

|  |
| --- |
| Programação Orientada por Objetos 2020/2021  Licenciatura em Engª. Informática |
| 1ª Fase |

**Nome**: Tomás Bota

**Nº**: 200221138

**Nome**: Rafael Maria

**Nº**:200221055

**Turma**: AGR-053

**Docente de Laboratório**: João Capinha

**Índice**

1. Introdução------------------------------------------------------------------------------------------ 3

2. Análise de Substantivos/verbos----------- -----------------------------------------------------3

2.1. O Jogo

2.2 Jogadores

3. Cartas CRC -------------------------------------------------------------------------------------------4

3.1. O Jogo

3.2 Jogadores

4. Diagrama de classes -------------------------------------------------------------------------------4

5. Conclusão --------------------------------------------------------------------------------------------4

1. **Introdução**

Neste relatório encontra-se as informações referentes ao projeto de Programação Orientada por Objetos, do ano letivo 2020/2021, referente ao desenvolvimento de uma versão do jogo Boats and Docks.

Nesta 1ª fase encontra-se a análise dos verbos e substantivos, das respetivas cartas CRC e do diagrama de classes.

1. **Análise de Substantivos/verbos**

O Jogo

O objetivo deste jogo passa por identificar todas as casas do tabuleiro de jogo, de forma a obter a maior pontuação possível, por nível. Quando o jogo inicia, apresenta todas as casas do tipo porto, cabe ao jogador determinar quais as casas do tipo água e barco. Existem três dimensões de barcos diferentes, sendo estes pequenos, médios e grandes. O jogo termina quando cada porto tiver um barco atracado, seguindo as respetivas regras. Apenas quando colocados todos os barcos, o jogador pode validar o tabuleiro. Quando validado, todas as casas desconhecidas passam automaticamente ao estado água. Se o tabuleiro estiver correto, o jogo termina e é calculada a pontuação obtida. Caso contrário, o jogador é informado do número de erros existentes no tabuleiro. As casas erradas no momento da validação do tabuleiro de jogo, são declaradas como falhadas. Uma casa, uma vez falhada já não voltará a alterar o seu estado. Se a casa falhada for do tipo barco, as restantes casas que compõem o barco (caso existam), devem ser marcadas igualmente como falhadas. No final do jogo, é contabilizado o número total de casas falhadas e esse valor reflete uma penalização na pontuação do nível.

Substantivos

* Jogo
* Casas
* Tabuleiro
* Nível
* Porto
* Jogador
* Água
* Barco

Verbos:

* Identificar
* Obter
* Apresentar
* Determinar
* Existir
* Terminar
* Atracar
* Colocar
* Validar
* Passar
* Calcular
* Informar
* Declarar
* Voltar
* Marcar
* Contabilizar

Jogadores

O jogo é jogado individualmente, por um jogador. Este jogador é identificado por um nome (nickname), introduzido pelo próprio jogador, que deverá ser único. Cada jogador deve manter um histórico de pontuações obtidas, para todos os jogos efetuados pelo próprio, com registo de data e hora do jogo. Deve ser possível aceder à pontuação máxima obtida pelo jogador, em cada um dos níveis. Deve ainda, ser apresentado um ranking, onde apresenta todos os jogadores, ordenados de forma decrescente, com base na pontuação acumulada de cada.

Substantivos

Jogo

Jogador

Nome (nickname)

Pontuações

Próprio

Data e Hora

Nível

Ranking

decrescente

Pontuação máxima

Verbos

Jogado

Identificado

Introduzido

Deverá

Manter

Histórico

Efetuados

Aceder

Apresentado

Ordenado

Registo

Base

1. **Cartas de CRC**

O Jogo

Cartas CRC:

* Jogo:
  + Tem um tabuleiro
  + Termina quando cada porto tiver um barco atracado
  + Está responsável de fazer a contabilização dos pontos de cada jogador no final do jogo
  + Colaboradores:
    - Tabuleiro
    - Jogador
* Casas:
  + Têm diferentes tipos possíveis
  + Fazem parte do tabuleiro
  + São validadas pelo jogador
  + São marcadas como falhadas caso haja um erro de validação
  + Colaboradores:
    - Tabuleiro
* Tabuleiro:
  + Tem uma coleção de casas.
  + Pode ser validado pelo jogador
  + Colaboradores:
    - Jogo
    - Casas
* Nível:
  + Cada nível tem tabuleiros diferentes
  + Cada nível tem um número de casas de cada tipo diferentes
  + Cada nível dá pontuações base diferentes ao jogador
  + Um jogador pode jogar vários nível e registar pontuações diferentes para cada nível.
  + Colaboradores:
    - Tabuleiro
    - Casas
    - Jogo
    - Jogador
* Jogador:
  + Cada jogador pode jogar diferentes níveis
  + Cada jogador vai ter a pontuação registadas que conseguiu num determinado jogo
  + Cada jogador pode criar níveis determinando a posição das casas inicias
  + Colaboradores:
    - Jogo
    - Tabuleiro
    - Casas
    - Nível

