational constant for the calculation.
delta_time	Time step for the update.

4.3.3 Friends And Related Symbol Documentation

4.3.3.1 operator<<

```
std::ofstream & operator<< (
             std::ofstream & out,
             const SpaceObject & obj)   [friend]
```

Serialization operator (<<) for saving SpaceObject to a file.

Parameters

out	Output file stream.
obj	SpaceObject to write.

Returns

Output file stream.

4.3.3.2 operator>>

```
std::ifstream & operator>> (
            std::ifstream & in,
            SpaceObject & obj)  [friend]
```

Deserialization operator (>>) for reading SpaceObject from a file.

Parameters

in	Input file stream.
obj	SpaceObject to read into.

Returns

Input file stream.

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

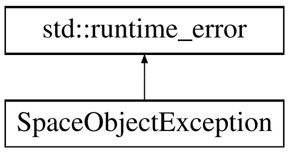
- space_object.hpp
- C:/Users/ugniu/code/cpp-2025/src/core/space_object.cpp

4.4 SpaceObjectException Class Reference

Exception class for handling errors related to SpaceObject.

`#include <exception.hpp>`

Inheritance diagram for SpaceObjectException:



Public Member Functions

- SpaceObjectException (const std::string &message)

    Constructs a SpaceObjectException with a given message.

4.4.1 Detailed Description

Exception class for handling errors related to SpaceObject.

This class extends `std::runtime_error` to provide a custom exception for errors that occur during the processing of space objects.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 SpaceObjectException()

```
SpaceObjectException::SpaceObjectException (
            const std::string & message)  [inline], [explicit]
```

Constructs a SpaceObjectException with a given message.

Parameters

message	The error message to describe the exception.

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

- exception.hpp

4.5 SpaceObject::SpaceObjectImpl Class Reference

Public Member Functions

- **SpaceObjectImpl** (std::string name, double mass, double radius, sf::Vector2f position, sf::Vector2f velocity, bool movable)
- **SpaceObjectImpl** (const SpaceObjectImpl &other)
- void **set_name** (std::string name)
- void **set_mass** (double mass)
- void **set_radius** (double radius)
- void **set_position** (sf::Vector2f position)
- void **set_velocity** (sf::Vector2f velocity)
- void **set_movability** (bool movable)
- void **set_movement_strategy** (bool movable)

Public Attributes

- int **id**
- std::string **name**
- double **mass**
- double **radius**
- sf::Vector2f **position**
- sf::Vector2f **velocity**
- bool **movable**
- IMovementStrategy ∗ **movement_strategy** = nullptr

Static Public Attributes

- static int **object_count** = 0

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

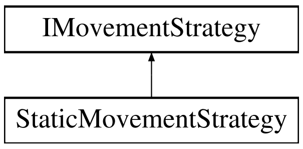
- C:/Users/ugniu/code/cpp-2025/src/core/space_object.cpp

4.6 StaticMovementStrategy Class Reference

Movement strategy for static objects with fixed position and velocity.

`#include <static.hpp>`

Inheritance diagram for StaticMovementStrategy:



Public Member Functions

- StaticMovementStrategy (double ∗mass_ptr, double ∗radius_ptr, sf::Vector2f ∗position_ptr, sf::Vector2f ∗velocity_ptr)

  Constructor that initializes the static movement strategy.
- void update_velocity (const std::vector< const SpaceObject ∗ > &others, const float gravitational_constant, const float delta_time) override

  Sets the objects velocity to zero.
- void update_position (const float delta_time) override

  Does not update the position for static objects.

Public Member Functions inherited from IMovementStrategy

- IMovementStrategy (double ∗mass_ptr, double ∗radius_ptr, sf::Vector2f ∗position_ptr, sf::Vector2f ∗velocity↩ _ptr)

  Constructor to initialize the movement strategy with required pointers.

Additional Inherited Members

Protected Attributes inherited from IMovementStrategy

- double ∗ **mass**

  Pointer to the object's mass.
- double ∗ **radius**

  Pointer to the object's radius.
- sf::Vector2f ∗ **position**

  Pointer to the object's position.
- sf::Vector2f ∗ **velocity**

  Pointer to the object's velocity.

4.6.1 Detailed Description

Movement strategy for static objects with fixed position and velocity.

Does not change the position or velocity of the space object.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 StaticMovementStrategy()

```
StaticMovementStrategy::StaticMovementStrategy (
             double * mass_ptr,
             double * radius_ptr,
             sf::Vector2f * position_ptr,
             sf::Vector2f * velocity_ptr)  [inline]
```

Constructor that initializes the static movement strategy.

Parameters

mass_ptr	Pointer to the object's mass.
radius_ptr	Pointer to the object's radius.
position_ptr	Pointer to the object's position.
velocity_ptr	Pointer to the object's velocity.

4.6.3 Member Function Documentation

4.6.3.1 update_position()

```
void StaticMovementStrategy::update_position (
             const float delta_time)  [override], [virtual]
```

Does not update the position for static objects.

Parameters

delta_time	Time step for the update.

