






Cave Keeper








The Game

GameController

- O GameController foi idealizado como um Design Pattern Singleton, pois só pode ser instanciado uma vez, por conter função main e instâncias das classes de outros componentes.
- O GameController também se encarrega de realizar comunicações entre as instâncias, utilizando os getters. Podemos pensar no Controller como um portão de comunicação entre as classes.

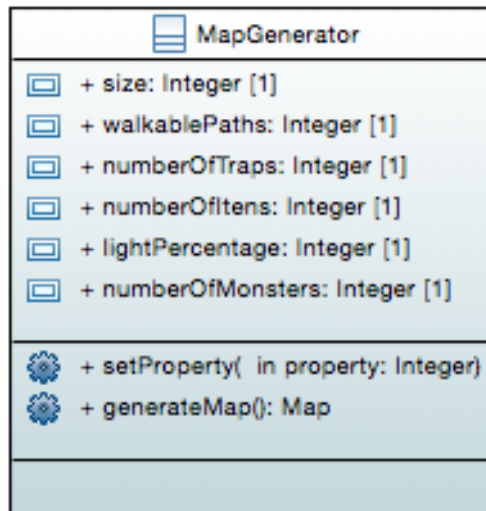
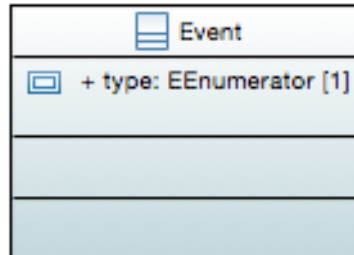
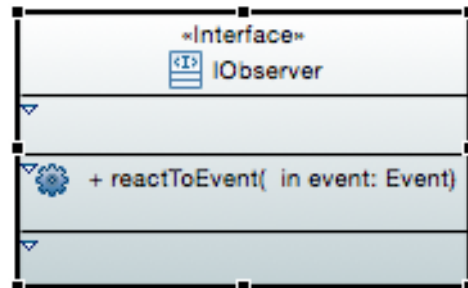
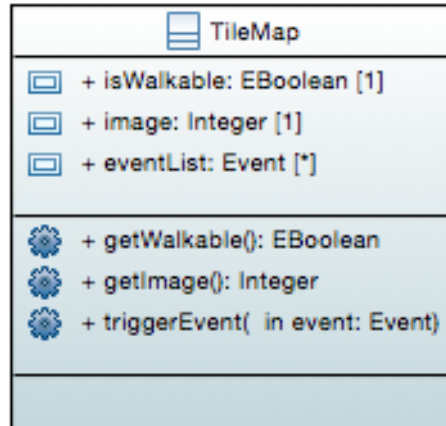
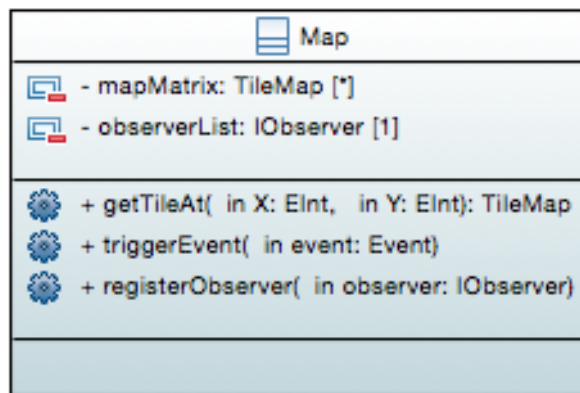
GameController

-  - MapInstance: IMapa [1]
-  - PlayerInstance: IPlayer [1]
-  - InputManagerInstance: InputManager [1]
-  - MonsterList: IMonster [1..*]
-  - gameController: GameController [1]

-  - GameController()
-  + getInstance(): GameController
-  + getMap(): IMapa
-  + getPlayer(): IPlayer
-  + getMonsters(): IMonster
-  + proceedTurns()
-  + main(in argc: EString)

