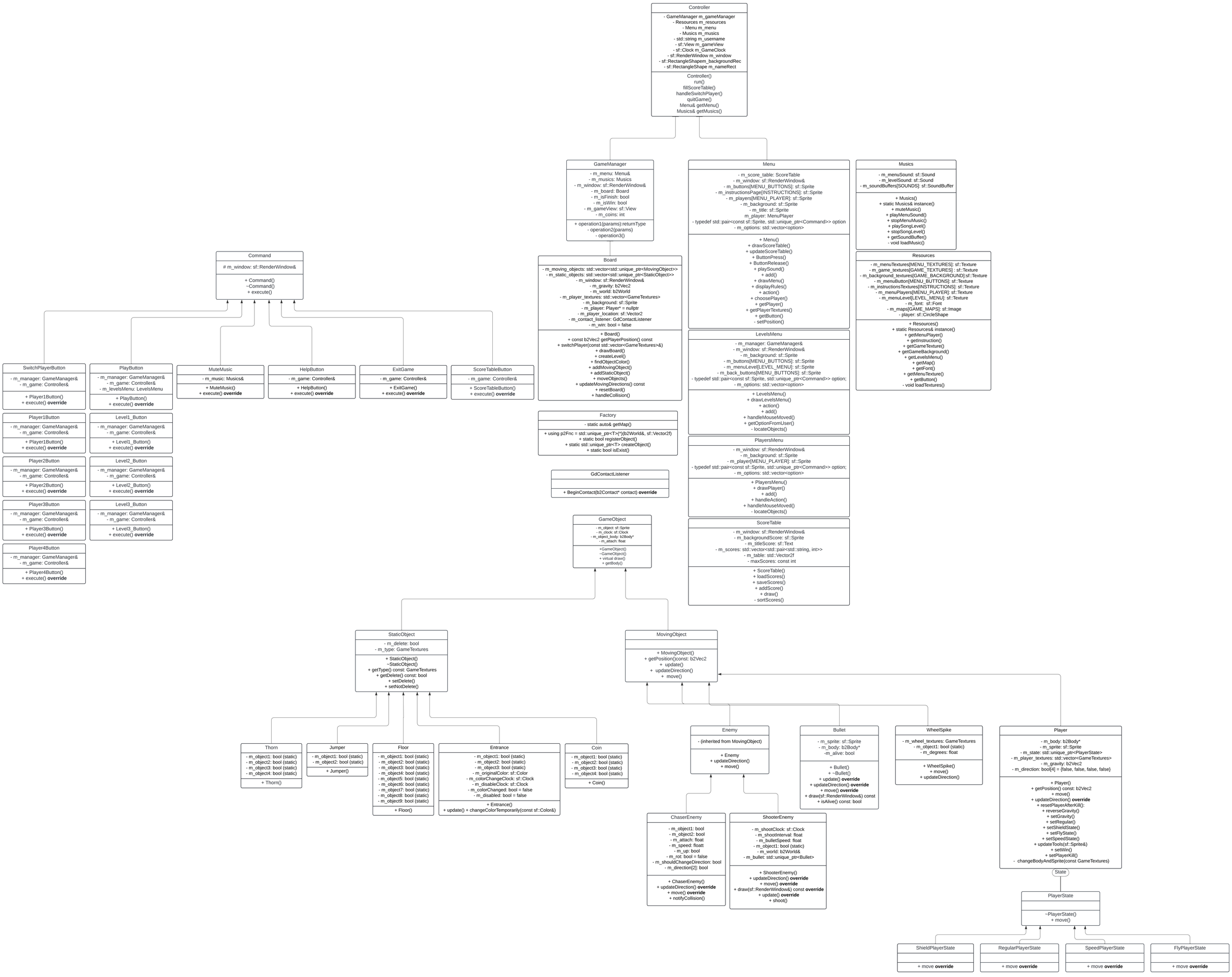
Controller
- GameManager m_gameManager
- Resources m_resources
- Menu m_menu
- Musics m_musics
- std::string m_username
- sf::View m_gameView
- sf::Clock m_GameClock
- sf::RenderWindow m_window
- sf::RectangleShapem_backgroundRec
- sf::RectangleShape m_nameRect
Controller()
run()
fillScoreTable()
handleSwitchPlayer()
quitGame()
Menu& getMenu()
Musics& getMusics()
GameManager
- m_menu: Menu&
- m_musics: Musics
- m_window: sf::RenderWindow&
- m_board: Board
- m_isFinish: bool
- m_isWin: bool
- m_gameView: sf::View
- m_coins: int
+ operation1(params):returnType
- operation2(params)
- operation3()
Menu
- m_score_table: ScoreTable
- m_window: sf::RenderWindow&
- m_buttons[MENU_BUTTONS]: sf::Sprite
- m_instructionsPage[INSTRUCTIONS]: sf::Sprite
- m_players[MENU_PLAYER]: sf::Sprite
- m_background: sf::Sprite
- m_title: sf::Sprite
m_player: MenuPlayer
- typedef std::pair<const sf::Sprite, std::unique_ptr<Command>> option
- m_options: std::vector<option>
+ Menu()
+ drawScoreTable()
+ updateScoreTable()
+ ButtonPress()
+ ButtonRelease()
+ playSound()
+ add()
+ drawMenu()
+ displayRules()
+ action()
+ choosePlayer()
+ getPlayer()
+ getPlayerTextures()
+ getButton()
- setPosition()
Musics
- m_menuSound: sf::Sound
- m_levelSound: sf::Sound
- m_soundBuffers[SOUNDS]: sf::SoundBuffer
+ Musics()
+ static Musics& instance()
+ muteMusic()
+ playMenuSound()
+ stopMenuMusic()
+ playSongLevel()
+ stopSongLevel()
+ getSoundBuffer()
- void loadMusic()
Resources
- m_menuTextures[MENU_TEXTURES]: sf::Texture
- m_game_textures[GAME_TEXTURES] : sf::Texture
- m_background_textures[GAME_BACKGROUND]:sf::Texture
- m_menuButton[MENU_BUTTONS]: sf::Texture
- m_instructionsTextures[INSTRUCTIONS]: sf::Texture
- m_menuPlayers[MENU_PLAYER]: sf::Texture
- m_menuLevel[LEVEL_MENU]: sf::Texture
- m_font: sf::Font
- m_maps[GAME_MAPS]: sf::Image
- player: sf::CircleShape
+ Resources()
+ static Resources& instance()
+ getMenuPlayer()
+ getInstruction()
+ getGameTexture()
+ getGameBackground()
+ getLevelsMenu()
+ getMap()
+ getFont()
+ getMenuTexture()
+ getButton()
- void loadTextures()
Command
m_window: sf::RenderWindow&
+ Command()
~Command()
+ execute()
Board
- m_moving_objects: std::vector<std::unique_ptr<MovingObject>>
- m_static_objects: std::vector<std::unique_ptr<StaticObject>>
- m_window: sf::RenderWindow&
- m_gravity: b2Vec2
- m_world: b2World
- m_player_textures: std::vector<GameTextures>
- m_background: sf::Sprite
- m_player: Player* = nullptr
- m_player_location: sf::Vector2
- m_contact_listener: GdContactListener
- m_win: bool = false
+ Board()
+ const b2Vec2 getPlayerPosition() const
+ switchPlayer(const std::vector<GameTextures>&)
+ drawBoard()
+ createLevel()
+ findObjectColor()
+ addMovingObject()
+ addStaticObject()
