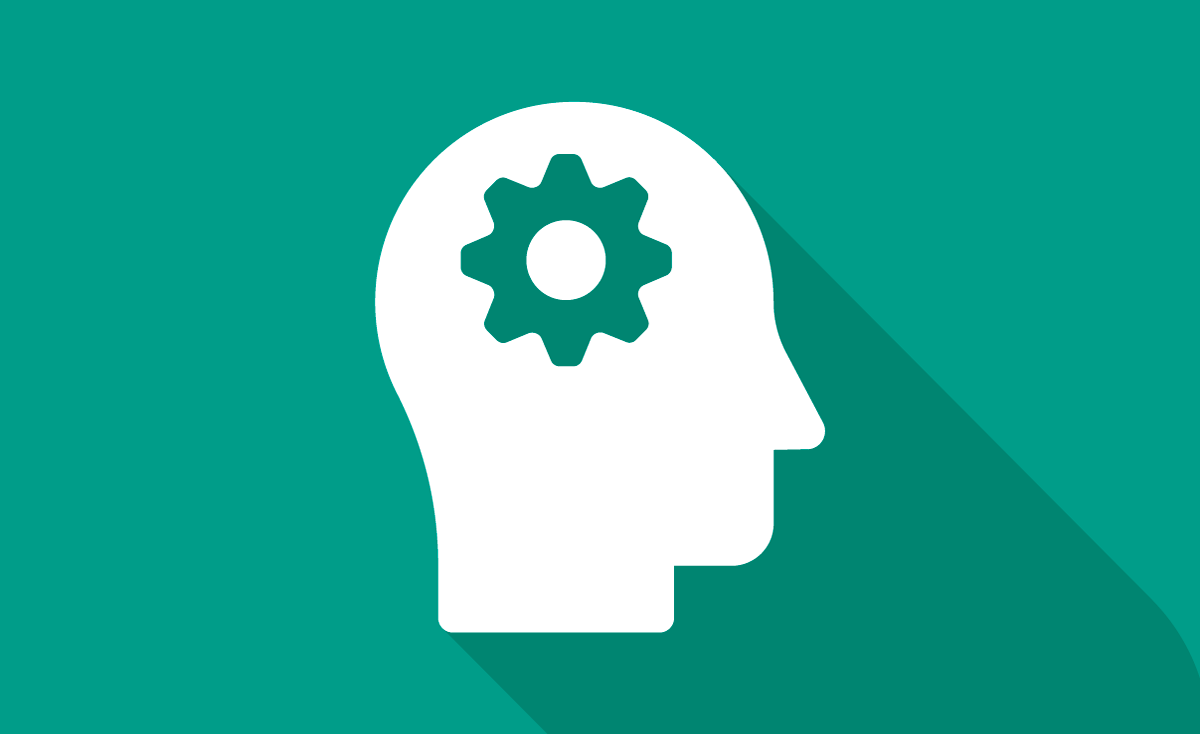


**Programação**

**Orientada a Objetos**

**POKOBAN**

****

**logotipo do projeto**

**Índice**

1. Grupo de Trabalho 3

1.1 Declaração: 4

2. Descrição do projeto 5

2.1 Definição do SOKOBAN 5

3. Arquitetura da aplicação 6

3.1 Constituição e descrição do projeto 6

4. Manual do Utilizador 7

5. Limitações e Desenvolvimentos Futuros 7

6. Conclusão 7

6.1 Conclusão do Alexandre 7

6.2 Conclusão do Bruno 8

7. Referências 8

# Grupo de Trabalho

|  |  |
| --- | --- |
| FOTO | Número: 80662  Nome: Alexandre Santos Coelho  Curso: Licenciatura em Engenharia Informática  Turma: C  Email: [alexfratzsantos@outlook.com](mailto:alexfratzsantos@outlook.com) |
| FOTO | Número: 13370  Nome: Bruno Remos  Curso: Licenciatura em Engenharia Informática  Turma: C  Email: |

## Declaração:

Os alunos declaram sob compromisso de honra que o projeto final é original e da sua autoria com exceção dos seguintes elementos:

|  |  |
| --- | --- |
| **Descrição** | **Fonte** |
| Biblioteca de parsing para JSON | Diretório de Maven |
| Imagens usadas para boneco, fundos, e objectos | Pesquisa no Google |
|  |  |

# Descrição do projeto

## Definição do SOKOBAN

O trabalho pratico desta unidade curricular consistiu no desenvolvimento de uma aplicação em linguagem Java, esta aplicação tem como objetivo permitir que o seu utilizador possa jogar um clone do jogo SOKOBAN.

O SOKOBAN é jogado num tabuleiro de quadrados, onde cada quadrado é pode ser zona livre (referida como chão, onde o jogador pode navegar) ou limites (zona obstruída, geralmente são obstáculos ou paredes). O jogador irá estar dentro dos limites (que tem como principal função manter o jogador dentro do tabuleiro), apenas se pode movimentar horizontalmente ou verticalmente, existem alguns obstáculos que são movíveis, é através da interação com estes obstáculos que o jogador poderá completar o puzzle (objetivo principal do SOKOBAN). Para completar estar puzzle existem dois “marcadores” distintos, os obstáculos movíveis e os quadrados de marcação, estes são quadrados do tipo zona livre, que se diferenciam pelo seu aspeto e também pela sua função.

