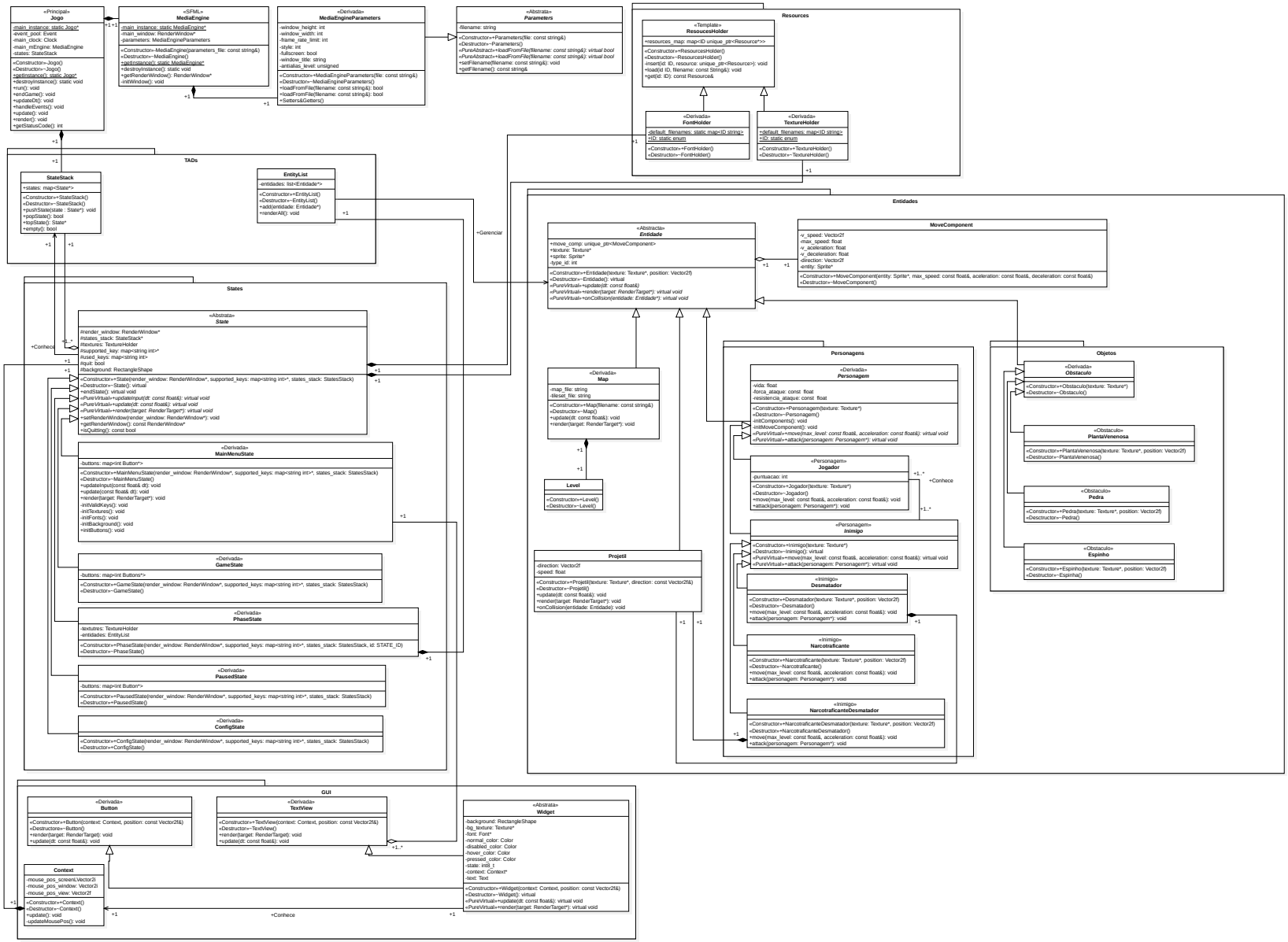
Game
«Principal»
Jogo
-main_instance: static Jogo*
-event_pool: Event
-main_clock: Clock
-main_mEngine: MediaEngine
-states: StateStack
+Constructor=~Jogo()
+Destructor=~Jogo()
+getInstance(): static Jogo*
+destroyInstance(): static void
+run(): void
+endGame(): void
+updateDt(): void
+handleEvents(): void
+update(): void
+render(): void
+getStatusCode(): int
«SFML»
MediaEngine
-main_instance: static MediaEngine*
-main_window: RenderWindow*
-parameters: MediaEngineParameters
+Constructor=~MediaEngine(parameters_file: const string&)
+Destructor=~MediaEngine()
+getInstance(): static MediaEngine*
+destroyInstance(): static void
+getRenderWindow(): RenderWindow*
-initWindow(): void
«Derivada»
MediaEngineParameters
-window_height: int
-window_width: int
-frame_rate_limit: int
-style: int
-fullscreen: bool
-window_title: string
-antialias_level: unsigned
+Constructor=+MediaEngineParameters(file: const string&)
+Destructor=~MediaEngineParameters()
+loadFromFile(filename: const string&): bool
+loadFromFile(filename: const string&): bool
+Setters&Getters()
«Abstrata»
Parameters
-filename: string
+Constructor=+Parameters(file: const string&)
+Destructor=~Parameters()
+PureAbstract=+loadFromFile(filename: const string&): virtual bool
+PureAbstract=+loadFromFile(filename: const string&): virtual bool
+setFilename(filename: const string&): void
+getFilename(): const string&
Resources
«Template»
ResoucesHolder
+resources_map: map<ID unique_ptr<Resource*>>
+Constructor=+ResoucesHolder()
+Destructor=~ResoucesHolder()
+insert(id: ID, resource: unique_ptr<Resource>): void
+load(id: ID, filename: const String&): void
+get(id: ID): const Resource&
«Derivada»
FontHolder
-default_filenames: static map<ID string>
+ID: static enum
+Constructor=+FontHolder()
+Destructor=~FontHolder()
«Derivada»
TextureHolder
+default_filenames: map<ID string>
+ID: static enum
+Constructor=+TextureHolder()
+Destructor=~TextureHolder()
TADs
StateStack
+states: map<State*>
+Constructor=+StateStack()
+Destructor=~StateStack()
+pushState(state : State*): void
+popState(): bool
+topState(): State*
+empty(): bool
EntityList
-entidades: list<Entidade*>
+Constructor=+EntityList()
+Destructor=~EntityList()
+add(entidade: Entidade*)
+renderAll(): void
+Gerenciar
Entidades
«Abstracta»
Entidade
+move_comp: unique_ptr<MoveComponent>
+texture: Texture*
+sprite: Sprite*
-type_id: int
+Constructor=+Entidade(texture: Texture*, position: Vector2f)
+Destructor=~Entidade(): virtual
+PureVirtual=+update(dt: const float&)
+PureVirtual=+render(target: RenderTarget*): virtual void
+PureVirtual=+onCollision(entidade: Entidade*): virtual void
MoveComponent
-v_speed: Vector2f
-max_speed: float
-v_aceleration: float