Jogadores

|  |
| --- |
| **Jogador** |
| Tem um nome (único)  Tem um histórico de pontuações |

|  |
| --- |
| **Histórico de Pontuações** |
| Conjunto de pontuações de um jogador |

|  |
| --- |
| **Pontuação** |
| Valor que o jogador obteve num nível  Data da aquisição dessa pontuação |

|  |
| --- |
| **Ranking** |
| Ordena de forma decrescente (Soma das pontuações de cada Jogador) |

|  |  |
| --- | --- |
| **Jogador** | **Colaboradores** |
| Criar jogadores com (Nome do Jogador)  Lista de Pontuações ->(Histórico de pontuações) | Pontuações  Histórico de Pontuações |

|  |  |
| --- | --- |
| **Ranking** | **Colaboradores** |
| Pega numa lista de jogadores  Soma as pontuações de cada jogador  Ordena de forma decrescente o Jogador com a maior pontuação | Jogador  Pontuações |

|  |  |
| --- | --- |
| **Histórico de Pontuações** | **Colaboradores** |
| Lista das pontuações de um Jogador | Pontuações  Jogador |

|  |  |
| --- | --- |
| **Pontuações** | **Colaboradores** |
| Contém o nível  A pontuação obtida  (associado a um Jogador) | Jogador |

1. **Diagrama de Classes**

|  |
| --- |
| <<Enum>>  **TileType** |
| Strings  Valores: DESCONHECIDA, PORTO, ÁGUA,BARCO |
|  |

|  |
| --- |
| **BarcoTile** |
|  |
|  |

|  |
| --- |
| **AguaTile** |
|  |
|  |

|  |
| --- |
| **DeconhecidaTile** |
|  |
|  |

|  |
| --- |
| **PortoTile** |
|  |
|  |

|  |
| --- |
| <<Abstract>>  **Tile** |
| -x: int  -y: int  -failed:boolean |
|  |

|  |
| --- |
| **Board** |
| -Tiles: List<Tiles> |
|  |

|  |
| --- |
| **Game** |
| -dificultyLevel: DificultyLevel  -timeExpectancy: int  -basePoints: int  -board: Board  -levels: ArrayList<LevelInfo>  -user: Player |
|  |

|  |
| --- |
| <<Enum>>  **DificultyLevel** |
| Strings  Valores: FÁCIL, INTERMÉDIO, DIFÍCIL |
|  |

|  |
| --- |
| <<Enum>>  **BoatSize** |
| Strings  Valores: PEQUENO, MÉDIO, GRANDE |
|  |

|  |
| --- |
| **Pontuation** |
| -time: LocalDateTime  -gamepoints: int  -level: int |
|  |

|  |
| --- |
| **Player** |
| -username: String  -gamePontuation: HashMap<Integer, Pontuation>  -maxPontuation: int |
|  |

|  |
| --- |
| **RANKING** |
| - ranking: ArrayList<Players> |
|  |

1. **Conclusão**

Com a conclusão desta fase do projeto conseguimos preparar cerca de 90% do trabalho das próximas fases. Podemos perceber a importância de uma boa fase de design, concessão, desenvolvimento e discussão para que as fases seguintes e para que a implementação de funcionalidades seja tranquila e sem grandes percalços ou alterações significativas ao projeto.

Embora tenha corrido sem problemas de maior, achamos que o enunciado não era muito explicito em algumas partes do mesmo, nomeadamente no que era necessário fazer nesta fase. Conseguimos perceber que, obviamente, o design e concessão das classes pertencem a esta fase mas ficámos na duvida se partes como, desenvolvimento de métodos que abordem certas funcionalidades do jogo, fazem parte desta fase do projeto ou não.

No entanto, e à exceção desta pequena dificuldade, achamos o projeto bastante interessante e achamos a ideia fixe e uma boa ideia para um projeto deste género tendo em conta os conhecimentos que fomos adquirindo.