Relatório Projeto Final

Daniel Cipriano RA 233228
Lucas Guarnieri RA 119756
Ricardo Bello RA 247326

Para o projeto final da disciplina de programação orientada a objetos foi proposto que fosse criado um jogo emulando os jogos da conhecida franquia Pokémon, utilizando os conceitos de POO aprendidos durante o curso. Esse relatório tem por objetivo clarificar e explicar algumas decisões tomadas durante a construção do código.

Para representarmos o mapa presente no jogo foi utilizado uma matriz de objetos do tipo Entity. Para usarmos apenas uma matriz para guardar todos os objetos, consideramos todos os objetos “físicos” como subclasse da superclasse Entity, visto que desde o jogador até os itens, eles compartilham um atributo em comum que é de posição no mapa. A priori foi considerada a possibilidade da classe entity ser uma interface, para respeitar o princípio de “depend on abstractions, not on constructs”, entretanto devido aos inúmeros objetos colocado sob a classe Entity, uma abstração geral se mostrou contra produtiva e preferirmos a utilização de uma superclasse que contasse apenas com um parâmetro: posição.

Entre outras escolhas feitas para implementar os preceitos de POO, a utilização da classe feita para simular o jogar de um número n de dados de x faces para obter probabilidades talvez seja a mais controversa na construção do código, entretanto para promover a reutilização de código seguimos com a decisão. Então uma probabilidade de 30% pode ser tomada como a probabilidade de jogar um dado com 3 faces e dar o número um. Esse caso também exemplifica a controvérsia: a probabilidade de qualquer face de uma dado de 3 faces cair é 33,33....%, não exatamente 30%. Mas consideramos que a diferença entre probabilidades não é significativa o suficiente para alterar os requerimentos do projeto.