Para completar o jogo o jogador deve mover todos os obstáculos movíveis para cima dos quadrados de marcação, apenas quando todos os quadrados de marcação forem sobrepostos por obstáculos movíveis é que o puzzle se considera completo.

* Projetos similares (apresentar hiperligações e imagens).
* Enquadramento e pertinência do projeto na disciplina

# Arquitetura da aplicação

## Constituição e descrição do projeto

Na solução do projeto resultou uma aplicação com o nome de “POKOBAN”, decidimos este nome porque o tabuleiro tem uma arte de imagens parecida com a do jogo POKEMON que se jogava num gameboy na altura da nossa infância. Esta aplicação está organiza em cinco *packages* principais, o vou passar a explicar o conteúdo e função de cada um em baixo:

* *Package* “Controlers” – neste package deverão ser contidas todas as classes que controlem as comunicações/interações com o exterior do ambiente (neste caso, a aplicação).
  + Classe “FileReader.java” – esta classe tem como função criar, escrever, e ler ficheiros, neste caso, o escopro de ficheiro com que esta classe interage é reduzido a dois ficheiros, ambos com o formato de JSON, um dos ficheiros contem a formatação dos nossos tabuleiros e alguma informação própria do tabuleiro, cada tabuleiro corresponde a um nível. O segundo ficheiro é o que se costuma chamar de save file, a sua função é conter o estado do jogo a quando o utilizador decido guardar o seu progresso, e também o estado inicial do nível em questão para que se possam resolver alguns problemas que sejam suscetíveis de aparecer.
* *Package* “JARS” – neste *package* estão vários ficheiros do tipo JAR, que servem como suporte, para a leitura, escrita e parsing de JSON para objetos em memória em vice-versa. Não vou descrever o objetivo de cada JAR mais abaixo pois, não somos os seus autores, e como referido noutra altura neste relatório são ficheiros obtidos no diretório de Maven que decidimos utilizar para facilitar a introdução de ficheiros do tipo JSON no nosso projeto.
* *Package* “Resources” - neste *package* estão inseridas todas as imagens que são utilizadas para desenhar os jogadores, os estados dos jogadores (num sub *package*), chão, obstáculos e objetivos, é aqui também que mantemos o ficheiro que contem a representação de cada tabuleiro do POKOBAN.
  + Imagem “*FateX.png*” - este ficheiro corresponde à representação gráfica dos objetivos do jogo;
  + Imagem “*FloorX.png*” - este ficheiro corresponde à representação gráfica do chão dos nossos tabuleiros;
  + Imagem “*ObstacleX.png*” - este ficheiro corresponde à representação gráfica dos obstáculos dos nossos tabuleiros;
  + Imagem “*WallX.png*” - este ficheiro corresponde à representação gráfica dos limites dos nossos tabuleiros;
  + X – corresponde ao nível em questão, ex: nível 1 usará as imagens “Fate1.png”, “Floor1.png”, “Obstacle1.png” e “Wall1.png”.
  + *Package* “PlayerSprites”;
    - *Imagem “Y\_UpZ.png” -* este ficheiro corresponde à representação gráfica do jogador nos nossos tabuleiros.
    - *Imagem “Y\_UpZ.png” -* este ficheiro corresponde à representação gráfica do jogador nos nossos tabuleiros.
    - *Imagem “Y\_UpZ.png” -* este ficheiro corresponde à representação gráfica do jogador nos nossos tabuleiros.
    - *Imagem “Y\_DownZ.png” -* este ficheiro corresponde à representação gráfica do jogador nos nossos tabuleiros.
    - Y – Corresponde ao jogador 1 ou jogador 2;
    - Z – Corresponde ao iterador que diferencia os estados do movimento do jogar. Ex: 1\_UP1.png será o jogador 1 com a perna direita a descrever o movimento.
* Package “” - ada
* Como se encontra estruturado o código da aplicação,
* Descrição geral das bibliotecas de classes (packages)
* Diagrama de classes
* Descrição detalhada de cada classe ou interface e justificação da sua existência:
  + Justificação da sua existência e a forma como foi construída (herança e/ou composição)
  + Descrição dos atributos
  + Descrição dos métodos

# Manual do Utilizador

* Como se utiliza a aplicação desenvolvida.
  + Imagem e descrição de cada janela da aplicação

# Limitações e Desenvolvimentos Futuros

* Quais as limitações que a aplicação tem.
* O que ficou por fazer.

# Conclusão

## Conclusão do Alexandre

No desenvolvimento deste trabalho aprendi principalmente a trabalhar com a framework de java, a minha experiencia ainda que pouca já tinha alguma solidez, mas em .net, logo a familiarização não foi difícil, mas foi certamente importante para solidificar conhecimentos específicos e criar confiança para futuramente fazer mais desenvolvimento em java.

Aprendi a importar bibliotecas externas para o meu projeto através de JARs, para que fosse possível usar a formatação JSON, nos ficheiros de gravação e storage do tipo de cada nível.

Finalmente gostaria de agradecer ao professor Paulo Santos que foi paciente com as minhas diversas perguntas e me ajudou a resolver problemas com JFrame no desenho de objetos, com a utilização da GUI do netbeans para controlar os comportamentos do JFrame e também a encontrar a solução para que o plugin easyUML fizesse a sua função.

## Conclusão do Bruno

# Referências

* Livros
* Documentos
* Referência web