

Diagrama de Classes da Modelação de Dados – MineSweeper

Engenharia Software

Ano Letivo: 2019/2020

Licenciatura em Engenharia Informática

Autor

Francisco José Félix Meireles nº 69174

Índice

1. Introdução	3
2. Objetivos da Etapa do Trabalho Prático	3
3. Desenvolvimento	4
3.1. Diagrama de Classes	4
3.2. Explicação do diagrama de classes	4

1. Introdução

Este relatório tem como objetivo principal que os alunos demonstrem ser capaz a subdivisão de um sistema em partes, mais especificadamente em classes, espera-se ainda que os alunos na conclusão desta fase do trabalho tenham solidificado os conhecimentos da programação orientada a objetos.

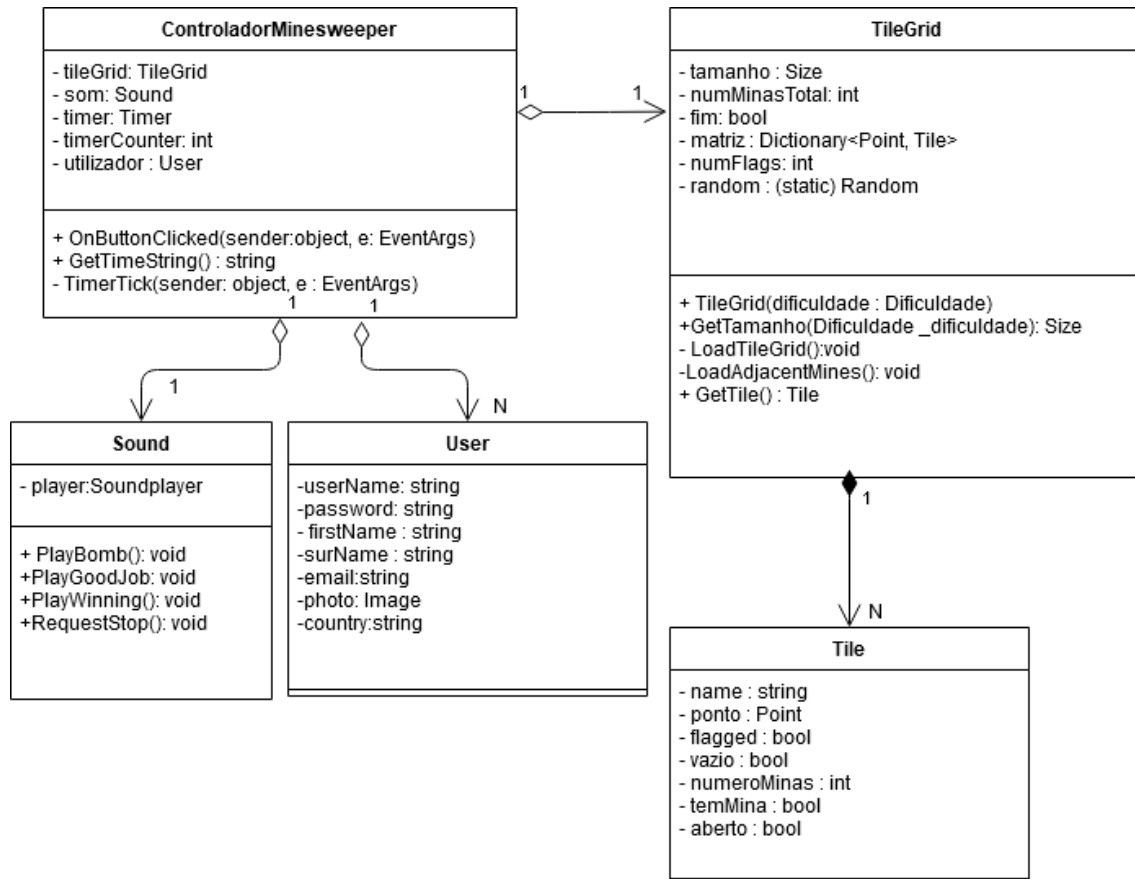
É necessário referir ainda que não chega apenas mostrar que estes conceitos foram assimilados, é também importante aplicá-los para o projeto desenvolvido mantendo sempre em mente que se pretende modelar o jogo “MineSweeper”. Todo este desenvolvimento começa pela análise dos requisitos funcionais e não funcionais, traduzindo-as em classes, associando-as de forma a que faça sentido, resultando em apenas num diagrama de classes em que seguida este será traduzido para código.

