Projeto LPOO: Entrega Intermédia

Lift

MESTRADO INTEGRADO EM ENGENHARIA INFORMÁTICA E COMPUTAÇÃO

Turma 4, Grupo 3:

Antero Campos Gandra, up201607926@fe.up.pt

Guilherme Jose Ferreira do Couto Fonseca da Silva, up201603647@fe.up.pt

Índice

Architecture Design

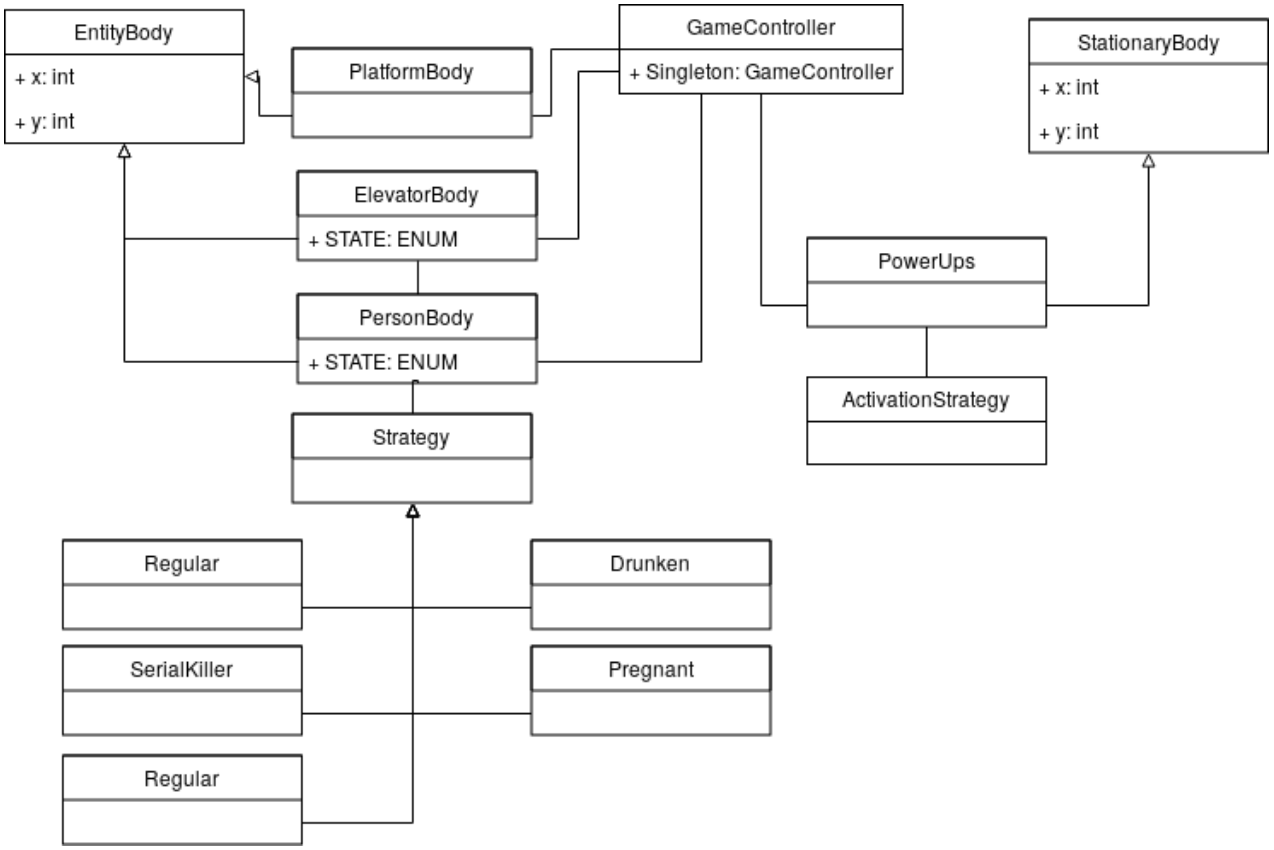
O programa é desenhado com base no padrão de desenho *Model View Controller,* cada um destes componentes é definido por um *singleton* para facilitar a comunicação entre os diferentes componentes.

UML

De seguida, encontram-se os diagramas de classes dos três componentes principais.

