

Sky Wars

Generated by Doxygen 1.9.0

1 Hierarchical Index	1
1.1 Class Hierarchy	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Class Documentation	7
4.1 Big_Plane Class Reference	7
4.1.1 Constructor & Destructor Documentation	8
4.1.1.1 Big_Plane()	8
4.1.1.2 ~Big_Plane()	8
4.1.2 Member Function Documentation	8
4.1.2.1 freeze()	8
4.1.2.2 get_type()	8
4.1.2.3 tick()	8
4.1.3 Member Data Documentation	9
4.1.3.1 freeze_state	9
4.1.3.2 shoot_speed	9
4.1.3.3 shoot_time	9
4.1.3.4 shoot_timer	9
4.2 Bomb Class Reference	9
4.2.1 Constructor & Destructor Documentation	10
4.2.1.1 Bomb()	10
4.2.1.2 ~Bomb()	10
4.2.2 Member Function Documentation	10
4.2.2.1 get_type()	10
4.3 Bullet Class Reference	11
4.3.1 Constructor & Destructor Documentation	11
4.3.1.1 Bullet()	11
4.3.1.2 ~Bullet()	11
4.4 End_State Class Reference	12
4.4.1 Constructor & Destructor Documentation	12
4.4.1.1 End_State()	12
4.4.2 Member Function Documentation	12
4.4.2.1 render()	13
4.4.3 Member Data Documentation	13
4.4.3.1 rating	13
4.4.3.2 score_info	13
4.4.3.3 status_text	13
4.4.3.4 texture	13

4.5 Enemy Struct Reference	14
4.5.1 Constructor & Destructor Documentation	14
4.5.1.1 Enemy()	14
4.5.1.2 ~Enemy()	14
4.5.2 Member Function Documentation	14
4.5.2.1 collision()	15
4.6 Enemy_Bullet Class Reference	15
4.6.1 Constructor & Destructor Documentation	15
4.6.1.1 Enemy_Bullet()	15
4.6.1.2 ~Enemy_Bullet()	16
4.6.2 Member Function Documentation	16
4.6.2.1 collision()	16
4.6.2.2 get_type()	16
4.7 Entity Class Reference	16
4.7.1 Constructor & Destructor Documentation	17
4.7.1.1 Entity() [1/2]	17
4.7.1.2 Entity() [2/2]	17
4.7.1.3 ~Entity()	17
4.7.2 Member Function Documentation	18
4.7.2.1 collision()	18
4.7.2.2 freeze()	18
4.7.2.3 get_health()	18
4.7.2.4 get_sprite()	18
4.7.2.5 get_type()	18
4.7.2.6 kill_me()	18
4.7.2.7 render()	19
4.7.2.8 tick()	19
4.7.3 Member Data Documentation	19
4.7.3.1 health	19
4.7.3.2 height	19
4.7.3.3 location	19
4.7.3.4 speed	19
4.7.3.5 sprite	20
4.7.3.6 width	20
4.8 Menu_State::Entry Struct Reference	20
4.8.1 Member Data Documentation	20
4.8.1.1 action	20
4.8.1.2 state	20
4.8.1.3 text	21
4.9 Exit_State Class Reference	21
4.9.1 Member Function Documentation	21
4.9.1.1 render()	21

4.9.1.2 tick()	22
4.10 Game_State Class Reference	22
4.10.1 Constructor & Destructor Documentation	22
4.10.1.1 Game_State() [1/2]	22
4.10.1.2 Game_State() [2/2]	23
4.10.2 Member Function Documentation	23
4.10.2.1 render()	23
4.10.2.2 tick()	23
4.10.3 Member Data Documentation	23
4.10.3.1 level_path	23
4.10.3.2 world	23
4.11 Help_State Class Reference	24
4.11.1 Constructor & Destructor Documentation	24
4.11.1.1 Help_State()	24
4.11.2 Member Function Documentation	24
4.11.2.1 render()	24
4.11.3 Member Data Documentation	25
4.11.3.1 keyboard_guide	25
4.11.3.2 texture	25
4.12 Level Class Reference	25
4.12.1 Constructor & Destructor Documentation	25
4.12.1.1 Level()	25
4.12.1.2 ~Level()	26
4.12.2 Member Function Documentation	26
4.12.2.1 load_enemy()	26
4.12.2.2 max_score()	26
4.12.3 Member Data Documentation	26
4.12.3.1 data	26
4.13 Level_Selection_State Class Reference	26
4.13.1 Constructor & Destructor Documentation	27
4.13.1.1 Level_Selection_State()	27
4.14 Main_Menu_State Class Reference	27
4.14.1 Constructor & Destructor Documentation	27
4.14.1.1 Main_Menu_State()	28
4.15 Menu_State Class Reference	28
4.15.1 Member Typedef Documentation	29
4.15.1.1 Action	29
4.15.2 Constructor & Destructor Documentation	29
4.15.2.1 Menu_State()	29
4.15.3 Member Function Documentation	29
4.15.3.1 add()	29
4.15.3.2 on_key_press()	29

4.15.3.3 render()	29
4.15.3.4 tick()	30
4.15.4 Member Data Documentation	30
4.15.4.1 background	30
4.15.4.2 delay	30
4.15.4.3 enter_pressed	30
4.15.4.4 entries	30
4.15.4.5 font	30
4.15.4.6 selected	31
4.16 Pause_State Class Reference	31
4.16.1 Constructor & Destructor Documentation	31
4.16.1.1 Pause_State()	31
4.17 Player Class Reference	32
4.17.1 Constructor & Destructor Documentation	33
4.17.1.1 Player()	33
4.17.1.2 ~Player()	33
4.17.2 Member Function Documentation	33
4.17.2.1 cheat()	33
4.17.2.2 collision()	33
4.17.2.3 freeze()	33
4.17.2.4 get_shield_time()	34
4.17.2.5 get_type()	34
4.17.2.6 has_shield()	34
4.17.2.7 is_invincible()	34
4.17.2.8 set_position()	34
4.17.2.9 shoot()	34
4.17.2.10 tick()	34
4.17.2.11 update_status()	35
4.17.3 Member Data Documentation	35
4.17.3.1 freeze_state	35
4.17.3.2 give_invincible	35
4.17.3.3 invincibility_time	35
4.17.3.4 invincibility_timer	35
4.17.3.5 invincible	35
4.17.3.6 shield	35
4.17.3.7 shield_time	36
4.17.3.8 shield_timer	36
4.17.3.9 shoot_speed	36
4.17.3.10 shoot_time	36
4.17.3.11 shoot_timer	36
4.17.3.12 triple_time	36
4.17.3.13 triple_timer	36

4.17.3.14 tripleshot	37
4.18 Player_Bullet Class Reference	37
4.18.1 Constructor & Destructor Documentation	37
4.18.1.1 Player_Bullet()	37
4.18.1.2 ~Player_Bullet()	38
4.18.2 Member Function Documentation	38
4.18.2.1 collision()	38
4.18.2.2 get_type()	38
4.18.2.3 tick()	38
4.19 Powerup Class Reference	38
4.19.1 Constructor & Destructor Documentation	39
4.19.1.1 Powerup()	39
4.19.1.2 ~Powerup()	39
4.19.2 Member Function Documentation	39
4.19.2.1 collision()	39
4.19.2.2 get_type()	40
4.19.3 Member Data Documentation	40
4.19.3.1 name	40
4.20 Small_Plane Class Reference	40
4.20.1 Constructor & Destructor Documentation	41
4.20.1.1 Small_Plane()	41
4.20.1.2 ~Small_Plane()	41
4.20.2 Member Function Documentation	41
4.20.2.1 freeze()	41
4.20.2.2 get_type()	41
4.20.2.3 tick()	42
4.20.2.4 vertical_move()	42
4.20.3 Member Data Documentation	42
4.20.3.1 freeze_state	42
4.20.3.2 shoot_speed	42
4.20.3.3 shoot_time	42
4.20.3.4 shoot_timer	42
4.20.3.5 upp_state	43
4.20.3.6 vertical_time	43
4.20.3.7 vertical_timer	43
4.21 State Class Reference	43
4.21.1 Constructor & Destructor Documentation	44
4.21.1.1 State()	44
4.21.1.2 ~State()	44
4.21.2 Member Function Documentation	44
4.21.2.1 on_key_press()	44
4.21.2.2 on_key_release()	44

4.21.2.3 render()	44
4.21.2.4 run()	45
4.21.2.5 tick()	45
4.22 Texture_Manager Class Reference	45
4.22.1 Constructor & Destructor Documentation	45
4.22.1.1 Texture_Manager()	46
4.22.2 Member Function Documentation	46
4.22.2.1 get()	46
4.22.3 Member Data Documentation	46
4.22.3.1 instance	46
4.22.3.2 textures	46
4.23 Textured_object Class Reference	46
4.23.1 Constructor & Destructor Documentation	47
4.23.1.1 Textured_object()	47
4.23.2 Member Function Documentation	47
4.23.2.1 render()	47
4.23.2.2 set_texture()	47
4.24 World Class Reference	47
4.24.1 Constructor & Destructor Documentation	48
4.24.1.1 World()	48
4.24.1.2 ~World()	48
4.24.2 Member Function Documentation	49
4.24.2.1 add_score()	49
4.24.2.2 freeze()	49
4.24.2.3 insert_object()	49
4.24.2.4 manage_collision()	49
4.24.2.5 render()	49
4.24.2.6 tick()	49
4.24.3 Member Data Documentation	49
4.24.3.1 counter	50
4.24.3.2 font	50
4.24.3.3 freeze_state	50
4.24.3.4 game_time	50
4.24.3.5 game_timer	50
4.24.3.6 level	50
4.24.3.7 level_name	50
4.24.3.8 life_info	50
4.24.3.9 objects	51
4.24.3.10 player	51
4.24.3.11 score	51
4.24.3.12 score_info	51
4.24.3.13 shield_info	51

