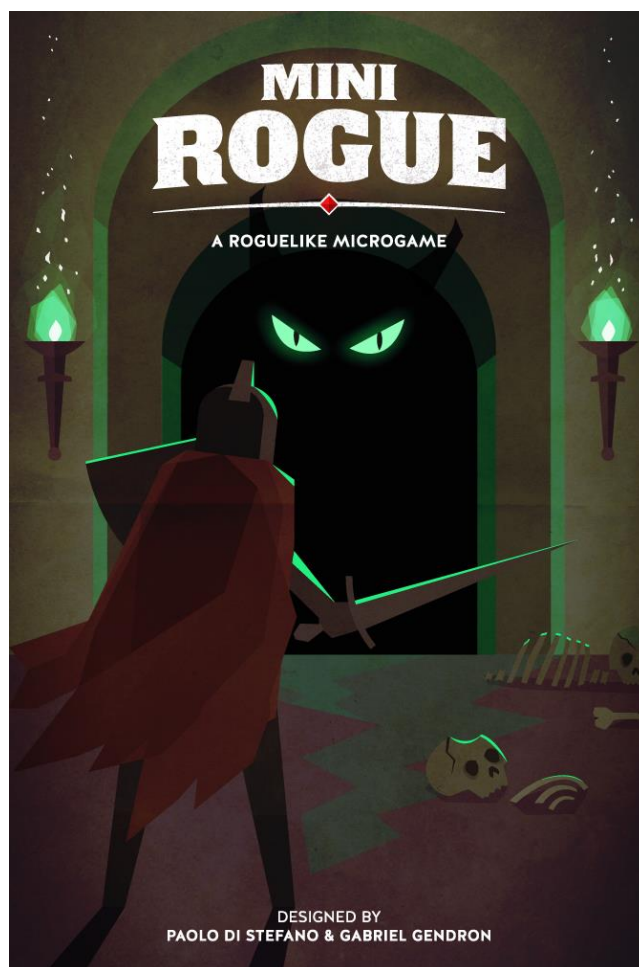


1/5/2017

MINI ROGUE

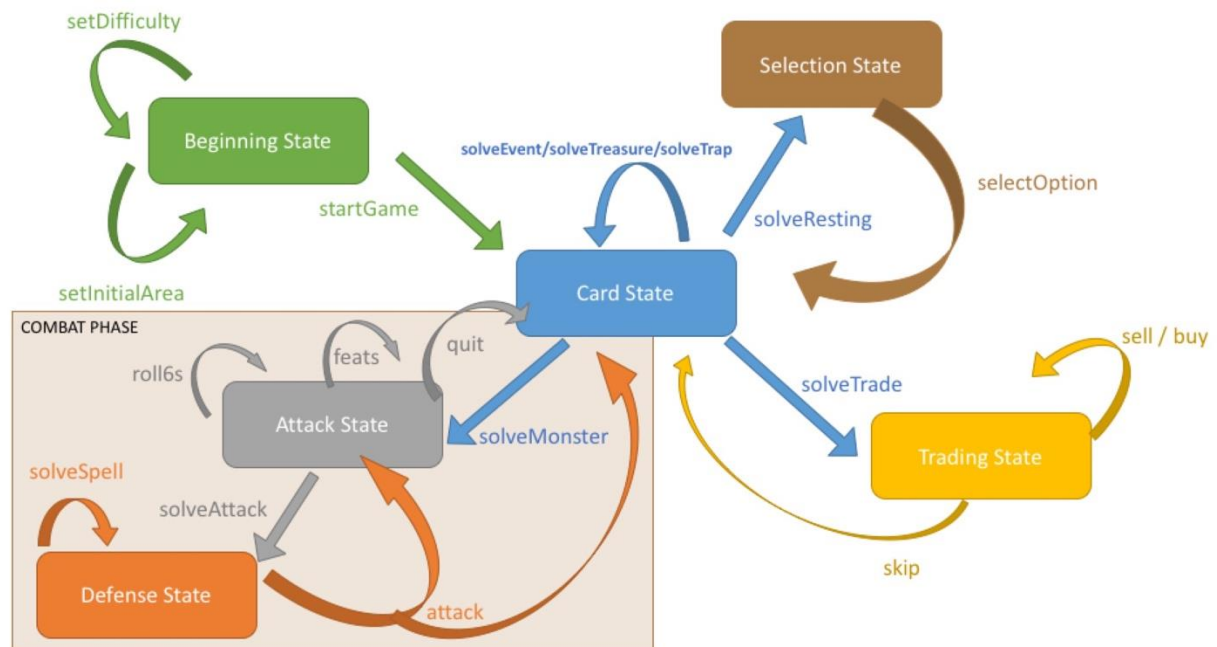
Programação Avançada (PA) – Meta 1



Tiago Miguel Matrola Simões

A21240385 - ANO LECTIVO 2016/2017

1. Diagrama da máquina de estados



2. Breve descrição das classes e pacotes

- PACOTES



Pacote UI

Este pacote contém a interface texto e a classe *Main*.



Pacote Logic

Este pacote contém a lógica do jogo.



Pacote Logic.states

Este pacote contém os estados do jogo.



Pacote Logic.dif

Este pacote contém todas as dificuldades do jogo.



Pacote Logic.cards

Este pacote contém as cartas do jogo.

- CLASSES



Classe Text

Esta classe corresponde à interface texto e contém as funções de interação com o utilizador (funções Scanner e System.out.print) e também é quem guarda o jogo.



Classe Dice

Esta classe sorteia números aleatórios.



Classe Dungeon

Esta classe representa a carta *Dungeon* e contém a área e a vida atuais do monstro.



Classe Player

Esta classe representa a carta *Character Stats* que contém todos dados do jogador e uma *flag* que informa o jogador se já matou o monstro ou não, quando passar pela carta Tesouro.



Classe GameData

Esta classe contém as classes *Player* e *Dungeon*, a coluna atual e o baralho de cartas. Ao início de cada área a classe *GameData* obtém da classe *Dungeon* uma variável, que informa se esta é a última área. Caso isso se verifique adiciona uma carta Boss ao baralho original.



Classe MiniRogue

Esta classe recebe pedidos da interface texto e controla todo o fluxo dos estados do jogo.