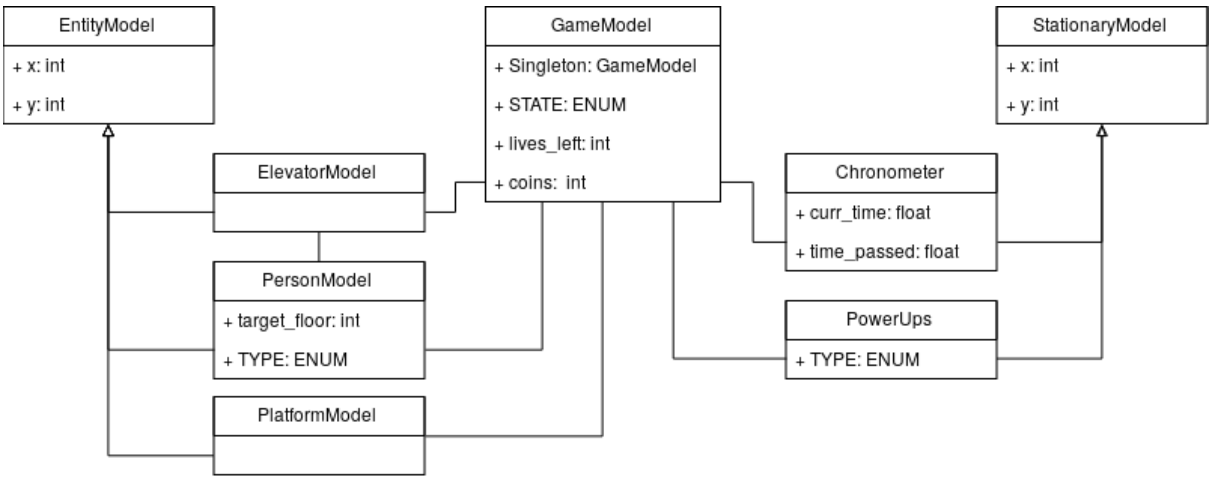
Não existe nenhuma relação representada entre estes componentes pois estas estão escondidas através de *singletons*.

Controller



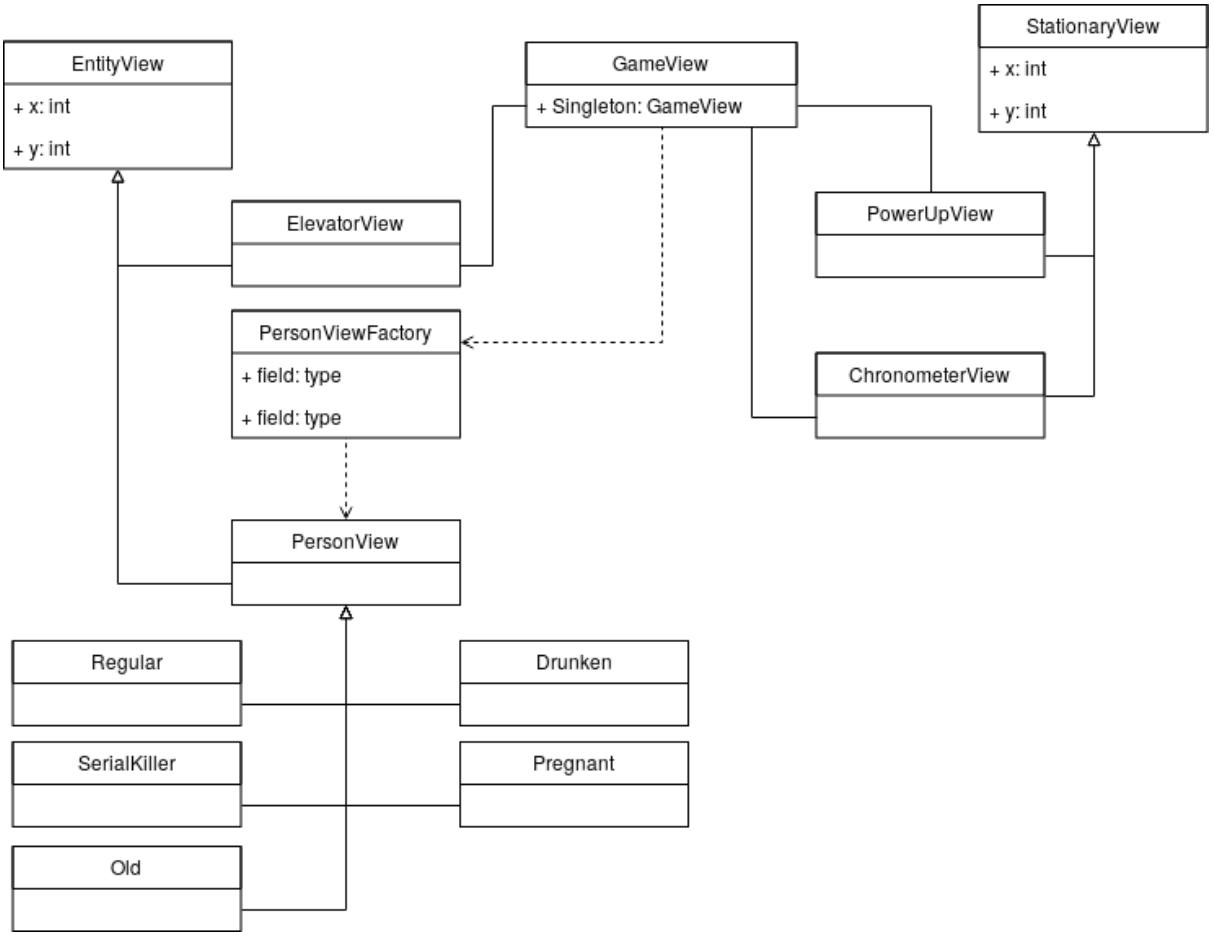
- **GameController:** Controla a física e a lógica do jogo.
- **StationaryBody:** Guarda a posição e corpo para objetos que não são afetados por forças.
- **PowerUps:** Guarda o corpo dos power ups que vão aparecendo during the game.
- **ActivationStrategy:** Ação correspondente ao power up, no futuro vão ser criadas subclasses para diferentes power ups.
- **EntityBody:** Guarda o corpo para objetos que são afetados por forças.
- **ElevatorBody:** Corpo correspondente ao elevador.
- **PersonBody:** Corpo correspondente a pessoas.
- **Strategy:** Estratégia de uma pessoa, todas as suas subclasses são diferente estratégias, as estratégias variam com o tipo de pessoas.
- **PlattormBody:** Guarda o corpo de uma plataforma.