+ moveObjects()
+ updateMovingDirections() const
+ resetBoard()
+ handleCollision()
LevelsMenu
- m_manager: GameManager&
- m_window: sf::RenderWindow&
- m_background: sf::Sprite
- m_buttons[MENU_BUTTONS]: sf::Sprite
- m_menuLevel[LEVEL_MENU]: sf::Sprite
- m_back_buttons[MENU_BUTTONS]: sf::Sprite
- typedef std::pair<const sf::Sprite, std::unique_ptr<Command>> option;
- m_options: std::vector<option>
+ LevelsMenu()
+ drawLevelsMenu()
+ action()
+ add()
+ handleMouseMoved()
+ getOptionFromUser()
- locateObjects()
PlayersMenu
- m_window: sf::RenderWindow&
- m_background: sf::Sprite
- m_player[MENU_PLAYER]: sf::Sprite
- typedef std::pair<const sf::Sprite, std::unique_ptr<Command>> option;
- m_options: std::vector<option>
+ PlayersMenu()
+ drawPlayer()
+ add()
+ handleAction()
+ handleMouseMoved()
- locateObjects()
ScoreTable
- m_window: sf::RenderWindow&
- m_backgroundScore: sf::Sprite
- m_titleScore: sf::Text
- m_scores: std::vector<std::pair<std::string, int>>
- m_table: std::Vector2f
- maxScores: const int
+ ScoreTable()
+ loadScores()
+ saveScores()
+ addScore()
+ draw()
- sortScores()
Factory
- static auto& getMap()
+ using p2Fnc = std::unique_ptr<T>(*)(b2World&, sf::Vector2f)
+ static bool registerObject()
+ static std::unique_ptr<T> createObject()
+ static bool isExist()
GdContactListener
+ BeginContact(b2Contact* contact) override
SwitchPlayerButton
- m_manager: GameManager&
- m_game: Controller&
+ Player1Button()
+ execute() override
PlayButton
- m_manager: GameManager&
- m_game: Controller&
- m_levelsMenu: LevelsMenu
+ PlayButton()
+ execute() override
MuteMusic
- m_music: Musics&
+ MuteMusic()
+ execute() override
HelpButton
- m_game: Controller&
+ HelpButton()
+ execute() override
ExitGame
- m_game: Controller&
+ ExitGame()
+ execute() override
ScoreTableButton
- m_game: Controller&
+ ScoreTableButton()
+ execute() override
Player1Button
- m_manager: GameManager&
- m_game: Controller&
+ Player1Button()
+ execute() override
Level1_Button
- m_manager: GameManager&
- m_game: Controller&
+ Level1_Button()
+ execute() override
Player2Button
- m_manager: GameManager&
- m_game: Controller&
+ Player2Button()
+ execute() override
Level2_Button
- m_manager: GameManager&
- m_game: Controller&
+ Level2_Button()
+ execute() override
Player3Button
- m_manager: GameManager&
- m_game: Controller&
+ Player3Button()
+ execute() override
Level3_Button
- m_manager: GameManager&
- m_game: Controller&
+ Level3_Button()
+ execute() override
Player4Button
- m_manager: GameManager&
- m_game: Controller&
+ Player4Button()
+ execute() override
GameObject
- m_object: sf::Sprite
- m_clock: sf::Clock
- m_object_body: b2Body*
- m_attach: float
+GameObject()
~GameObject()
+ virtual draw()
+ getBody()
StaticObject
- m_delete: bool
- m_type: GameTextures