Classe Card

Esta classe é a classe base de todas as cartas do baralho.



Classe Difficulty

Esta classe contém os parâmetros dos dados do jogador, obtidos através das suas subclasses.



Interface IStates

Esta classe contém todos os métodos de transição para os próximos estados.



Classe BeginnigState

Este estado aguarda pelo o inicio do jogo.



Classe CardState

Este estado aguarda pela escolha da seleção da próxima carta.



Classe SelectionState

Este estado aguarda pela decisão do utilizador de qual a opção que deve escolher na carta *Resting*.



Classe TradingState

Este estado aguarda pela decisão do utilizador de qual a opção que deve escolher na carta *Treasure*.



Classe AttackState

No caso deste estado ter sido criado inicia a vida do monstro, cria os dados, e aguarda pela decisão do utilizador. Se este pretender voltar a lançar os dados, tem as opções de rolar seis ou de executar um *feats*. Quando pretender executar o ataque, o seu nível de ataque é transmitido na criação da classe *Defense*.



Classe DefenseState

Este estado aguarda pela decisão do utilizador em relação ao spell que pretende usar, para reforçar o seu nível de ataque e atacar o monstro. Se o monstro estiver vivo, este estado volta ao estado anterior (*AttackState*), caso o monstro esteja morto este retorna ao estado *CardState*, e caso o jogador morra volta ao estado *BeginningState*.



Classe StateAdapter

Esta classe implementa a interface *IStates* e tem redefinidos todos os métodos.