4.24.3.14 spawn_time	51
4.24.3.15 spawn_timer	51
5 File Documentation	53
5.1 /home/nils/TDP005/Game/Bullet.cpp File Reference	53
5.2 /home/nils/TDP005/Game/Bullet.h File Reference	53
5.3 /home/nils/TDP005/Game/common.h File Reference	53
5.4 /home/nils/TDP005/Game/Enemy.cpp File Reference	53
5.5 /home/nils/TDP005/Game/Enemy.h File Reference	54
5.6 /home/nils/TDP005/Game/Entity.cpp File Reference	54
5.7 /home/nils/TDP005/Game/Entity.h File Reference	54
5.8 /home/nils/TDP005/Game/game_state.cpp File Reference	54
5.9 /home/nils/TDP005/Game/game_state.h File Reference	54
5.10 /home/nils/TDP005/Game/Level.cpp File Reference	55
5.11 /home/nils/TDP005/Game/Level.h File Reference	55
5.12 /home/nils/TDP005/Game/main.cpp File Reference	55
5.12.1 Function Documentation	55
5.12.1.1 main()	55
5.12.2 Variable Documentation	56
5.12.2.1 height	56
5.12.2.2 width	56
5.13 /home/nils/TDP005/Game/menu_state.cpp File Reference	56
5.14 /home/nils/TDP005/Game/menu_state.h File Reference	56
5.15 /home/nils/TDP005/Game/Player.cpp File Reference	56
5.16 /home/nils/TDP005/Game/Player.h File Reference	57
5.17 /home/nils/TDP005/Game/Powerup.cpp File Reference	57
5.18 /home/nils/TDP005/Game/Powerup.h File Reference	57
5.19 /home/nils/TDP005/Game/state.cpp File Reference	57
5.20 /home/nils/TDP005/Game/state.h File Reference	57
5.21 /home/nils/TDP005/Game/Texture_manager.cpp File Reference	58
5.22 /home/nils/TDP005/Game/Texture_manager.h File Reference	58
5.23 /home/nils/TDP005/Game/World.cpp File Reference	58
5.24 /home/nils/TDP005/Game/World.h File Reference	58
Index	59

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

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

enable_shared_from_this	
State	43
Exit_State	21
Game_State	22
Menu_State	28
End_State	12
Help_State	24
Level_Selection_State	26
Main_Menu_State	27
Pause_State	31
Entity	16
Textured_object	46
Bullet	11
Enemy_Bullet	15
Player_Bullet	37
Enemy	14
Big_Plane	7
Bomb	9
Small_Plane	40
Player	32
Powerup	38
Menu_State::Entry	20
Level	25
Texture_Manager	45
World	47

Chapter 2

Class Index

2.1 Class List

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

Big_Plane	7
Bomb	9
Bullet	11
End_State	12
Enemy	14
Enemy_Bullet	15
Entity	16
Menu_State::Entry	20
Exit_State	21
Game_State	22
Help_State	24
Level	25
Level_Selection_State	26
Main_Menu_State	27
Menu_State	28
Pause_State	31
Player	32
Player_Bullet	37
Powerup	38
Small_Plane	40
State	43
Texture_Manager	45
Textured_object	46
World	47

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

/home/nils/TDP005/Game/Bullet.cpp	53
/home/nils/TDP005/Game/Bullet.h	53
/home/nils/TDP005/Game/common.h	53
/home/nils/TDP005/Game/Enemy.cpp	53
/home/nils/TDP005/Game/Enemy.h	54
/home/nils/TDP005/Game/Entity.cpp	54
/home/nils/TDP005/Game/Entity.h	54
/home/nils/TDP005/Game/game_state.cpp	54
/home/nils/TDP005/Game/game_state.h	54
/home/nils/TDP005/Game/Level.cpp	55
/home/nils/TDP005/Game/Level.h	55
/home/nils/TDP005/Game/main.cpp	55
/home/nils/TDP005/Game/menu_state.cpp	56
/home/nils/TDP005/Game/menu_state.h	56
/home/nils/TDP005/Game/Player.cpp	56
/home/nils/TDP005/Game/Player.h	57
/home/nils/TDP005/Game/Powerup.cpp	57
/home/nils/TDP005/Game/Powerup.h	57
/home/nils/TDP005/Game/state.cpp	57
/home/nils/TDP005/Game/state.h	57
/home/nils/TDP005/Game/Texture_manager.cpp	58
/home/nils/TDP005/Game/Texture_manager.h	58
/home/nils/TDP005/Game/World.cpp	58
/home/nils/TDP005/Game/World.h	58

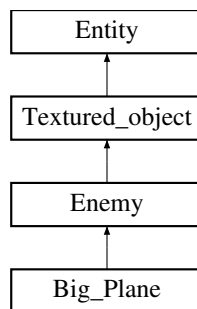
Chapter 4

Class Documentation

4.1 Big_Plane Class Reference

```
#include <Enemy.h>
```

Inheritance diagram for Big_Plane:



Public Member Functions

- [Big_Plane](#) (sf::Vector2f [location](#))
- [~Big_Plane](#) () override=default
- bool [tick](#) (sf::Time delta, [World](#) &world) override
- void [freeze](#) () override
- string [get_type](#) () override

Private Attributes

- float [shoot_speed](#)
- sf::Clock [shoot_timer](#)
- sf::Time [shoot_time](#)
- bool [freeze_state](#)

Additional Inherited Members

4.1.1 Constructor & Destructor Documentation

4.1.1.1 `Big_Plane()`

```
Big_Plane::Big_Plane (
    sf::Vector2f location ) [explicit]
```

4.1.1.2 `~Big_Plane()`

```
Big_Plane::~~Big_Plane ( ) [override], [default]
```

4.1.2 Member Function Documentation

4.1.2.1 `freeze()`

```
void Big_Plane::freeze ( ) [override], [virtual]
```

Reimplemented from [Entity](#).

4.1.2.2 `get_type()`

```
string Big_Plane::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

4.1.2.3 `tick()`

```
bool Big_Plane::tick (
    sf::Time delta,
    World & world ) [override], [virtual]
```

Reimplemented from [Entity](#).

4.1.3 Member Data Documentation

4.1.3.1 freeze_state

```
bool Big_Plane::freeze_state [private]
```

4.1.3.2 shoot_speed

```
float Big_Plane::shoot_speed [private]
```

4.1.3.3 shoot_time

```
sf::Time Big_Plane::shoot_time [private]
```

4.1.3.4 shoot_timer

```
sf::Clock Big_Plane::shoot_timer [private]
```

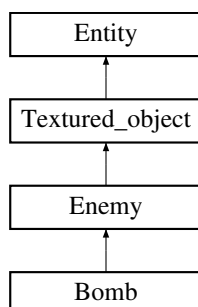
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/[Enemy.h](#)
- /home/nils/TDP005/Game/[Enemy.cpp](#)

4.2 Bomb Class Reference

```
#include <Enemy.h>
```

Inheritance diagram for Bomb:



Public Member Functions

- [Bomb](#) (sf::Vector2f [location](#))
- [~Bomb](#) () override=default
- string [get_type](#) () override

Additional Inherited Members

4.2.1 Constructor & Destructor Documentation

4.2.1.1 Bomb()

```
Bomb::Bomb (
    sf::Vector2f location ) [explicit]
```

4.2.1.2 ~Bomb()

```
Bomb::~Bomb ( ) [override], [default]
```

4.2.2 Member Function Documentation

4.2.2.1 get_type()

```
string Bomb::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

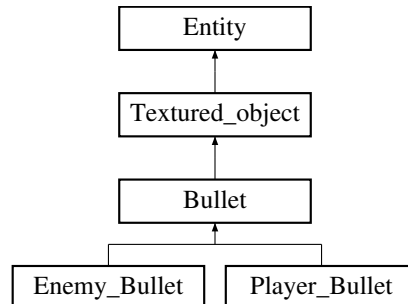
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/[Enemy.h](#)
- /home/nils/TDP005/Game/[Enemy.cpp](#)

4.3 Bullet Class Reference

```
#include <Bullet.h>
```

Inheritance diagram for Bullet:



Public Member Functions

- [Bullet](#) (sf::Vector2f [location](#), int bullet_width, int bullet_height, std::string const &texture_name)
- [~Bullet](#) () override=default

Additional Inherited Members

4.3.1 Constructor & Destructor Documentation

4.3.1.1 Bullet()

```
Bullet::Bullet (
    sf::Vector2f location,
    int bullet_width,
    int bullet_height,
    std::string const & texture_name ) [explicit]
```

4.3.1.2 ~Bullet()

```
Bullet::~Bullet ( ) [override], [default]
```

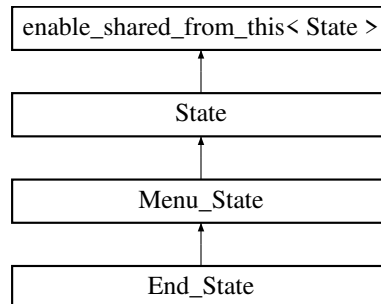
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/[Bullet.h](#)
- /home/nils/TDP005/Game/[Bullet.cpp](#)

4.4 End_State Class Reference

```
#include <menu_state.h>
```

Inheritance diagram for End_State:



Public Member Functions

- [End_State](#) (int player_health, int player_score, string const &level_name, int max_score)
- void [render](#) (sf::RenderWindow &window) override

Private Attributes

- sf::Text [status_text](#)
- sf::Texture [texture](#)
- sf::Sprite [rating](#)
- sf::Text [score_info](#)

Additional Inherited Members

4.4.1 Constructor & Destructor Documentation

4.4.1.1 End_State()

```
End_State::End_State (
    int player_health,
    int player_score,
    string const & level_name,
    int max_score )
```

4.4.2 Member Function Documentation

4.4.2.1 render()

```
void End_State::render (
    sf::RenderWindow & window ) [override], [virtual]
```

Reimplemented from [Menu_State](#).

