

Tutorial do TiledMapEditor com LibGdx

Universidade federal Fluminense

Rio das Ostras – Abril de 2015

João Machado Da Silva Junior

Sumário

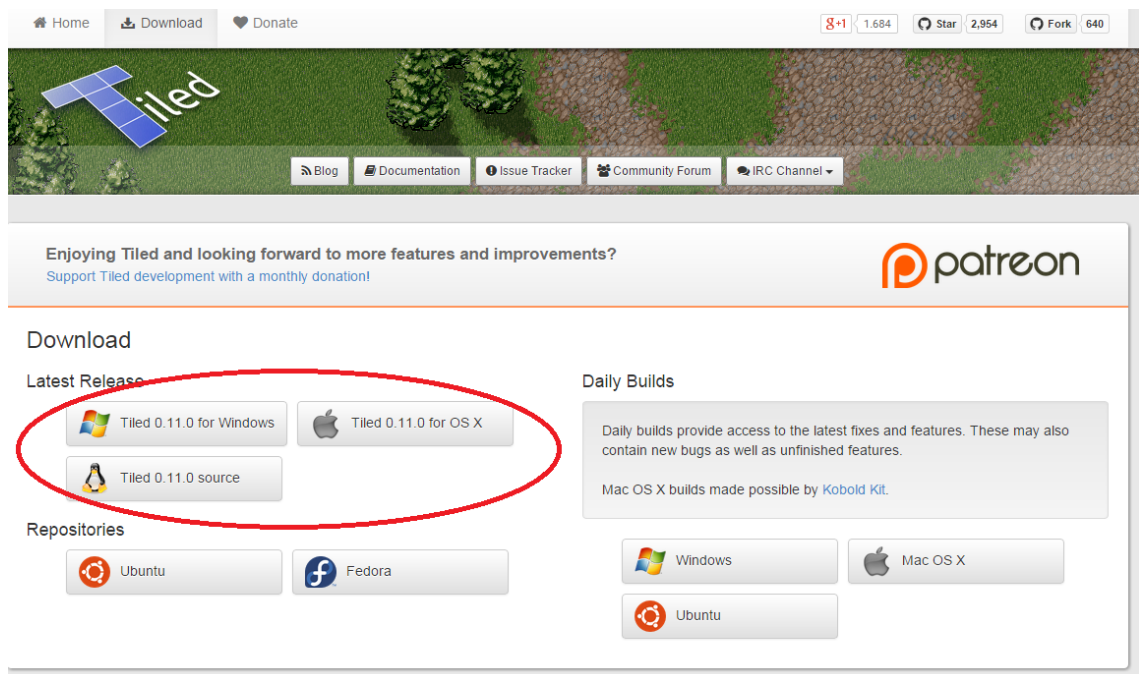
1- Introdução:.....	2
2- Download do TiledMapEditor.....	2
3- Instalação do TiledMapEditor.....	2
4- Criando um novo arquivo de mapa.....	5
5- Importando um TileSet.....	6
6- Criando um mapa simples.	7
7- Importando seu mapa para um projeto do LibGDX no Eclipse.	10
8- Criação de uma câmera:	12
9- Edição das propriedades do arquivo tmx.	14

1- Introdução:

O TiledMapEditor é um software de uso simples e intuitivo para a criação de mapas do tipo TiledMap que podem ser usados em jogos feitos com o LibGdx. A utilização do TiledMapEditor não requer nenhum conhecimento de programação, seu uso se assemelha com o uso de uma ferramenta de design de imagens, portanto basta ter criatividade. Nesse tutorial iremos apresentar como baixar o TiledmapEditor, como instalar, como criar um mapa simples, e como carrega-lo em um projeto libgdx usando o eclipse.

