sf::Drawable

draw(inout target: RenderTarget&, states: RenderStates)

Status

+ is_physics: bool
+ is_interrupted: bool
+ is_drawable: bool
+ is_static: bool

План:
ok|file manager
ok|tile map
> |scene
> |hero
> |main program
> |bullet
> |enemy

> |final gameplay

<class _Type>
Matrix

- h_size: const size_t
- w_size: const size_t

+ Matrix(init: std::initializer_list<std::vector< _Type >>, h_size_: size_t, v_size_: size_t)
+ Matrix(const Matrix&)
+ Matrix(Matrix&&)

+ Matrix& operator=(const Matrix&)
+ Matrix& operator=(Matrix&&)

+ getHorizontalSize(): size_t
+ getVerticalSize(): size_t

TailMap

using Matrix_texture = Matrix<sf::Texture*>
- mtx: Matrix_texture*

+ TailMap(std::string lua_file, b2Body* body)
+ TailMap(const TailMap&)
+ TailMap(TailMap&&)
+ override ~TailMap()

+ operator=(unit: const TailMap&): TailMap&
+ operator=(unit: TailMap&&): TailMap&

+ getMatrix(): Matrix<sf::Texture*>

+ override setMainSizeBody(size: const sf::Vector2f&)

+ override draw(inout target: RenderTarget&, states: RenderStates)

- initMainBody_after_mtx_init()

- override setTexture(texture: sf::Texture)
- override getTexture(): sf::Textue

- override addCollisionObject(b2fd: const b2FixtureDef*)

- override addStateAndRectOfTexture(state: std::string, rect: sf::Rect)
- override delStateAndRectOfTexture(state: std::string)
- override getStatesAndRectsOfTexture(): std::map<std::string, RectAndFrames>
- override getRectOfTexture(state: std::string): RectAndFrames

- override setStateOfTexture(state: std::string)
- override getStateOftexture(): std::string

Unit:
конструктори: ok

положення:ok

поворот: ok

маса: ok

статуси: ok

швидкість: ok

затухання сили:
ok

user data: ok

texture: ok

size of body: ok

collision: ok

сили: ok

updaters: ~

draw:
ok(доробити угол поворота).

Unit

status: Status = {false, false, false, true}
main_body: b2Body*
main_texture: sf::Texture
main_state_and_rest_of_texture: std::map<std::string, RectAndFrames>
main_state_of_texture: std::string
main_collisions: std::vector<b2Fixture*>
main_size_body: sf::Vector2f

+ Unit(body: b2Body*&)
+ Unit(unit: const Unit&)
+ Unit(unit: Unit&&)
+ ~Unit()

+ operator=(unit: const Unit&): Unit&
+ operator=(unit: Unit&&): Unit&

+ setStaticStatus(stat: bool)
+ setDrawStatus(stat: bool)
+ setPhysicsStatus(stat: bool)
+ setInterruptStatus(stat: bool)
+ getStatus(): Status

- setPhysicsBodyUserData(name: std::string)
- getPhysicsBodyUserData(): std::string

+ addCollisionObject(b2fd: const b2FixtureDef*)
+ getCollisionObjects(): const std::vector<b2Fixture*>&

+ setTexture(texture: sf::Texture*)
+ getTexture(): const sf::Texture*

+ addStateAndRectOfTexture(state: std::string, rect:RectAndFrames)
+ delStateAndRectOfTexture(state: std::string)
+ getStatesAndRectsOfTexture(): std::map<std::string,RectAndFrames>
+ getRectOfTexture(state: std::string):RectAndFrames

+ setStateOfTexture(state: std::string)
+ getStateOfTexture(): std::string

+ getPosition(): sf::Vector2f

+ getRotation(): float

+ setMass(mass: float, point sf::Vector2f)
+ setMassToCenter(mass: float)
+ getMass(): float
+ getMassData(): MassData

+ addImpulse(impuls: sf::Vector2f, point: sf::Vector2f)
+ addImpulseToCenter(impuls: sf::Vector2f)

+ addForce(force: sf::Vector2f, point: sf::Vector2f)
+ addForceToCenter(force: sf::Vector2f)

+ setLinearDamping(damping: float)
+ getLinearDamping(): float

+ setAngularDamping(damping: float)
+ getAngularDamping(): float

+ setLinearSpeed(vec: sf::Vector2f)
+ getLinearSpeed(): sf::Vector2f

+ setAngularSpeed(speed: float)
+ getAngularSpeed(): float

+ setMainSizeBody(size: const sf::Vector2f&)
+ getMainSizeBody(): sf::Vector2f&

+ getMainBody(): b2Body*

+ updateEveryFrame()

+ override draw(inout target: RenderTarget&, states: RenderStates)

RectAndFrames

+ rect: sf::IntRect
+ frames: size_t

UnitCreator:
Колізія:
add[різні види колізій]

<class... _MainUnit>
UnitCreator

- main_collisions: std::vector<b2FixtureDef>
- main_world: b2World*

+ status: Status
+ start_linear_speed: sf::Vector2f
+ start_angular_speed: float
+ start_linear_damping: float
+ start_angular_damping: float
+ start_angle: float
+ start_position: sf::Vector2f

+ is_fixed: bool
+ is_bullet: bool

+ mass: float
+ gravity_scale: float

+ texture: sf::Texture*
+ states_and_texture_rects: Unit::StatesAndRectsOfTexture
+ start_state: std::string

+ size_of_visible_texture: sf::Vector2f

+ UnitCreator(world: b2World*)
+ ~UnitCreator()

<class... _Params>
+ create(_Vals: _Params...): std::shared_ptr<_MainUnit>

+ restart(new_world: b2World* = nullptr)

+ addCollision(collision: b2FixtureDef)

+ addBoxCollision(size: sf::Vector2f,
position: sf::Vector2f = {0,0},
angle: float = 0.f,
density: float = 0.5f,
friction: float = 0.2f,
restitution: float = 0.f)

+ addCircleCollision(redius: float,
position: sf::Vector2f = {0,0},
density: float = 0.5f,
friction: float = 0.2f,
restitution: float = 0.f)

+ addStateAndTextureRect(state: std::string, raf: Unit::RectAndFrames)
+ addStatesAndTexturesRect(init: std::initializer_list < std::pair < std::string, Unit::RectAndFrames> >)

+ loadTextureFromFile(file_name: std::string)

Доробити логи
...

FileManager_singleton

+ const FOLDER_WITH_RESOURCES: std::string = "ResDir"
+ const FOLDER_WITH_LOGS: std::string = "LogDir"
+ const FOLDER_WITH_SOURCE: std::string = "Simple2dStrilalka"

+ FileManager_singleton()
+ ~FileManager_singleton()

FileManager

- static fm_singleton: std::shared_ptr<FileManager_singleton>

+ static getResFile(file_name: std::string): std::string
+ static getLogFile(file_name: std::string): std::string

0..*

0..*