4.4.3 Member Data Documentation

4.4.3.1 rating

```
sf::Sprite End_State::rating [private]
```

4.4.3.2 score_info

```
sf::Text End_State::score_info [private]
```

4.4.3.3 status_text

```
sf::Text End_State::status_text [private]
```

4.4.3.4 texture

```
sf::Texture End_State::texture [private]
```

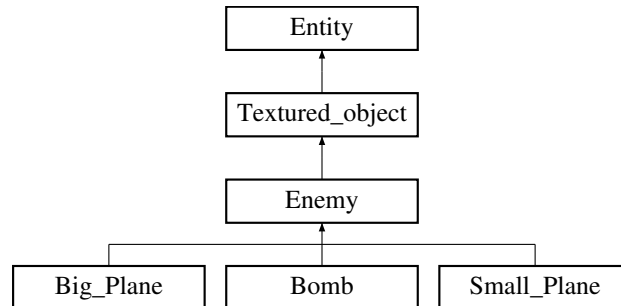
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/menu_state.h](#)
- [/home/nils/TDP005/Game/menu_state.cpp](#)

4.5 Enemy Struct Reference

```
#include <Enemy.h>
```

Inheritance diagram for Enemy:



Public Member Functions

- [Enemy](#) (sf::Vector2f [location](#), int [width](#), int [height](#), const string &texture_name)
- [~Enemy](#) () override=default
- void [collision](#) (vector< shared_ptr< [Enemy](#) >> const &objects, [World](#) &world) override

Additional Inherited Members

4.5.1 Constructor & Destructor Documentation

4.5.1.1 Enemy()

```
Enemy::Enemy (
    sf::Vector2f location,
    int width,
    int height,
    const string & texture_name ) [explicit]
```

4.5.1.2 ~Enemy()

```
Enemy::~Enemy ( ) [override], [default]
```

4.5.2 Member Function Documentation

4.5.2.1 collision()

```
void Enemy::collision (
    vector< shared_ptr< Entity >> const & objects,
    World & world ) [override], [virtual]
```

Implements [Entity](#).

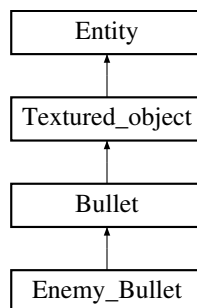
The documentation for this struct was generated from the following files:

- [/home/nils/TDP005/Game/Enemy.h](#)
- [/home/nils/TDP005/Game/Enemy.cpp](#)

4.6 Enemy_Bullet Class Reference

```
#include <Bullet.h>
```

Inheritance diagram for Enemy_Bullet:



Public Member Functions

- [Enemy_Bullet](#) (sf::Vector2f [location](#))
- [~Enemy_Bullet](#) () override=default
- void [collision](#) (vector< shared_ptr< [Entity](#) >> const &objects, [World](#) &world) override
- string [get_type](#) () override

Additional Inherited Members

4.6.1 Constructor & Destructor Documentation

4.6.1.1 Enemy_Bullet()

```
Enemy_Bullet::Enemy_Bullet (
    sf::Vector2f location ) [explicit]
```

4.6.1.2 ~Enemy_Bullet()

```
Enemy_Bullet::~Enemy_Bullet ( ) [override], [default]
```

4.6.2 Member Function Documentation

4.6.2.1 collision()

```
void Enemy_Bullet::collision (
    vector< shared_ptr< Entity >> const & objects,
    World & world ) [override], [virtual]
```

Implements [Entity](#).

4.6.2.2 get_type()

```
string Enemy_Bullet::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

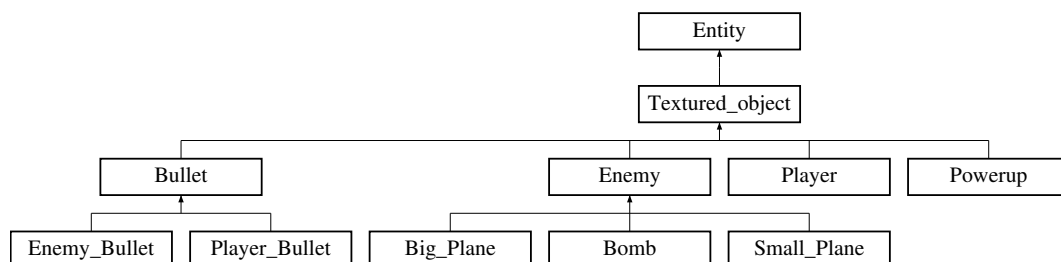
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/Bullet.h](#)
- [/home/nils/TDP005/Game/Bullet.cpp](#)

4.7 Entity Class Reference

```
#include <Entity.h>
```

Inheritance diagram for Entity:



Public Member Functions

- `Entity()`=delete
- `Entity(sf::Vector2f location, int width, int height)`
- `virtual ~Entity()`=default
- `virtual bool tick(sf::Time delta, World &world)`
- `virtual void render(sf::RenderWindow &window)=0`
- `bool kill_me() const`
- `virtual void collision(vector< shared_ptr< Entity >> const &objects, World &world)=0`
- `virtual void freeze()`
- `virtual std::string get_type()=0`
- `virtual sf::Sprite get_sprite() const`
- `int get_health() const`

Protected Attributes

- `sf::Vector2f location {}`
- `float speed {}`
- `int health {}`
- `int width {}`
- `int height {}`
- `sf::Sprite sprite {}`

4.7.1 Constructor & Destructor Documentation

4.7.1.1 Entity() [1/2]

```
Entity::Entity ( ) [delete]
```

4.7.1.2 Entity() [2/2]

```
Entity::Entity (
    sf::Vector2f location,
    int width,
    int height ) [explicit]
```

4.7.1.3 ~Entity()

```
virtual Entity::~~Entity ( ) [virtual], [default]
```

4.7.2 Member Function Documentation

4.7.2.1 collision()

```
virtual void Entity::collision (
    vector< shared_ptr< Entity >> const & objects,
    World & world ) [pure virtual]
```

Implemented in [Powerup](#), [Player](#), [Enemy](#), [Enemy_Bullet](#), and [Player_Bullet](#).

4.7.2.2 freeze()

```
virtual void Entity::freeze ( ) [inline], [virtual]
```

Reimplemented in [Player](#), [Big_Plane](#), and [Small_Plane](#).

4.7.2.3 get_health()

```
int Entity::get_health ( ) const
```

4.7.2.4 get_sprite()

```
virtual sf::Sprite Entity::get_sprite ( ) const [inline], [virtual]
```

4.7.2.5 get_type()

```
virtual std::string Entity::get_type ( ) [pure virtual]
```

Implemented in [Powerup](#), [Player](#), [Big_Plane](#), [Small_Plane](#), [Bomb](#), [Enemy_Bullet](#), and [Player_Bullet](#).

4.7.2.6 kill_me()

```
bool Entity::kill_me ( ) const
```

4.7.2.7 render()

```
virtual void Entity::render (
    sf::RenderWindow & window ) [pure virtual]
```

Implemented in [Textured_object](#).

4.7.2.8 tick()

```
bool Entity::tick (
    sf::Time delta,
    World & world ) [virtual]
```

Reimplemented in [Player](#), [Big_Plane](#), [Small_Plane](#), and [Player_Bullet](#).

4.7.3 Member Data Documentation

4.7.3.1 health

```
int Entity::health {} [protected]
```

4.7.3.2 height

```
int Entity::height {} [protected]
```

4.7.3.3 location

```
sf::Vector2f Entity::location {} [protected]
```

4.7.3.4 speed

```
float Entity::speed {} [protected]
```

4.7.3.5 sprite

```
sf::Sprite Entity::sprite {} [protected]
```

4.7.3.6 width

```
int Entity::width {} [protected]
```

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

- /home/nils/TDP005/Game/[Entity.h](#)
- /home/nils/TDP005/Game/[Entity.cpp](#)

4.8 Menu_State::Entry Struct Reference