Implements IMovementStrategy.

4.6.3.2 update_velocity()

```
void StaticMovementStrategy::update_velocity (
             const std::vector< const SpaceObject * > & others,
             const float gravitational_constant,
             const float delta_time)  [override], [virtual]
```

Sets the objects velocity to zero.

Parameters

others	List of other space objects.
gravitational_constant	Gravitational constant.
delta_time	Time step for the update.

Implements IMovementStrategy.

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

- movement/static.hpp
- C:/Users/ugniu/code/cpp-2025/src/core/movement_static.cpp

Chapter 5

File Documentation

5.1 exception.hpp

```
#ifndef EXCEPTION_HPP
#define EXCEPTION_HPP

#include <stdexcept>
#include <string>

class SpaceObjectException : public std::runtime_error
{
public:
    explicit SpaceObjectException(const std::string &message)
        : std::runtime_error(message) {}
};

#endif // EXCEPTION_HPP
```

5.2 dynamic.hpp

```
#ifndef DYNAMIC_HPP
#define DYNAMIC_HPP

#include "i_movement_strategy.hpp"

class DynamicMovementStrategy : public IMovementStrategy
{
public:
    DynamicMovementStrategy(double *mass_ptr, double *radius_ptr, sf::Vector2f *position_ptr,
sf::Vector2f *velocity_ptr)
        : IMovementStrategy(mass_ptr, radius_ptr, position_ptr, velocity_ptr) {}

    void update_velocity(const std::vector<const SpaceObject *> &others,
                         const float gravitational_constant, const float delta_time) override;

    void update_position(const float delta_time) override;
};

#endif // DYNAMIC_HPP
```

5.3 i_movement_strategy.hpp

```
#ifndef I_MOVEMENT_STRATEGY_HPP
#define I_MOVEMENT_STRATEGY_HPP

#include "space_object.hpp"

class SpaceObject;

class IMovementStrategy
{
```

```cpp
protected:
    double *mass;
    double *radius;
    sf::Vector2f *position;
    sf::Vector2f *velocity;

public:
    IMovementStrategy(double *mass_ptr, double *radius_ptr, sf::Vector2f *position_ptr, sf::Vector2f *velocity_ptr)
        : mass(mass_ptr), radius(radius_ptr), position(position_ptr), velocity(velocity_ptr) {}

    virtual ~IMovementStrategy() = default;

    virtual void update_velocity(const std::vector<const SpaceObject *> &others,
                                 const float gravitational_constant, const float delta_time) = 0;

    virtual void update_position(const float delta_time) = 0;
};

#endif // I_MOVEMENT_STRATEGY_HPP
```

5.4 static.hpp

```cpp
#ifndef STATIC_HPP
#define STATIC_HPP

#include "i_movement_strategy.hpp"

class StaticMovementStrategy : public IMovementStrategy
{
public:
    StaticMovementStrategy(double *mass_ptr, double *radius_ptr, sf::Vector2f *position_ptr, sf::Vector2f *velocity_ptr)
        : IMovementStrategy(mass_ptr, radius_ptr, position_ptr, velocity_ptr) {}

    void update_velocity(const std::vector<const SpaceObject *> &others,
                         const float gravitational_constant, const float delta_time) override;

    void update_position(const float delta_time) override;
};

#endif // STATIC_HPP
```

5.5 space_object.hpp

```cpp
#ifndef SPACE_OBJECT_HPP
#define SPACE_OBJECT_HPP

#include <string>
#include <iostream>
#include <fstream>
#include <vector>
#include <SFML/System.hpp>

#include "exception.hpp"

class SpaceObject
{
private:
    // Pointer to an implementation of SpaceObject
    class SpaceObjectImpl;
    SpaceObjectImpl *impl;

public:
    // Constructors and Deconstructor

    SpaceObject();

    SpaceObject(std::string name, double mass, double radius, sf::Vector2f position, sf::Vector2f velocity = {0, 0}, bool is_movable = true);

    SpaceObject(const SpaceObject &other);

    ~SpaceObject();

    // Operators

    SpaceObject &operator=(const SpaceObject &other);

    friend std::ofstream &operator«(std::ofstream &out, const SpaceObject &obj);

    friend std::ifstream &operator»(std::ifstream &in, SpaceObject &obj);

    // Methods

    void print_info(std::ostream &output) const;

    void update_velocity(std::vector<const SpaceObject *> &others, const float gravitational_constant,
const float delta_time);

    void update_position(const float delta_time);

    // Getters

    static int get_object_count();

    int get_id() const;

    std::string get_name() const;

    double get_mass() const;

    double get_radius() const;

    sf::Vector2f get_velocity() const;

    sf::Vector2f get_position() const;

    bool is_movable() const;

    // Setters

    void set_name(std::string name);

    void set_mass(double mass);

    void set_radius(double radius);

    void set_position(sf::Vector2f position);

    void set_velocity(sf::Vector2f velocity);

    void set_movability(bool movable);
};

#endif // SPACE_OBJECT_HPP
```