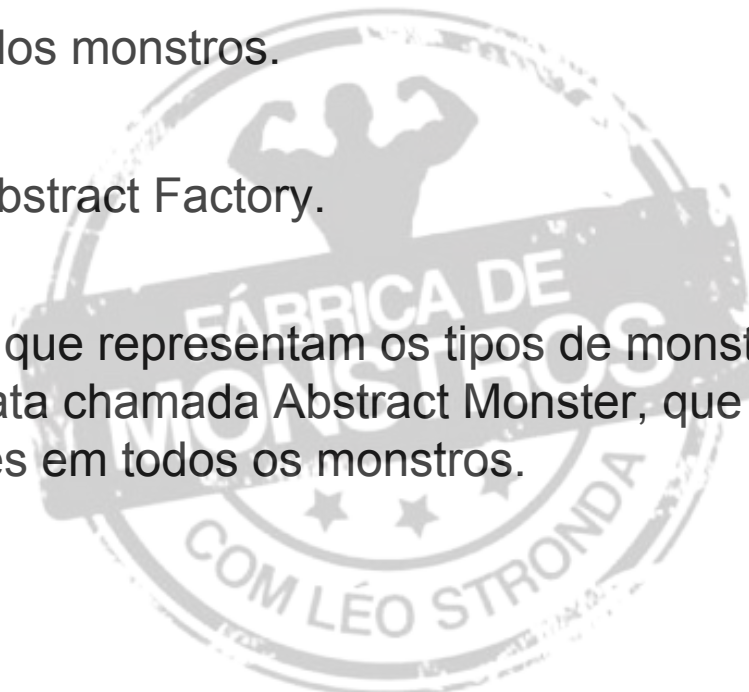
Mapas

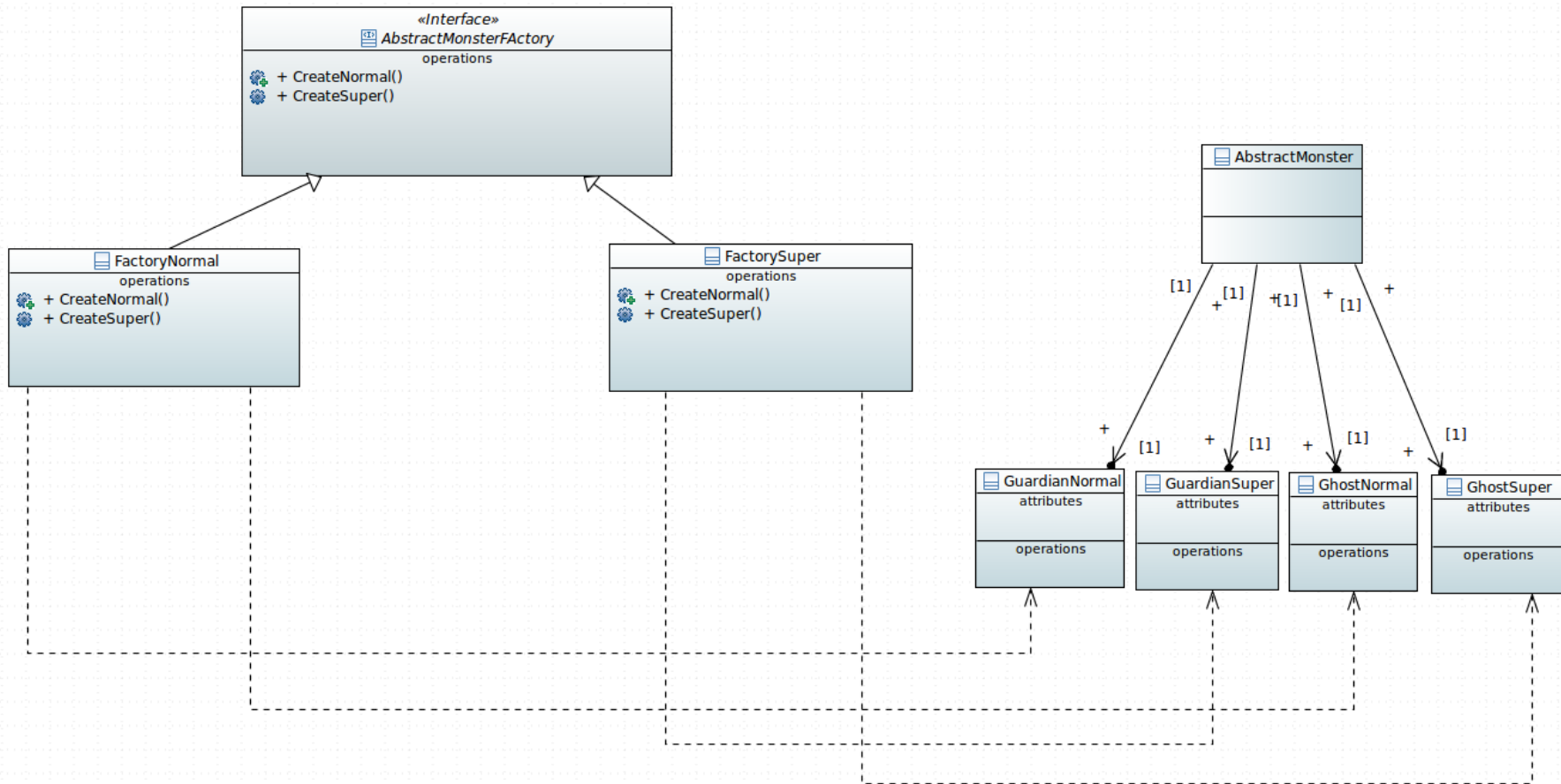
- Componente responsável tanto pela criação, armazenamento e gerenciamento dos mapas do jogo, quanto dos eventos gerados por este.
- Nesse componente foram usados patterns de singleton(para a classe MapGenerator que cria os mapas) e de Observer Pattern(Avisar a ativação dos eventos).