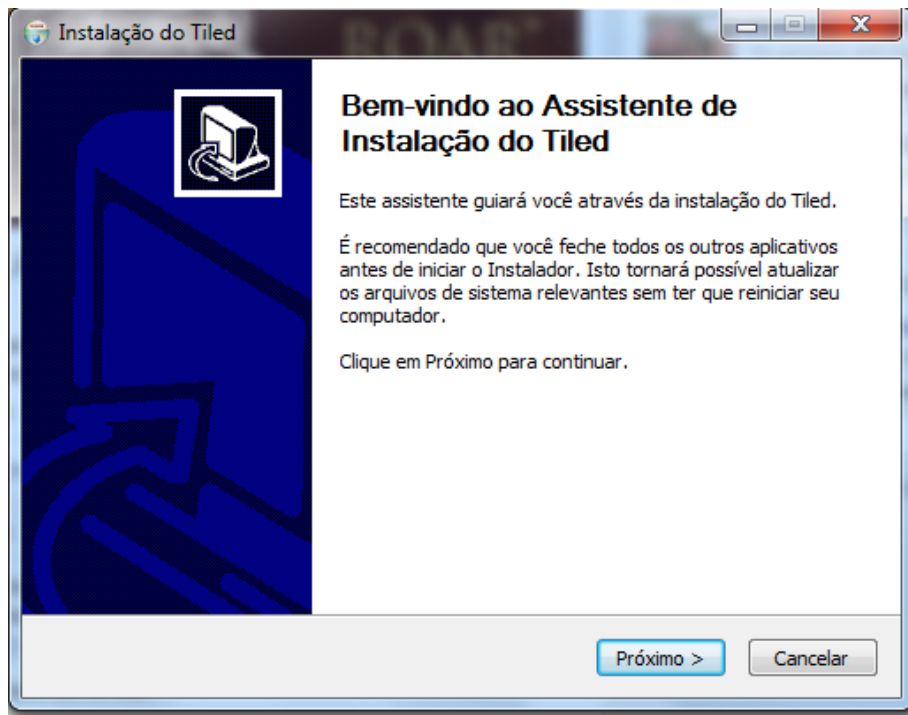
2- Download do TiledMapEditor.

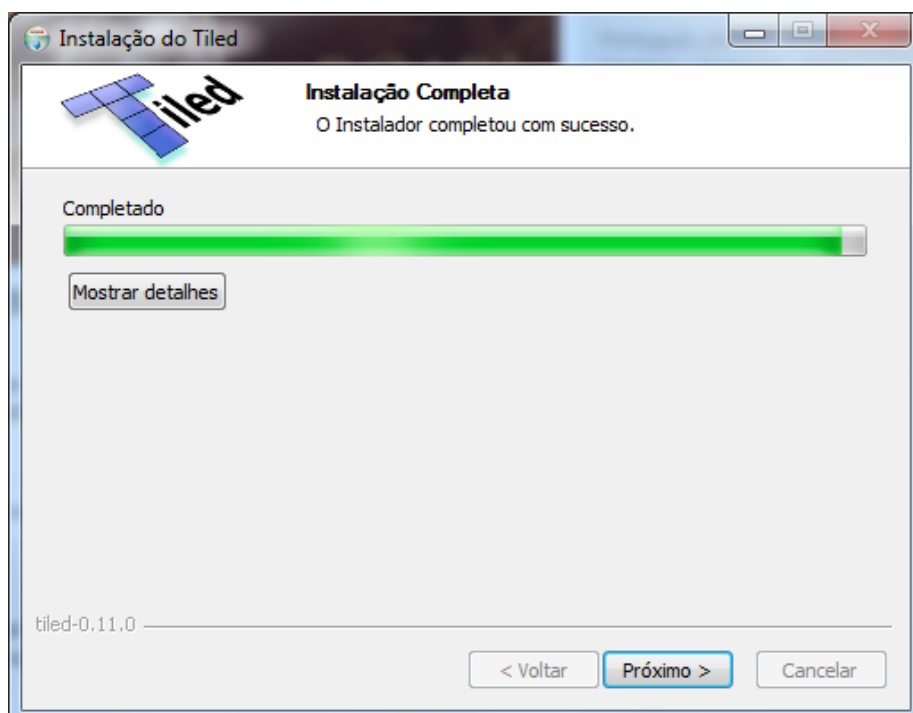
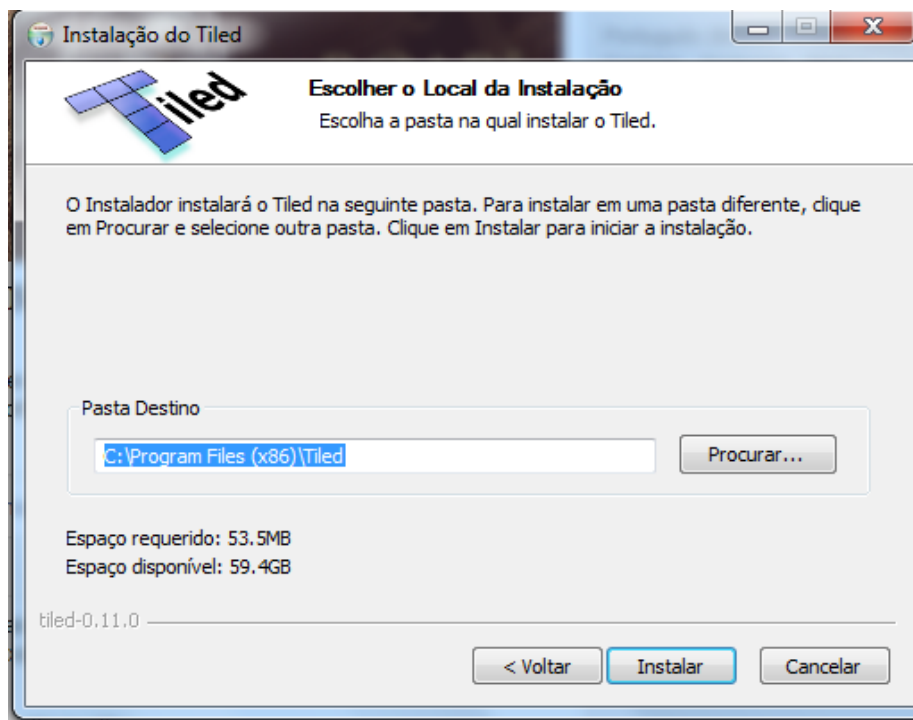
Você pode fazer o download do instalador do TiledmapEditor pelo seu site oficial (<http://www.mapeditor.org/download.html>).