```
#include <menu_state.h>
```

Public Attributes

- sf::Text [text](#)
- float [state](#)
- [Action](#) [action](#)

4.8.1 Member Data Documentation

4.8.1.1 action

```
Action Menu_State::Entry::action
```

4.8.1.2 state

```
float Menu_State::Entry::state
```

4.8.1.3 text

```
sf::Text Menu_State::Entry::text
```

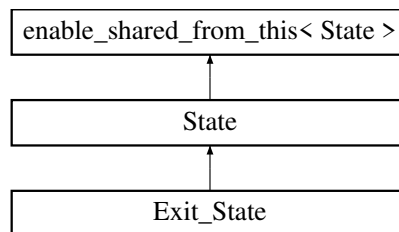
The documentation for this struct was generated from the following file:

- [/home/nils/TDP005/Game/menu_state.h](#)

4.9 Exit_State Class Reference

```
#include <state.h>
```

Inheritance diagram for Exit_State:



Public Member Functions

- `shared_ptr< State > tick` (sf::Time) override
- `void render` (sf::RenderWindow &) override

Additional Inherited Members

4.9.1 Member Function Documentation

4.9.1.1 render()

```
void Exit_State::render (
    sf::RenderWindow & ) [inline], [override], [virtual]
```

Implements [State](#).

4.9.1.2 tick()

```
shared_ptr<State> Exit_State::tick (
    sf::Time ) [inline], [override], [virtual]
```

Implements [State](#).

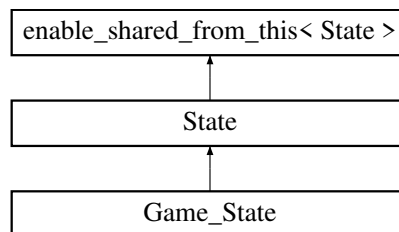
The documentation for this class was generated from the following file:

- [/home/nils/TDP005/Game/state.h](#)

4.10 Game_State Class Reference

```
#include <game_state.h>
```

Inheritance diagram for Game_State:



Public Member Functions

- [Game_State](#) ()=default
- [Game_State](#) (string const &[level_path](#))
- [shared_ptr< State > tick](#) (sf::Time delta) override
- void [render](#) (sf::RenderWindow &to) override

Private Attributes

- [World world](#)
- string const [level_path](#)

Additional Inherited Members

4.10.1 Constructor & Destructor Documentation

4.10.1.1 Game_State() [1/2]

```
Game_State::Game_State ( ) [default]
```


4.10.1.2 Game_State() [2/2]

```
Game_State::Game_State (
    string const & level_path )
```

4.10.2 Member Function Documentation

4.10.2.1 render()

```
void Game_State::render (
    sf::RenderWindow & to ) [override], [virtual]
```

Implements [State](#).

4.10.2.2 tick()

```
shared_ptr< State > Game_State::tick (
    sf::Time delta ) [override], [virtual]
```

Implements [State](#).

4.10.3 Member Data Documentation

4.10.3.1 level_path

```
string const Game_State::level_path [private]
```

4.10.3.2 world

```
World Game_State::world [private]
```

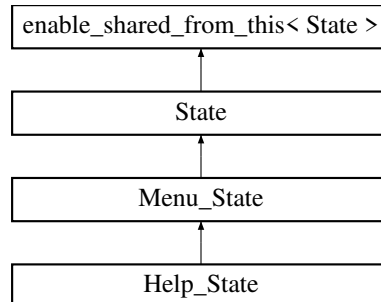
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/game_state.h](#)
- [/home/nils/TDP005/Game/game_state.cpp](#)

4.11 Help_State Class Reference

```
#include <menu_state.h>
```

Inheritance diagram for Help_State:



Public Member Functions

- [Help_State](#) ()
- void [render](#) (sf::RenderWindow &window) override

Private Attributes

- sf::Sprite [keyboard_guide](#)
- sf::Texture [texture](#)

Additional Inherited Members

4.11.1 Constructor & Destructor Documentation

4.11.1.1 Help_State()

```
Help_State::Help_State ( )
```

4.11.2 Member Function Documentation

4.11.2.1 render()

```
void Help_State::render (
    sf::RenderWindow & window ) [override], [virtual]
```

Reimplemented from [Menu_State](#).

4.11.3 Member Data Documentation

4.11.3.1 keyboard_guide

```
sf::Sprite Help_State::keyboard_guide [private]
```

4.11.3.2 texture

```
sf::Texture Help_State::texture [private]
```

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

- [/home/nils/TDP005/Game/menu_state.h](#)
- [/home/nils/TDP005/Game/menu_state.cpp](#)

4.12 Level Class Reference

```
#include <Level.h>
```

Public Member Functions

- [Level](#) (string const &file_name)
- [~Level](#) ()=default
- vector< shared_ptr< [Entity](#) > > [load_enemy](#) (int index)
- int [max_score](#) () const

Private Attributes

- vector< map< int, string > > [data](#) {}

4.12.1 Constructor & Destructor Documentation

4.12.1.1 Level()

```
Level::Level (  
    string const & file_name )
```

4.12.1.2 ~Level()

```
Level::~~Level ( ) [default]
```

4.12.2 Member Function Documentation

4.12.2.1 load_enemy()

```
vector< shared_ptr< Entity > > Level::load_enemy (
    int index )
```

4.12.2.2 max_score()

```
int Level::max_score ( ) const
```

4.12.3 Member Data Documentation

4.12.3.1 data

```
vector< map<int, string> > Level::data {} [private]
```

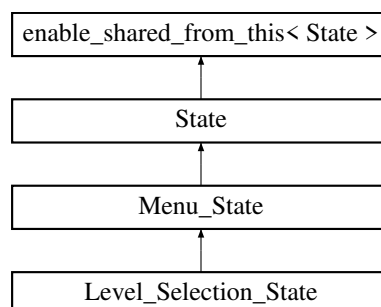
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/Level.h](#)
- [/home/nils/TDP005/Game/Level.cpp](#)

4.13 Level_Selection_State Class Reference

```
#include <menu_state.h>
```

Inheritance diagram for Level_Selection_State:



Public Member Functions

- [Level_Selection_State](#) ()

Additional Inherited Members

4.13.1 Constructor & Destructor Documentation

4.13.1.1 Level_Selection_State()

```
Level_Selection_State::Level_Selection_State ( )
```

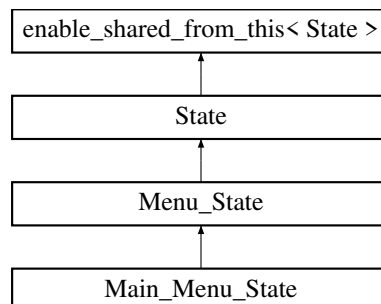
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/[menu_state.h](#)
- /home/nils/TDP005/Game/[menu_state.cpp](#)

4.14 Main_Menu_State Class Reference

```
#include <menu_state.h>
```

Inheritance diagram for Main_Menu_State:



Public Member Functions

- [Main_Menu_State](#) ()

Additional Inherited Members

4.14.1 Constructor & Destructor Documentation

4.14.1.1 Main_Menu_State()

```
Main_Menu_State::Main_Menu_State ( )
```

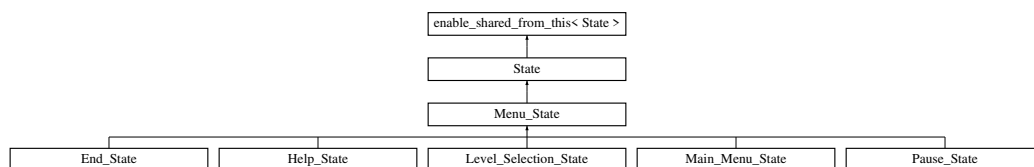
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/menu_state.h
- /home/nils/TDP005/Game/menu_state.cpp

4.15 Menu_State Class Reference

```
#include <menu_state.h>
```

Inheritance diagram for Menu_State:



Classes

- struct [Entry](#)

Public Member Functions

- [Menu_State](#) ()
- void [on_key_press](#) (sf::Keyboard::Key key) override
- shared_ptr< [State](#) > [tick](#) (sf::Time time) override
- void [render](#) (sf::RenderWindow &window) override

Protected Types

- using [Action](#) = function< shared_ptr< [State](#) >()>

Protected Member Functions

- void [add](#) (const string &text, [Action](#) action)

Protected Attributes

- sf::Font [font](#)
- vector< [Entry](#) > [entries](#)
- size_t [selected](#)
- bool [enter_pressed](#)
- sf::Time [delay](#)
- shared_ptr< [State](#) > [background](#)

Additional Inherited Members

4.15.1 Member Typedef Documentation

4.15.1.1 Action

```
using Menu_State::Action = function<shared_ptr<State>()> [protected]
```

4.15.2 Constructor & Destructor Documentation

4.15.2.1 Menu_State()

```
Menu_State::Menu_State ( )
```

4.15.3 Member Function Documentation

4.15.3.1 add()

```
void Menu_State::add (
    const string & text,
    Action action ) [protected]
```

4.15.3.2 on_key_press()

```
void Menu_State::on_key_press (
    sf::Keyboard::Key key ) [override], [virtual]
```

Reimplemented from [State](#).

4.15.3.3 render()