Monstros

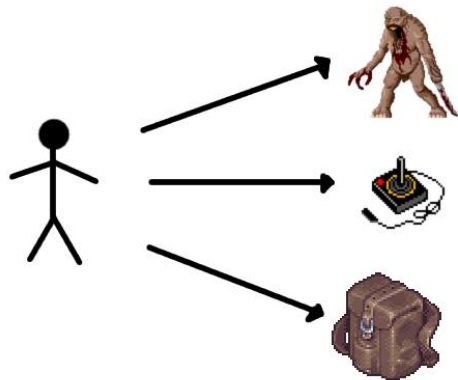
- Descrição breve dos monstros.
- Design Pattern: Abstract Factory.
- Todas as classes que representam os tipos de monstros são herdeiros de uma classe abstrata chamada Abstract Monster, que implementa alguns métodos presentes em todos os monstros.

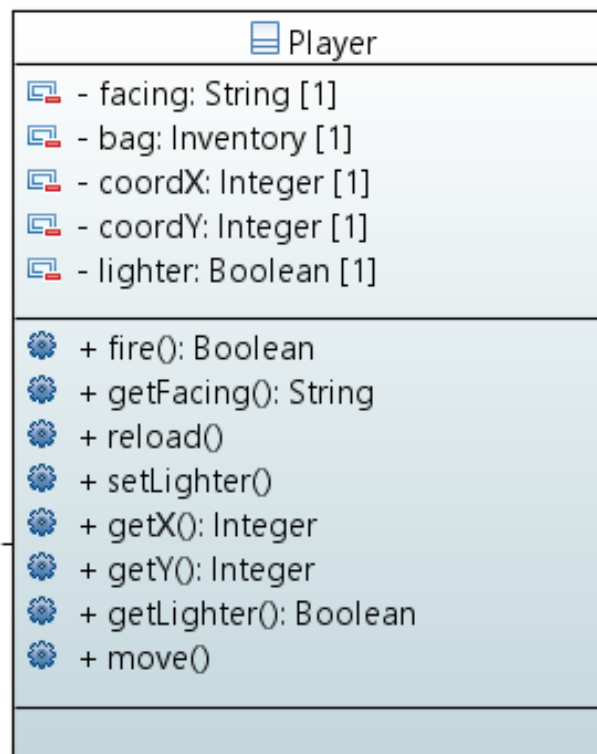
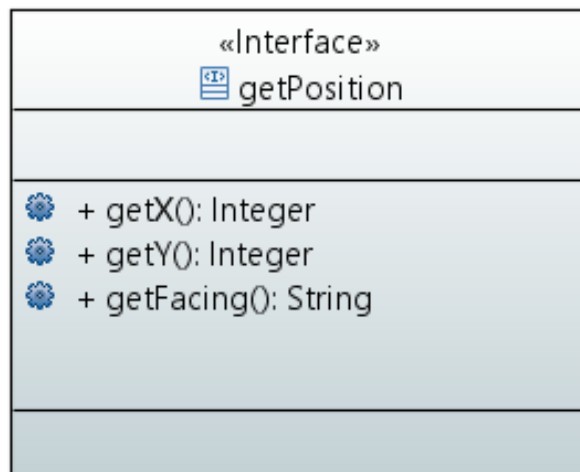
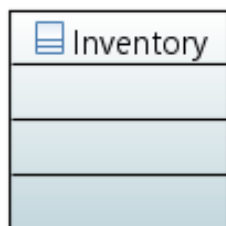




Player

- Componente responsável por armazenar os dados do jogador e realizar todas as suas possíveis ações.
- Nesse componente será usado o pattern Observer para comunicar às demais classes a ativação dos métodos.





Inventário

- Itens:
 - Flare, Fuel, PowerUp, SaltAmmo, Stick
- Inventário
- Design Pattern
 - Observer Pattern: todos os itens observam a classe de administração(ItemManagement)