-v_deceleration: float
-direction: Vector2f
-entity: Sprite*
+Constructor=+MoveComponent(entity: Sprite*, max_speed: const float&, aceleration: const float&, deceleration: const float&)
+Destructor=~MoveComponent()
States
«Abstrata»
State
#render_window: RenderWindow*
#states_stack: StateStack*
#textures: TextureHolder
#supported_keys: map<string int>*
#used_keys: map<string int>
#quit: bool
#background: RectangleShape
+Constructor=+State(render_window: RenderWindow*, supported_keys: map<string int>*, states_stack: StatesStack)
+Destructor=~State(): virtual
+endState(): virtual void
+PureVirtual=+updateInput(dt: const float&): virtual void
+PureVirtual=+update(dt: const float&): virtual void
+PureVirtual=+render(target: RenderTarget*): virtual void
+setRenderWindow(render_window: RenderWindow*): void
+getRenderWindow(): const RenderWindow*
+isQuitting(): const bool
+Conhece
«Derivada»
MainMenuState
-buttons: map<int Button*>
+Constructor=+MainMenuState(render_window: RenderWindow*, supported_keys: map<string int>*, states_stack: StatesStack)
+Destructor=~MainMenuState()
+updateInput(const float& dt): void
+update(const float& dt): void
+render(target: RenderTarget*): void
-initValidKeys(): void
-initTextures(): void
-initFonts(): void
-initBackground(): void
-initButtons(): void
«Derivada»
GameState
-buttons: map<int Buttons*>
+Constructor=+GameState(render_window: RenderWindow*, supported_keys: map<string int>*, states_stack: StatesStack)
+Destructor=~GameState()
«Derivada»
PhaseState
-textutres: TextureHolder
-entidades: EntityList
+Constructor=+PhaseState(render_window: RenderWindow*, supported_keys: map<string int>*, states_stack: StatesStack, id: STATE_ID)
+Destructor=~PhaseState()
«Derivada»
PausedState
-buttons: map<int Button*>
+Constructor=+PausedState(render_window: RenderWindow*, supported_keys: map<string int>*, states_stack: StatesStack)
+Destructor=+PausedState()
«Derivada»
ConfigState
+Constructor=+ConfigState(render_window: RenderWindow*, supported_keys: map<string int>*, states_stack: StatesStack)
+Destructor=+ConfigState()
«Derivada»
Map
-map_file: string
-tileset_file: string
+Constructor=+Map(filename: const string&)
+Destructor=~Map()
+update(dt: const float&): void
+render(target: RenderTarget*): void
Level
+Constructor=+Level()
+Destructor=~Level()
Projetil
-direction: Vector2f
-speed: float
+Constructor=+Projetil(texture: Texture*, direction: const Vector2f&)
+Destructor=~Projetil()
+update(dt: const float&): void
+render(target: RenderTarget*): void
+onCollision(entidade: Entidade): void
Personagens
«Derivada»
Personagem
-vida: float
-forca_ataque: const float
-resistencia_ataque: const float
+Constructor=+Personagem(texture: Texture*)
+Destructor=~Personagem()