```
void Menu_State::render (
    sf::RenderWindow & window ) [override], [virtual]
```

Implements [State](#).

Reimplemented in [End_State](#), and [Help_State](#).

4.15.3.4 tick()

```
shared_ptr< State > Menu_State::tick (
    sf::Time time ) [override], [virtual]
```

Implements [State](#).

4.15.4 Member Data Documentation

4.15.4.1 background

```
shared_ptr<State> Menu_State::background [protected]
```

4.15.4.2 delay

```
sf::Time Menu_State::delay [protected]
```

4.15.4.3 enter_pressed

```
bool Menu_State::enter_pressed [protected]
```

4.15.4.4 entries

```
vector<Entry> Menu_State::entries [protected]
```

4.15.4.5 font

```
sf::Font Menu_State::font [protected]
```


4.15.4.6 selected

```
size_t Menu_State::selected [protected]
```

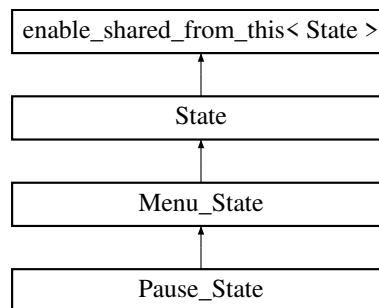
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/menu_state.h
- /home/nils/TDP005/Game/menu_state.cpp

4.16 Pause_State Class Reference

```
#include <menu_state.h>
```

Inheritance diagram for Pause_State:



Public Member Functions

- [Pause_State](#) (shared_ptr< [State](#) > resume, string current_level)

Additional Inherited Members

4.16.1 Constructor & Destructor Documentation

4.16.1.1 Pause_State()

```
Pause_State::Pause_State (
    shared_ptr< State > resume,
    string current_level )
```

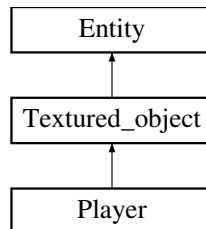
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/menu_state.h
- /home/nils/TDP005/Game/menu_state.cpp

4.17 Player Class Reference

```
#include <Player.h>
```

Inheritance diagram for Player:



Public Member Functions

- [Player](#) (sf::Vector2f [location](#))
- [~Player](#) () override=default
- bool [tick](#) (sf::Time delta, [World](#) &world) override
- void [collision](#) (vector< shared_ptr< [Entity](#) >> const &objects, [World](#) &world) override
- void [freeze](#) () override
- string [get_type](#) () override
- string [get_shield_time](#) () const
- bool [has_shield](#) () const

Static Public Member Functions

- static bool [is_invincible](#) ()

Private Member Functions

- void [cheat](#) ()
- void [set_position](#) (sf::Time delta)
- void [update_status](#) ()
- void [shoot](#) ([World](#) &world)

Private Attributes

- float [shoot_speed](#)
- sf::Clock [shoot_timer](#)
- sf::Time [shoot_time](#)
- bool [shield](#)
- sf::Clock [shield_timer](#)
- sf::Time [shield_time](#)
- bool [tripleshot](#)
- sf::Clock [triple_timer](#)
- sf::Time [triple_time](#)
- sf::Clock [invincibility_timer](#)
- sf::Time [invincibility_time](#)
- bool [give_invincible](#)
- bool [freeze_state](#)

Static Private Attributes

- static bool [invincible](#)

Additional Inherited Members

4.17.1 Constructor & Destructor Documentation

4.17.1.1 Player()

```
Player::Player (
    sf::Vector2f location ) [explicit]
```

4.17.1.2 ~Player()

```
Player::~Player ( ) [override], [default]
```

4.17.2 Member Function Documentation

4.17.2.1 cheat()

```
void Player::cheat ( ) [private]
```

4.17.2.2 collision()

```
void Player::collision (
    vector< shared_ptr< Entity >> const & objects,
    World & world ) [override], [virtual]
```

Implements [Entity](#).

4.17.2.3 freeze()

```
void Player::freeze ( ) [override], [virtual]
```

Reimplemented from [Entity](#).

4.17.2.4 get_shield_time()

```
string Player::get_shield_time ( ) const
```

4.17.2.5 get_type()

```
string Player::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

4.17.2.6 has_shield()

```
bool Player::has_shield ( ) const [inline]
```

4.17.2.7 is_invincible()

```
static bool Player::is_invincible ( ) [inline], [static]
```

4.17.2.8 set_position()

```
void Player::set_position (
    sf::Time delta ) [private]
```

4.17.2.9 shoot()

```
void Player::shoot (
    World & world ) [private]
```

4.17.2.10 tick()

```
bool Player::tick (
    sf::Time delta,
    World & world ) [override], [virtual]
```

Reimplemented from [Entity](#).

4.17.2.11 update_status()

```
void Player::update_status ( ) [private]
```

4.17.3 Member Data Documentation

4.17.3.1 freeze_state

```
bool Player::freeze_state [private]
```

4.17.3.2 give_invincible

```
bool Player::give_invincible [private]
```

4.17.3.3 invincibility_time

```
sf::Time Player::invincibility_time [private]
```

4.17.3.4 invincibility_timer

```
sf::Clock Player::invincibility_timer [private]
```

4.17.3.5 invincible

```
bool Player::invincible [static], [private]
```

4.17.3.6 shield

```
bool Player::shield [private]
```

4.17.3.7 shield_time

```
sf::Time Player::shield_time [private]
```

4.17.3.8 shield_timer

```
sf::Clock Player::shield_timer [private]
```

4.17.3.9 shoot_speed

```
float Player::shoot_speed [private]
```

4.17.3.10 shoot_time

```
sf::Time Player::shoot_time [private]
```

4.17.3.11 shoot_timer

```
sf::Clock Player::shoot_timer [private]
```

4.17.3.12 triple_time

```
sf::Time Player::triple_time [private]
```

4.17.3.13 triple_timer

```
sf::Clock Player::triple_timer [private]
```

4.17.3.14 tripleshot

```
bool Player::tripleshot [private]
```

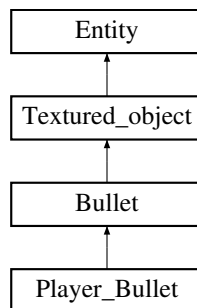
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/Player.h](#)
- [/home/nils/TDP005/Game/Player.cpp](#)

4.18 Player_Bullet Class Reference

```
#include <Bullet.h>
```

Inheritance diagram for Player_Bullet:



Public Member Functions

- [Player_Bullet](#) (sf::Vector2f [location](#))
- [~Player_Bullet](#) () override=default
- bool [tick](#) (sf::Time delta, [World](#) &world) override
- void [collision](#) (vector< shared_ptr< [Entity](#) >> const &objects, [World](#) &world) override
- string [get_type](#) () override

Additional Inherited Members

4.18.1 Constructor & Destructor Documentation

4.18.1.1 Player_Bullet()

```
Player_Bullet::Player_Bullet (
    sf::Vector2f location ) [explicit]
```

4.18.1.2 ~Player_Bullet()

```
Player_Bullet::~~Player_Bullet ( ) [override], [default]
```

4.18.2 Member Function Documentation

4.18.2.1 collision()

```
void Player_Bullet::collision (
    vector< shared_ptr< Entity >> const & objects,
    World & world ) [override], [virtual]
```

Implements [Entity](#).

4.18.2.2 get_type()

```
string Player_Bullet::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

4.18.2.3 tick()

```
bool Player_Bullet::tick (
    sf::Time delta,
    World & world ) [override], [virtual]
```

Reimplemented from [Entity](#).

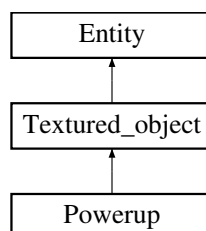
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/Bullet.h](#)
- [/home/nils/TDP005/Game/Bullet.cpp](#)

4.19 Powerup Class Reference

```
#include <Powerup.h>
```

Inheritance diagram for Powerup:



Public Member Functions

- [Powerup](#) (string *name*, string const &texture_name, sf::Vector2f *location*)
- [~Powerup](#) () override=default
- void [collision](#) (vector< shared_ptr< [Entity](#) >> const &objects, [World](#) &world) override
- string [get_type](#) () override

Private Attributes

- string const [name](#)

Additional Inherited Members

4.19.1 Constructor & Destructor Documentation

4.19.1.1 Powerup()

```
Powerup::Powerup (
    string name,
    string const & texture_name,
    sf::Vector2f location )
```

4.19.1.2 ~Powerup()

```
Powerup::~Powerup ( ) [override], [default]
```

4.19.2 Member Function Documentation

4.19.2.1 collision()

```
void Powerup::collision (
    vector< shared_ptr< Entity >> const & objects,
    World & world ) [override], [virtual]
```

Implements [Entity](#).

4.19.2.2 get_type()

```
string Powerup::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

4.19.3 Member Data Documentation

4.19.3.1 name

```
string const Powerup::name [private]
```

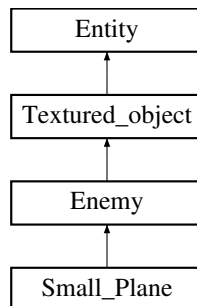
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/Powerup.h](#)
- [/home/nils/TDP005/Game/Powerup.cpp](#)

4.20 Small_Plane Class Reference

```
#include <Enemy.h>
```

Inheritance diagram for Small_Plane:



Public Member Functions

- [Small_Plane](#) (sf::Vector2f location)
- [~Small_Plane](#) () override=default
- bool [tick](#) (sf::Time delta, [World](#) &world) override
- void [freeze](#) () override
- string [get_type](#) () override

Private Member Functions

- void [vertical_move](#) (sf::Time delta)

Private Attributes

- float [shoot_speed](#)
- sf::Clock [shoot_timer](#)
- sf::Time [shoot_time](#)
- bool [upp_state](#)
- sf::Clock [vertical_timer](#)
- sf::Time [vertical_time](#)
- bool [freeze_state](#)

Additional Inherited Members

4.20.1 Constructor & Destructor Documentation

4.20.1.1 Small_Plane()

```
Small_Plane::Small_Plane (
    sf::Vector2f location ) [explicit]
```

4.20.1.2 ~Small_Plane()

```
Small_Plane::~~Small_Plane ( ) [override], [default]
```

4.20.2 Member Function Documentation

4.20.2.1 freeze()

```
void Small_Plane::freeze ( ) [override], [virtual]
```

Reimplemented from [Entity](#).