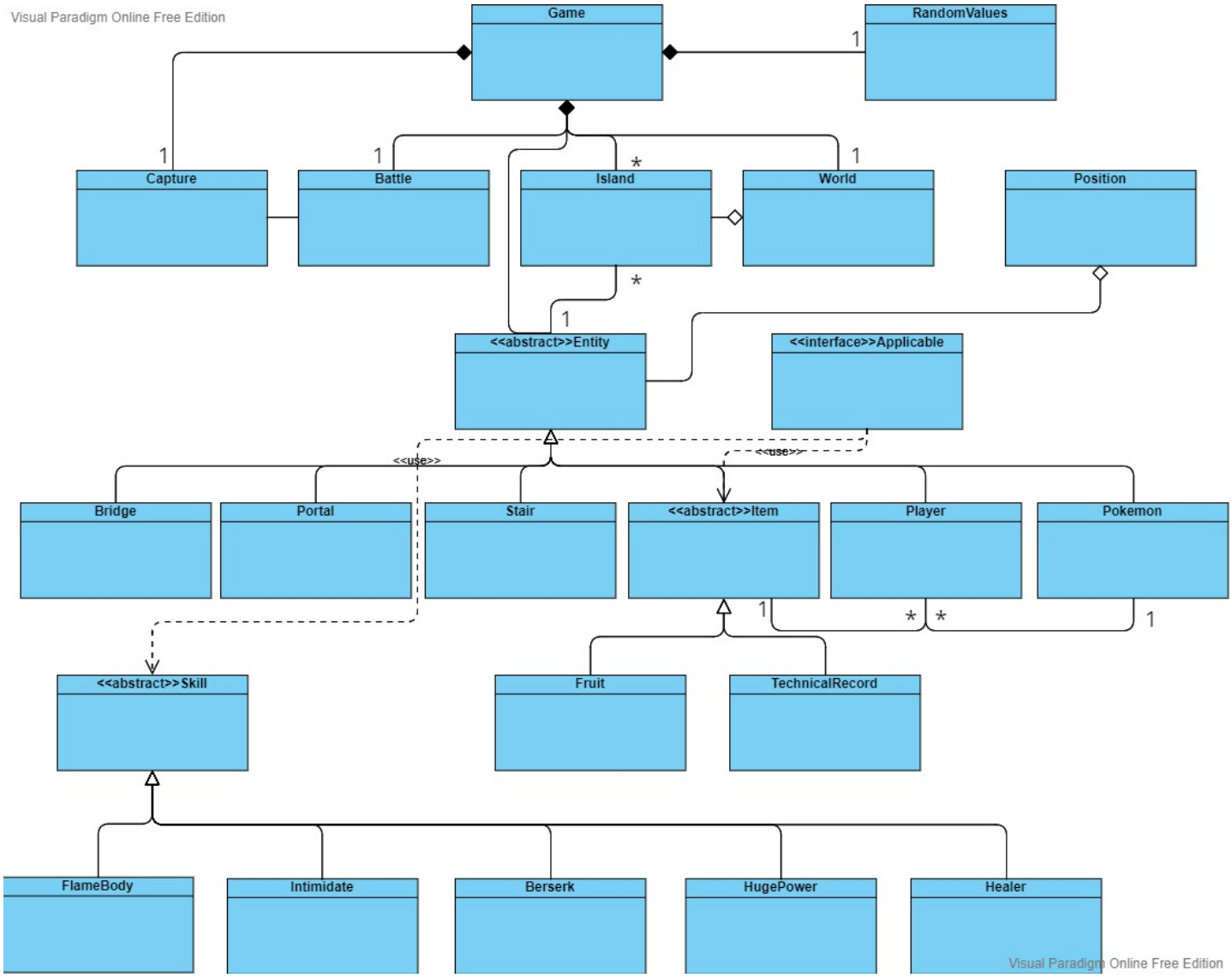
Em relação a criação de ilhas via arquivos de texto, há funções disponíveis para se construir qualquer tipo de ilha desejado, mas optamos por dar ao jogo um mapa default caso o objetivo seja apenas jogar o jogo. Os requerimentos do jogo também pediam por uma opção para se jogar em um mundo randomizado. Devido a dificuldades de implementação a função ficou inacabada. O programa foi construído com a premissa que um usuário iria apresentar os inputs para construção de um mundo, o que tornou difícil a implementação “automática” dessa característica.

Foram adicionadas duas exceções personalizadas no projeto. A exceção InsufficientCaptureDistanceException é lançada quando a distância de captura entre o player e o pokemon é insuficiente, ela detalha a distância necessária para captura e a distância obtida. A exceção InvalidInputOptionException é lançada quando existe uma entrada de dados inválida com as opções do jogo, foi feita de forma a ser estendida para situações em diferentes áreas de controle do jogo.

Foi adicionada a interface Applicable que é implementada pelas classes que possuem métodos que aplicam efeitos, como os que herdam Item e Skill. Além disso, optou-se por utilizar a enumeração, Types, para listar os tipos possíveis para os pokémons.

O sistema de movimentação do jogo utiliza a base das teclas W(cima), S(baixo), A(esquerda), D(direita). O método toString() de cada objeto que herda Entity foi sobrescrito para retornar um prefixo para cada tipo de objeto, seguindo, Pokemon(PK), Player(P), Item(I), Bridge(B), Portal(PO), Stair(S). Foi implementado o método printMap que percorre a matriz representante da ilha, imprimindo os prefixos de cada objeto, ou, o caractere * caso a posição não possua objetos.

Abaixo segue um diagrama simplificado representando as principais classes do programa e as relações entre si.



A maioria das classes foi planejada antecipadamente para atender ao máximo os preceitos de POO vistos neste semestre. Apesar do planejamento, algumas concessões foram feitas, vide exemplo do da classe abstrata Entity, que inicialmente foi pensada como uma interface. O código foi também escrito considerando a possibilidade de adição futura de novas funcionalidades, como por exemplo novas habilidades ou novos itens.