Model



- **GameModel:** Guarda o estado atual do jogo.
- **EntityModel:** Guarda a informação básica de uma entidade.
- **ElevatorModel:** Guarda a informação respectiva a um elevador.
- **PersonModel:** Guarda a informação relativa a uma pessoa.
- **PlatformModel:** Guarda a informação relativa a uma plataforma.
- **StationaryModel:** Guarda a informação básica de um objeto que não é afetado por forças.
- **Chronometer:** Guarda o estado do cronómetro.
- **PowerUps:** Guarda a informação relativa a um *power up*.

View



- **GameView:** Cria todas as view respectivas ao jogo.
- **EntityView:** Guarda as informações básicas referentes às *views.*
- **ElevatorView:** Vista do elevador.
- **PlatformView:** Vista das plataformas.
- **PersonView:** Vista básica de uma pessoa, todas as suas subclasses dizem respeito a diferentes tipos de pessoas.
- **StationaryView:** Vista para objetos estacionários.
- **PowerUpView:** Vista dos power ups.
- **ChronometerView:** Vista do cronômetro.

Aspetos comportamentais

A lógica do jogo é bastante simples: o elevador apanha pessoas e tem que as levar até ao andar desejado.

Para o elevador parar nos andares são usadas colisões entre o elevador e a plataforma. Quando o elevador aproxima-se de uma plataforma por baixo é usado o fim da colisão, enquanto que o início da colisão ocorre quando o elevador aproxima a plataforma por cima.

O comportamento das pessoas divide-se em em três estados (esperar, no elevador e no destino). As pessoas começam a esperar quando são criadas, passam a estar no elevador quando o jogador clica nelas (e o elevador está na plataforma certa) e finalmente passam ao estado "no destino" quando o elevador pára no andar desejado.

Design Patterns

- **Singleton:** as classes GameView, GameController e GameModel.
- **State:** as pessoas e o elevador têm diversos estados:
    - **Pessoas**
        - Á espera
        - No elevador
        - No destino
    - **Elevador**
        - Em movimento
        - Parado
- **Strategy:** Cada pessoa possui uma estratégia para determinar os seus movimentos.
- **Factory:** Uma fábrica é usada para criar as views para os diferentes tipos de pessoa.

Test Design

Os testes vão ser feitos apenas para as classes GameController e GameModel, invocando as funções de GameController e verificando os seus efeitos no modelo.

Lista de testes

- **Movimento do elevador**: Mudar o target floor e verifica se o elevador atinge o andar pretendido.
- **Mover uma pessoa para o elevador:** Verifica se um objeto se move da plataforma para o elevador.
- **Mover uma pessoa do elevador para a plataforma desejada:** Verifica se o objeto é movido do elevador para a plataforma desejada.
- **Verificar se as pessoas valem menos pontos com o tempo:** Verificar a variável dos pontos para pessoa.
- **Verificar os efeitos dos power ups:** Diferentes testes dependendo do power up.
- **Verificar o fim do jogo:**
    - O tempo esgotou-se.
    - As vidas acabaram.