4.20.2.2 get_type()

```
string Small_Plane::get_type ( ) [inline], [override], [virtual]
```

Implements [Entity](#).

4.20.2.3 tick()

```
bool Small_Plane::tick (
    sf::Time delta,
    World & world ) [override], [virtual]
```

Reimplemented from [Entity](#).

4.20.2.4 vertical_move()

```
void Small_Plane::vertical_move (
    sf::Time delta ) [private]
```

4.20.3 Member Data Documentation

4.20.3.1 freeze_state

```
bool Small_Plane::freeze_state [private]
```

4.20.3.2 shoot_speed

```
float Small_Plane::shoot_speed [private]
```

4.20.3.3 shoot_time

```
sf::Time Small_Plane::shoot_time [private]
```

4.20.3.4 shoot_timer

```
sf::Clock Small_Plane::shoot_timer [private]
```

4.20.3.5 upp_state

```
bool Small_Plane::upp_state [private]
```

4.20.3.6 vertical_time

```
sf::Time Small_Plane::vertical_time [private]
```

4.20.3.7 vertical_timer

```
sf::Clock Small_Plane::vertical_timer [private]
```

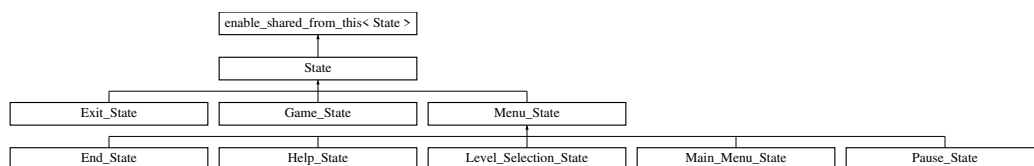
The documentation for this class was generated from the following files:

- /home/nils/TDP005/Game/[Enemy.h](#)
- /home/nils/TDP005/Game/[Enemy.cpp](#)

4.21 State Class Reference

```
#include <state.h>
```

Inheritance diagram for State:



Public Member Functions

- [State](#) ()=default
- virtual [~State](#) ()=default
- virtual void [on_key_press](#) (sf::Keyboard::Key key)
- virtual void [on_key_release](#) (sf::Keyboard::Key key)
- virtual shared_ptr< [State](#) > [tick](#) (sf::Time time)=0
- virtual void [render](#) (sf::RenderWindow &to)=0

Static Public Member Functions

- static void [run](#) (sf::RenderWindow &window, shared_ptr< [State](#) > state)

4.21.1 Constructor & Destructor Documentation

4.21.1.1 State()

```
State::State ( ) [default]
```

4.21.1.2 ~State()

```
virtual State::~~State ( ) [virtual], [default]
```

4.21.2 Member Function Documentation

4.21.2.1 on_key_press()

```
void State::on_key_press (
    sf::Keyboard::Key key ) [virtual]
```

Reimplemented in [Menu_State](#).

4.21.2.2 on_key_release()

```
void State::on_key_release (
    sf::Keyboard::Key key ) [virtual]
```

4.21.2.3 render()

```
virtual void State::render (
    sf::RenderWindow & to ) [pure virtual]
```

Implemented in [End_State](#), [Help_State](#), [Menu_State](#), [Game_State](#), and [Exit_State](#).

4.21.2.4 run()

```
void State::run (
    sf::RenderWindow & window,
    shared_ptr< State > state ) [static]
```

4.21.2.5 tick()

```
virtual shared_ptr<State> State::tick (
    sf::Time time ) [pure virtual]
```

Implemented in [Exit_State](#), [Menu_State](#), and [Game_State](#).

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

- [/home/nils/TDP005/Game/state.h](#)
- [/home/nils/TDP005/Game/state.cpp](#)

4.22 Texture_Manager Class Reference

```
#include <Texture_manager.h>
```

Static Public Member Functions

- static sf::Texture * [get](#) (const std::string &name, int [width](#), int [height](#))

Private Member Functions

- [Texture_Manager](#) ()=default

Private Attributes

- std::map< std::string, std::unique_ptr< sf::Texture > > [textures](#)

Static Private Attributes

- static [Texture_Manager instance](#)

4.22.1 Constructor & Destructor Documentation

4.22.1.1 Texture_Manager()

```
Texture_Manager::Texture_Manager ( ) [private], [default]
```

4.22.2 Member Function Documentation

4.22.2.1 get()

```
sf::Texture * Texture_Manager::get (
    const std::string & name,
    int width,
    int height ) [static]
```

4.22.3 Member Data Documentation

4.22.3.1 instance

```
Texture_Manager Texture_Manager::instance [static], [private]
```

4.22.3.2 textures

```
std::map<std::string, std::unique_ptr<sf::Texture> > Texture_Manager::textures [private]
```

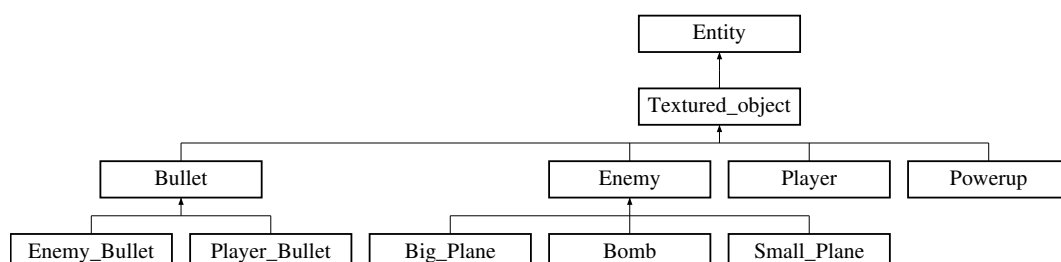
The documentation for this class was generated from the following files:

- [/home/nils/TDP005/Game/Texture_manager.h](#)
- [/home/nils/TDP005/Game/Texture_manager.cpp](#)

4.23 Textured_object Class Reference

```
#include <Entity.h>
```

Inheritance diagram for Textured_object:



Public Member Functions

- [Textured_object](#) (sf::Vector2f [location](#), int [width](#), int [height](#), std::string const &texture_name)
- void [render](#) (sf::RenderWindow &window) override
- void [set_texture](#) (string const &texture_name)

Additional Inherited Members

4.23.1 Constructor & Destructor Documentation

4.23.1.1 Textured_object()

```
Textured_object::Textured_object (
    sf::Vector2f location,
    int width,
    int height,
    std::string const & texture_name )
```

4.23.2 Member Function Documentation

4.23.2.1 render()

```
void Textured_object::render (
    sf::RenderWindow & window ) [override], [virtual]
```

Implements [Entity](#).

4.23.2.2 set_texture()

```
void Textured_object::set_texture (
    string const & texture_name )
```

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

- /home/nils/TDP005/Game/[Entity.h](#)
- /home/nils/TDP005/Game/[Entity.cpp](#)

4.24 World Class Reference

```
#include <World.h>
```

Public Member Functions

- [World](#) (string [level](#))
- [~World](#) ()=default
- void [insert_object](#) (shared_ptr< [Entity](#) > const &object)
- shared_ptr< [State](#) > [tick](#) (sf::Time delta)
- void [render](#) (sf::RenderWindow &window)
- void [add_score](#) (int num)
- void [freeze](#) ()

Private Member Functions

- void [manage_collision](#) ()

Private Attributes

- vector< shared_ptr< [Entity](#) > > [objects](#) {}
- shared_ptr< [Player](#) > [player](#) {}
- int [score](#)
- int [counter](#)
- sf::Font [font](#)
- sf::Text [life_info](#)
- sf::Text [shield_info](#)
- sf::Text [score_info](#)
- sf::Clock [spawn_timer](#)
- sf::Time [spawn_time](#)
- sf::Clock [game_timer](#)
- sf::Time [game_time](#)
- [Level](#) [level](#)
- string [level_name](#)
- bool [freeze_state](#)