+ StaticObject()
~StaticObject()
+ getType() const: GameTextures
+ getDelete() const: bool
+ setDelete()
+ setNotDelete()
MovingObject
+ MovingObject()
+ getPosition()const: b2Vec2
+ update()
+ updateDirection()
+ move()
Thorn
- m_object1: bool (static)
- m_object2: bool (static)
- m_object3: bool (static)
- m_object4: bool (static)
+ Thorn()
Jumper
- m_object1: bool (static)
- m_object2: bool (static)
+ Jumper()
Floor
- m_object1: bool (static)
- m_object2: bool (static)
- m_object3: bool (static)
- m_object4: bool (static)
- m_object5: bool (static)
- m_object6: bool (static)
- m_object7: bool (static)
- m_object8: bool (static)
- m_object9: bool (static)
+ Floor()
Entrance
- m_object1: bool (static)
- m_object2: bool (static)
- m_object3: bool (static)
- m_originalColor: sf::Color
- m_colorChangeClock: sf::Clock
- m_disableClock: sf::Clock
- m_colorChanged: bool = false
- m_disabled: bool = false
+ Entrance()
+ update() + changeColorTemporarily(const sf::Color&)
Coin
- m_object1: bool (static)
- m_object2: bool (static)
- m_object3: bool (static)
- m_object4: bool (static)
+ Coin()
Enemy
- (inherited from MovingObject)
+ Enemy
+ updateDirection()
+ move()
Bullet
- m_sprite: sf::Sprite
- m_body: b2Body*
-m_alive: bool
+ Bullet()
+ ~Bullet()
+ update() override
+ updateDirection() override
+ move() override
+ draw(sf: RenderWindow&) const
+ isAlive() const: bool
WheelSpike
- m_wheel_textures: GameTextures
- m_object1: bool (static)
- m_degrees: float
+ WheelSpike()
+ move()
+ updateDirection()
Player
- m_body: b2Body*
- m_sprite: sf::Sprite
- m_state: std::unique_ptr<PlayerState>
- m_player_textures: std::vector<GameTextures>
- m_gravity: b2Vec2
- m_direction: bool[4] = {false, false, false, false}
+ Player()
+ getPosition() const: b2Vec2
+ move()
+ updateDirection() override
+ resetPlayerAfterKill():
+ reverseGravity()
+ setGravity()
+ setRegular()
+ setShieldState()
+ setFlyState()
+ setSpeedState()
+ updateTools(sf::Sprite&)
+ setWin()
+ setPlayerKill()
- changeBodyAndSprite(const GameTextures)
State
ChaserEnemy
- m_object1: bool
- m_object2: bool
- m_attach: float
- m_speed: floatt
- m_up: bool
- m_rot: bool = false
- m_shouldChangeDirection: bool
- m_direction[2]: bool
+ ChaserEnemy()
+ updateDirection() override
+ move() override
+ notifyCollision()
ShooterEnemy
- m_shootClock: sf::Clock
- m_shootInterval: float
- m_bulletSpeed: float
- m_object1: bool (static)
- m_world: b2World&
- m_bullet: std::unique_ptr<Bullet>
+ ShooterEnemy()
+ updateDirection() override
+ move() override
+ draw(sf::RenderWindow&) const override
+ update() override
+ shoot()
PlayerState
~PlayerState()
+ move()
ShieldPlayerState
+ move override
RegularPlayerState
+ move override
SpeedPlayerState
+ move override
FlyPlayerState
+ move override