-initComponents(): void
-initMoveComponent(): void
+PureVirtual=+move(max_level: const float&, acceleration: const float&): virtual void
+PureVirtual=+attack(personagem: Personagem*): virtual void
«Personagem»
Jogador
-pontuacao: int
+Constructor=+Jogador(texture: Texture*)
+Destructor=~Jogador()
+move(max_level: const float&, acceleration: const float&): void
+attack(personagem: Personagem*): void
+Conhece
«Personagem»
Inimigo
+Constructor=+Inimigo(texture: Texture*)
+Destructor=~Inimigo(): virtual
+PureVirtual=+move(max_level: const float&, acceleration: const float&): virtual void
+PureVirtual=+attack(personagem: Personagem*): virtual void
«Inimigo»
Desmatador
+Constructor=+Desmatador(texture: Texture*, position: Vector2f)
+Destructor=~Desmatador()
+move(max_level: const float&, acceleration: const float&): void
+attack(personagem: Personagem*): void
«Inimigo»
Narcotraficante
+Constructor=+Narcotraficante(texture: Texture*, position: Vector2f)
+Destructor=~Narcotraficante()
+move(max_level: const float&, acceleration: const float&): void
+attack(personagem: Personagem*): void
«Inimigo»
NarcotraficanteDesmatador
+Constructor=+NarcotraficanteDesmatador(texture: Texture*, position: Vector2f)
+Destructor=~NarcotraficanteDesmatador()
+move(max_level: const float&, acceleration: const float&): void
+attack(personagem: Personagem*): void
Objetos
«Derivada»
Obstaculo
+Constructor=+Obstaculo(texture: Texture*)
+Destructor=~Obstaculo()
«Obstaculo»
PlantaVenenosa
+Constructor=+PlantaVenenosa(texture: Texture*, position: Vector2f)
+Destructor=~PlantaVenenosa()
«Obstaculo»
Pedra
+Constructor=+Pedra(texture: Texture*, position: Vector2f)
+Destructor=~Pedra()
«Obstaculo»
Espinho
+Constructor=+Espinho(texture: Texture*, position: Vector2f)
+Destructor=~Espinho()
GUI
«Derivada»
Button
+Constructor=+Button(context: Context, position: const Vector2f&)
+Destructor=~Button()
+render(target: RenderTarget): void
+update(dt: const float&): void
«Derivada»
TextView
+Constructor=+TextView(context: Context, position: const Vector2f&)
+Destructor=~TextView()
+render(target: RenderTarget): void
+update(dt: const float&): void
«Abstrata»
Widget
-background: RectangleShape
-bg_texture: Texture*
-font: Font*
-normal_color: Color
-disabled_color: Color
-hover_color: Color
-pressed_color: Color
-state: int8_t
-context: Context*
-text: Text
+Constructor=+Widget(context: Context, position: const Vector2f&)
+Destructor=~Widget(): virtual
+PureVirtual=+update(dt: const float&): virtual void
+PureVirtual=+render(target: RenderTarget*): virtual void
Context
-mouse_pos_screen:Vector2i
-mouse_pos_window: Vector2i
-mouse_pos_view: Vector2f
+Constructor=+Context()
+Destructor=~Context()
+update(): void
-updateMousePos(): void
+Conhece