2. Objetivos da Etapa do Trabalho Prático

O objetivo desta etapa do projeto em desenvolvimento consiste por criar um diagrama de classes e uma sucinta explicação das suas relações. O desenvolvimento passa através da divisão do jogo Minesweeper em objetos que possuam características e comportamentos semelhantes, a partir dos quais criamos classes – diagrama de classes.

3. Desenvolvimento

3.1. Diagrama de Classes



3.2. Explicação do diagrama de classes

As classes constituintes do “Model” são: User, Sound, Tile, TileGrid.

Classe User:

A classe User tem como pressuposto guardar apenas a informação de login de cada jogador, é também guardado o perfil do utilizador, para posteriormente fazer o tratamento desses dados e podermos exibir sob a forma de uma lista, por exemplo sob uma lista dos top 10 dos jogos mais curtos, exibindo o tempo atingido assim como o utilizador que atingiu, dando igualmente ao utilizador a capacidade de premir (usando HyperlinkLabel) sobre o *username* para obter mais informações sobre o jogador que atingiu

Para cada individuo será guardado: o Nome abreviado, o Username, o Password, o Email, o Fotografia, o País.

Nesta etapa inicial, a password de cada individuo será guardado em “*plain text*”, uma vez que gerar e fazer a gestão de “*hashes*” afastam de um modo geral dos objetivos propostos para este trabalho.

A classe user de um modo geral é bastante simples, na medida em que apenas pretende-se que esta classe guarde informação sobre um jogador.

Abaixo encontra-se o protótipo desta classe.

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace MineSweeperProjeto.Model
{
    // Class Simples que permite guardar informações sobre o Vencedor
    internal class User
    {
        public string Username { get; set; }
        public string Password { get; set; }
        public string Firstname { get; set; }
        public string Surname { get; set; }
        public string Email { get; set; }

        public Image Photo { get; set; }
        public string Country { get; set; }

        public User()
        {
        }

        public User(string Username, string Password)
        {
            this.Username = Username;
            this.Password = Password;
        }

        public User(string Firstname, string Surname, string Email, Image
Photo, string Country)
        {
            this.Firstname = Firstname;
            this.Surname = Surname;
            this.Email = Email;
            this.Photo = Photo;
            this.Country = Country;
        }
    }
}
```

Classe Tile:

Tile é a unidade básica do Jogo, dentro de uma matriz, Tile corresponde a cada elemento da matriz, partindo disto, podemos inferir já que vamos necessitar de guardar as coordenadas dentro de matriz. É nesta classe instanciada que iremos guardar se a coordenada possui uma mina, o número de minas adjacentes, uma

variável bool que indica se a coordenada está vazia. Como já foi referido anteriormente é igualmente guardado o ponto de coordenada para efeitos de verificação de evitar erros na programação, o armazenamento do ponto não era estritamente necessário, uma vez que a gestão dos pontos poderá ser feito a partir da coleção utilizada (lista, dictionary, hashtable, etc.) numa outra classe com esse propósito.

Classe TileGrid:

TileGrid corresponde ao aglomerado de Tiles. É a matriz de objetos instanciados de Tiles.

Nesta classe será utilizado a coleção Dictionary, de forma simplista, a coleção dictionary é basicamente uma lista indexada. Esta coleção permite-nos armazenar pares de “key” + “value”, sendo que estes podem ser de qualquer tipo, é, portanto, genérica.

O melhor é que esta lista é guardada através da chave que lhe atribuímos, sendo muito fácil localizar qualquer item dentro desta coleção, extremamente útil para o que pretendemos desenvolver.

Classe ControladorMinesweeper:

Dado que o controlador se afasta de um modo geral dos objetivos propostos para esta etapa de desenvolvimento do projeto, esta classe não é explicada. Foi colocada no diagrama de uma forma a dar a que as classes do “Model” se pudessem relacionar.

4. Anexos

Ver projeto à plataforma GitLab pesquisando pelo username: @Eleusis300 ou pelo email: franciscomeireles-10@hotmail.com