4.24.1 Constructor & Destructor Documentation

4.24.1.1 [World\(\)](#)

```
World::World (
    string level )
```

4.24.1.2 [~World\(\)](#)

```
World::~~World ( ) [default]
```

4.24.2 Member Function Documentation

4.24.2.1 add_score()

```
void World::add_score (
    int num )
```

4.24.2.2 freeze()

```
void World::freeze ( )
```

4.24.2.3 insert_object()

```
void World::insert_object (
    shared_ptr< Entity > const & object )
```

4.24.2.4 manage_collision()

```
void World::manage_collision ( ) [private]
```

4.24.2.5 render()

```
void World::render (
    sf::RenderWindow & window )
```

4.24.2.6 tick()

```
shared_ptr< State > World::tick (
    sf::Time delta )
```

4.24.3 Member Data Documentation

4.24.3.1 counter

```
int World::counter [private]
```

4.24.3.2 font

```
sf::Font World::font [private]
```

4.24.3.3 freeze_state

```
bool World::freeze_state [private]
```

4.24.3.4 game_time

```
sf::Time World::game_time [private]
```

4.24.3.5 game_timer

```
sf::Clock World::game_timer [private]
```

4.24.3.6 level

```
Level World::level [private]
```

4.24.3.7 level_name

```
string World::level_name [private]
```

4.24.3.8 life_info

```
sf::Text World::life_info [private]
```

4.24.3.9 objects

```
vector<shared_ptr <Entity> > World::objects {} [private]
```

4.24.3.10 player

```
shared_ptr<Player> World::player {} [private]
```

4.24.3.11 score

```
int World::score [private]
```

4.24.3.12 score_info

```
sf::Text World::score_info [private]
```

4.24.3.13 shield_info

```
sf::Text World::shield_info [private]
```

4.24.3.14 spawn_time

```
sf::Time World::spawn_time [private]
```

4.24.3.15 spawn_timer

```
sf::Clock World::spawn_timer [private]
```

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

- /home/nils/TDP005/Game/[World.h](#)
- /home/nils/TDP005/Game/[World.cpp](#)

Chapter 5

File Documentation

5.1 /home/nils/TDP005/Game/Bullet.cpp File Reference

```
#include "Bullet.h"
```

5.2 /home/nils/TDP005/Game/Bullet.h File Reference

```
#include "common.h"  
#include "Entity.h"
```

Classes

- class [Bullet](#)
- class [Player_Bullet](#)
- class [Enemy_Bullet](#)

5.3 /home/nils/TDP005/Game/common.h File Reference

```
#include <SFML/Graphics.hpp>  
#include <iostream>  
#include <memory>
```

5.4 /home/nils/TDP005/Game/Enemy.cpp File Reference

```
#include "Enemy.h"  
#include "Bullet.h"  
#include "World.h"  
#include "Player.h"
```

5.5 /home/nils/TDP005/Game/Enemy.h File Reference

```
#include "common.h"
#include "Entity.h"
```

Classes

- struct [Enemy](#)
- class [Bomb](#)
- class [Small_Plane](#)
- class [Big_Plane](#)

5.6 /home/nils/TDP005/Game/Entity.cpp File Reference

```
#include "Entity.h"
#include "Texture_manager.h"
```

5.7 /home/nils/TDP005/Game/Entity.h File Reference

```
#include "common.h"
```

Classes

- class [Entity](#)
- class [Textured_object](#)

5.8 /home/nils/TDP005/Game/game_state.cpp File Reference

```
#include "game_state.h"
#include "menu_state.h"
```

5.9 /home/nils/TDP005/Game/game_state.h File Reference

```
#include "state.h"
#include "World.h"
```


Classes

- class [Game_State](#)

5.10 /home/nils/TDP005/Game/Level.cpp File Reference

```
#include "Level.h"
```

5.11 /home/nils/TDP005/Game/Level.h File Reference

```
#include <fstream>
#include <map>
#include "Enemy.h"
#include "Powerup.h"
```

Classes

- class [Level](#)

5.12 /home/nils/TDP005/Game/main.cpp File Reference

```
#include "common.h"
#include "menu_state.h"
#include "SFML/Audio.hpp"
```

Functions

- int [main](#) ()

Variables

- size_t const [width](#) = 1600
- size_t const [height](#) = 900

5.12.1 Function Documentation

5.12.1.1 main()

```
int main ( )
```

5.12.2 Variable Documentation

5.12.2.1 height

```
size_t const height = 900
```

5.12.2.2 width

```
size_t const width = 1600
```

5.13 /home/nils/TDP005/Game/menu_state.cpp File Reference

```
#include "menu_state.h"  
#include "game_state.h"
```

5.14 /home/nils/TDP005/Game/menu_state.h File Reference

```
#include <functional>  
#include "common.h"  
#include "state.h"  
#include "game_state.h"  
#include "Level.h"
```

Classes

- class [Menu_State](#)
- struct [Menu_State::Entry](#)
- class [Main_Menu_State](#)
- class [Help_State](#)
- class [Level_Selection_State](#)
- class [Pause_State](#)
- class [End_State](#)

5.15 /home/nils/TDP005/Game/Player.cpp File Reference

```
#include "Player.h"  
#include "World.h"  
#include "state.h"  
#include "menu_state.h"
```

5.16 /home/nils/TDP005/Game/Player.h File Reference

```
#include "common.h"
#include "Bullet.h"
```

Classes

- class [Player](#)

5.17 /home/nils/TDP005/Game/Powerup.cpp File Reference

```
#include "Powerup.h"
```

5.18 /home/nils/TDP005/Game/Powerup.h File Reference

```
#include "common.h"
#include "Entity.h"
```

Classes

- class [Powerup](#)

5.19 /home/nils/TDP005/Game/state.cpp File Reference

```
#include "state.h"
```

5.20 /home/nils/TDP005/Game/state.h File Reference

```
#include "common.h"
#include <SFML/Audio.hpp>
```

Classes

- class [State](#)
- class [Exit_State](#)

5.21 /home/nils/TDP005/Game/Texture_manager.cpp File Reference

```
#include "Texture_manager.h"  
#include <stdexcept>
```

5.22 /home/nils/TDP005/Game/Texture_manager.h File Reference

```
#include <iostream>  
#include <string>  
#include <vector>  
#include <memory>  
#include <cmath>  
#include <SFML/Graphics.hpp>  
#include <map>
```

Classes

- class [Texture_Manager](#)

5.23 /home/nils/TDP005/Game/World.cpp File Reference

```
#include "menu_state.h"  
#include "World.h"
```

5.24 /home/nils/TDP005/Game/World.h File Reference

```
#include "common.h"  
#include "state.h"  
#include "Entity.h"  
#include "Player.h"  
#include "Level.h"
```

Classes

- class [World](#)

Index

- /home/nils/TDP005/Game/Bullet.cpp, [53](#)
- /home/nils/TDP005/Game/Bullet.h, [53](#)
- /home/nils/TDP005/Game/Enemy.cpp, [53](#)
- /home/nils/TDP005/Game/Enemy.h, [54](#)
- /home/nils/TDP005/Game/Entity.cpp, [54](#)
- /home/nils/TDP005/Game/Entity.h, [54](#)
- /home/nils/TDP005/Game/Level.cpp, [55](#)
- /home/nils/TDP005/Game/Level.h, [55](#)
- /home/nils/TDP005/Game/Player.cpp, [56](#)
- /home/nils/TDP005/Game/Player.h, [57](#)
- /home/nils/TDP005/Game/Powerup.cpp, [57](#)
- /home/nils/TDP005/Game/Powerup.h, [57](#)
- /home/nils/TDP005/Game/Texture_manager.cpp, [58](#)
- /home/nils/TDP005/Game/Texture_manager.h, [58](#)
- /home/nils/TDP005/Game/World.cpp, [58](#)
- /home/nils/TDP005/Game/World.h, [58](#)
- /home/nils/TDP005/Game/common.h, [53](#)
- /home/nils/TDP005/Game/game_state.cpp, [54](#)
- /home/nils/TDP005/Game/game_state.h, [54](#)
- /home/nils/TDP005/Game/main.cpp, [55](#)
- /home/nils/TDP005/Game/menu_state.cpp, [56](#)
- /home/nils/TDP005/Game/menu_state.h, [56](#)
- /home/nils/TDP005/Game/state.cpp, [57](#)
- /home/nils/TDP005/Game/state.h, [57](#)
- ~Big_Plane
 - Big_Plane, [8](#)
- ~Bomb
 - Bomb, [10](#)
- ~Bullet
 - Bullet, [11](#)
- ~Enemy
 - Enemy, [14](#)
- ~Enemy_Bullet
 - Enemy_Bullet, [15](#)
- ~Entity
 - Entity, [17](#)
- ~Level
 - Level, [25](#)
- ~Player
 - Player, [33](#)
- ~Player_Bullet
 - Player_Bullet, [37](#)
- ~Powerup
 - Powerup, [39](#)
- ~Small_Plane
 - Small_Plane, [41](#)
- ~State
 - State, [44](#)
- ~World
 - World, [48](#)
- Action
 - Menu_State, [29](#)
- action
 - Menu_State::Entry, [20](#)
- add
 - Menu_State, [29](#)
- add_score
 - World, [49](#)
- background
 - Menu_State, [30](#)
- Big_Plane, [7](#)
 - ~Big_Plane, [8](#)
 - Big_Plane, [8](#)
 - freeze, [8](#)
 - freeze_state, [9](#)
 - get_type, [8](#)
 - shoot_speed, [9](#)
 - shoot_time, [9](#)
 - shoot_timer, [9](#)
 - tick, [8](#)
- Bomb, [9](#)
 - ~Bomb, [10](#)
 - Bomb, [10](#)
 - get_type, [10](#)
- Bullet, [11](#)
 - ~Bullet, [11](#)
 - Bullet, [11](#)
- cheat
 - Player, [33](#)
- collision
 - Enemy, [14](#)
 - Enemy_Bullet, [16](#)
 - Entity, [18](#)
 - Player, [33](#)
 - Player_Bullet, [38](#)
 - Powerup, [39](#)
- counter
 - World, [49](#)
- data
 - Level, [26](#)
- delay
 - Menu_State, [30](#)
- End_State, [12](#)
 - End_State, [12](#)
 - rating, [13](#)

- render, 12
- score_info, 13
- status_text, 13
- texture, 13
- Enemy, 14
 - ~Enemy, 14
 - collision, 14
 - Enemy, 14
- Enemy_Bullet, 15
 - ~Enemy_Bullet, 15
 - collision, 16
 - Enemy_Bullet, 15
 - get_type, 16
- enter_pressed
 - Menu_State, 30
- Entity, 16
 - ~Entity, 17
 - collision, 18
 - Entity, 17
 - freeze, 18
 - get_health, 18
 - get_sprite, 18
 - get_type, 18
 - health, 19
 - height, 19
 - kill_me, 18
 - location, 19
 - render, 18
 - speed, 19
 - sprite, 19
 - tick, 19
 - width, 20
- entries
 - Menu_State, 30
- Exit_State, 21
 - render, 21
 - tick, 21
- font
 - Menu_State, 30
 - World, 50
- freeze
 - Big_Plane, 8
 - Entity, 18
 - Player, 33
 - Small_Plane, 41
 - World, 49
- freeze_state
 - Big_Plane, 9
 - Player, 35
 - Small_Plane, 42
 - World, 50
- Game_State, 22
 - Game_State, 22
 - level_path, 23
 - render, 23
 - tick, 23
 - world, 23
- game_time
 - World, 50
- game_timer
 - World, 50
- get
 - Texture_Manager, 46
- get_health
 - Entity, 18
- get_shield_time
 - Player, 33
- get_sprite
 - Entity, 18
- get_type
 - Big_Plane, 8
 - Bomb, 10
 - Enemy_Bullet, 16
 - Entity, 18
 - Player, 34
 - Player_Bullet, 38
 - Powerup, 39
 - Small_Plane, 41
- give_invincible
 - Player, 35
- has_shield
 - Player, 34
- health
 - Entity, 19
- height
 - Entity, 19
 - main.cpp, 56
- Help_State, 24
 - Help_State, 24
 - keyboard_guide, 25
 - render, 24
 - texture, 25
- insert_object
 - World, 49
- instance
 - Texture_Manager, 46
- invincibility_time
 - Player, 35
- invincibility_timer
 - Player, 35
- invincible
 - Player, 35
- is_invincible
 - Player, 34
- keyboard_guide
 - Help_State, 25
- kill_me
 - Entity, 18
- Level, 25
 - ~Level, 25
 - data, 26
 - Level, 25

- load_enemy, 26
- max_score, 26
- level
 - World, 50
- level_name
 - World, 50
- level_path
 - Game_State, 23
- Level_Selection_State, 26
 - Level_Selection_State, 27
- life_info
 - World, 50
- load_enemy
 - Level, 26
- location
 - Entity, 19
- main
 - main.cpp, 55
- main.cpp
 - height, 56
 - main, 55
 - width, 56
- Main_Menu_State, 27
 - Main_Menu_State, 27
- manage_collision
 - World, 49
- max_score
 - Level, 26
- Menu_State, 28
 - Action, 29
 - add, 29
 - background, 30
 - delay, 30
 - enter_pressed, 30
 - entries, 30
 - font, 30
 - Menu_State, 29
 - on_key_press, 29
 - render, 29
 - selected, 30
 - tick, 29
- Menu_State::Entry, 20
 - action, 20
 - state, 20
 - text, 20
- name
 - Powerup, 40
- objects
 - World, 50
- on_key_press
 - Menu_State, 29
 - State, 44
- on_key_release
 - State, 44
- Pause_State, 31
 - Pause_State, 31
- Player, 32
 - ~Player, 33
 - cheat, 33
 - collision, 33
 - freeze, 33
 - freeze_state, 35
 - get_shield_time, 33
 - get_type, 34
 - give_invincible, 35
 - has_shield, 34
 - invincibility_time, 35
 - invincibility_timer, 35
 - invincible, 35
 - is_invincible, 34
 - Player, 33
 - set_position, 34
 - shield, 35
 - shield_time, 35
 - shield_timer, 36
 - shoot, 34
 - shoot_speed, 36
 - shoot_time, 36
 - shoot_timer, 36
 - tick, 34
 - triple_time, 36
 - triple_timer, 36
 - tripleshot, 36
 - update_status, 34
- player
 - World, 51
- Player_Bullet, 37
 - ~Player_Bullet, 37
 - collision, 38
 - get_type, 38
 - Player_Bullet, 37
 - tick, 38
- Powerup, 38
 - ~Powerup, 39
 - collision, 39
 - get_type, 39
 - name, 40
 - Powerup, 39
- rating
 - End_State, 13
- render
 - End_State, 12
 - Entity, 18
 - Exit_State, 21
 - Game_State, 23
 - Help_State, 24
 - Menu_State, 29
 - State, 44
 - Textured_object, 47
 - World, 49
- run
 - State, 44

- score
 - World, [51](#)
- score_info
 - End_State, [13](#)
 - World, [51](#)
- selected
 - Menu_State, [30](#)
- set_position
 - Player, [34](#)
- set_texture
 - Textured_object, [47](#)
- shield
 - Player, [35](#)
- shield_info
 - World, [51](#)
- shield_time
 - Player, [35](#)
- shield_timer
 - Player, [36](#)
- shoot
 - Player, [34](#)
- shoot_speed
 - Big_Plane, [9](#)
 - Player, [36](#)
 - Small_Plane, [42](#)
- shoot_time
 - Big_Plane, [9](#)
 - Player, [36](#)
 - Small_Plane, [42](#)
- shoot_timer
 - Big_Plane, [9](#)
 - Player, [36](#)
 - Small_Plane, [42](#)
- Small_Plane, [40](#)
 - ~Small_Plane, [41](#)
 - freeze, [41](#)
 - freeze_state, [42](#)
 - get_type, [41](#)
 - shoot_speed, [42](#)
 - shoot_time, [42](#)
 - shoot_timer, [42](#)
 - Small_Plane, [41](#)
 - tick, [41](#)
 - upp_state, [42](#)
 - vertical_move, [42](#)
 - vertical_time, [43](#)
 - vertical_timer, [43](#)
- spawn_time
 - World, [51](#)
- spawn_timer
 - World, [51](#)
- speed
 - Entity, [19](#)
- sprite
 - Entity, [19](#)
- State, [43](#)
 - ~State, [44](#)
 - on_key_press, [44](#)
 - on_key_release, [44](#)
 - render, [44](#)
 - run, [44](#)
 - State, [44](#)
 - tick, [45](#)
- state
 - Menu_State::Entry, [20](#)
- status_text
 - End_State, [13](#)
- text
 - Menu_State::Entry, [20](#)
- texture
 - End_State, [13](#)
 - Help_State, [25](#)
- Texture_Manager, [45](#)
 - get, [46](#)
 - instance, [46](#)
 - Texture_Manager, [45](#)
 - textures, [46](#)
- Textured_object, [46](#)
 - render, [47](#)
 - set_texture, [47](#)
 - Textured_object, [47](#)
- textures
 - Texture_Manager, [46](#)
- tick
 - Big_Plane, [8](#)
 - Entity, [19](#)
 - Exit_State, [21](#)
 - Game_State, [23](#)
 - Menu_State, [29](#)
 - Player, [34](#)
 - Player_Bullet, [38](#)
 - Small_Plane, [41](#)
 - State, [45](#)
 - World, [49](#)
- triple_time
 - Player, [36](#)
- triple_timer
 - Player, [36](#)
- tripleshot
 - Player, [36](#)
- update_status
 - Player, [34](#)
- upp_state
 - Small_Plane, [42](#)
- vertical_move
 - Small_Plane, [42](#)
- vertical_time
 - Small_Plane, [43](#)
- vertical_timer
 - Small_Plane, [43](#)
- width
 - Entity, [20](#)
 - main.cpp, [56](#)

World, [47](#)
 ~World, [48](#)
 add_score, [49](#)
 counter, [49](#)
 font, [50](#)
 freeze, [49](#)
 freeze_state, [50](#)
 game_time, [50](#)
 game_timer, [50](#)
 insert_object, [49](#)
 level, [50](#)
 level_name, [50](#)
 life_info, [50](#)
 manage_collision, [49](#)
 objects, [50](#)
 player, [51](#)
 render, [49](#)
 score, [51](#)
 score_info, [51](#)
 shield_info, [51](#)
 spawn_time, [51](#)
 spawn_timer, [51](#)
 tick, [49](#)
 World, [48](#)
world
 Game_State, [23](#)