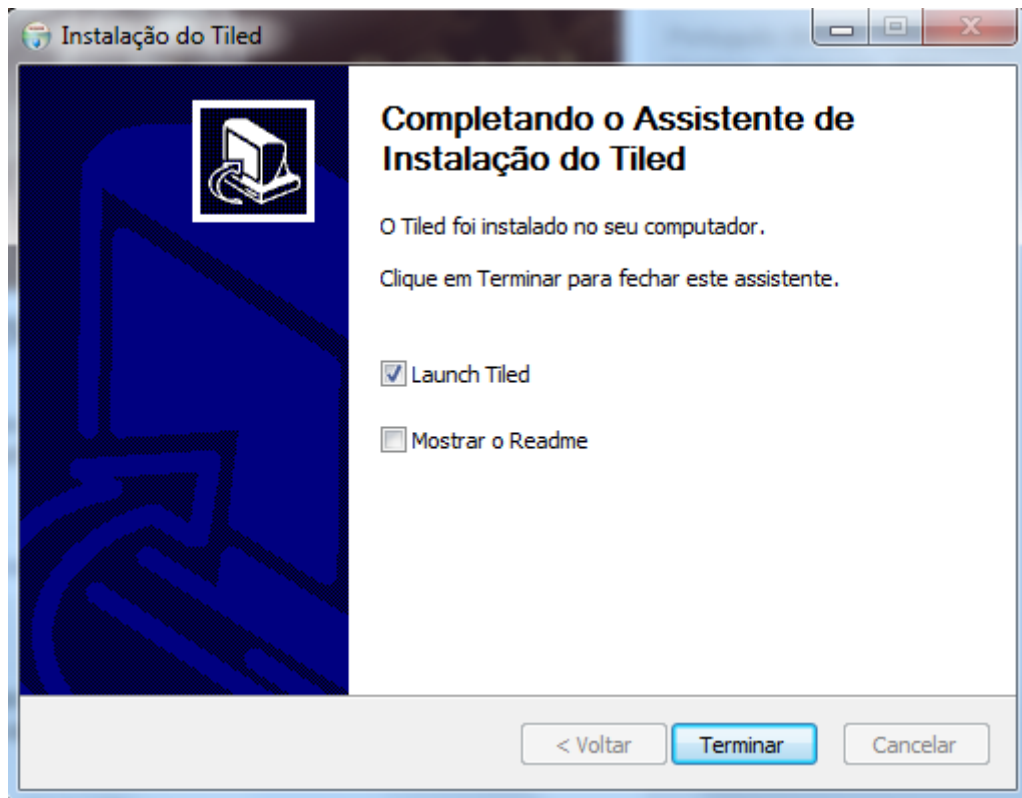


3- Instalação do TiledMapEditor

A instalação do TiledMapEditor não possui segredos, basta avançar as telas deixando as configurações padrões que o programa oferece.



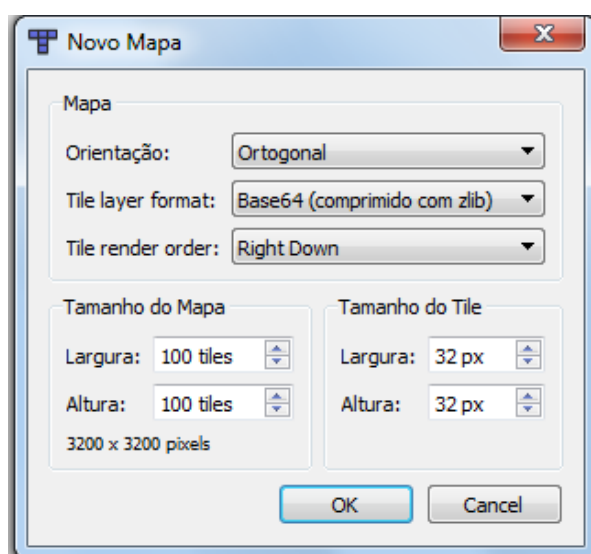




4- Criando um novo arquivo de mapa.

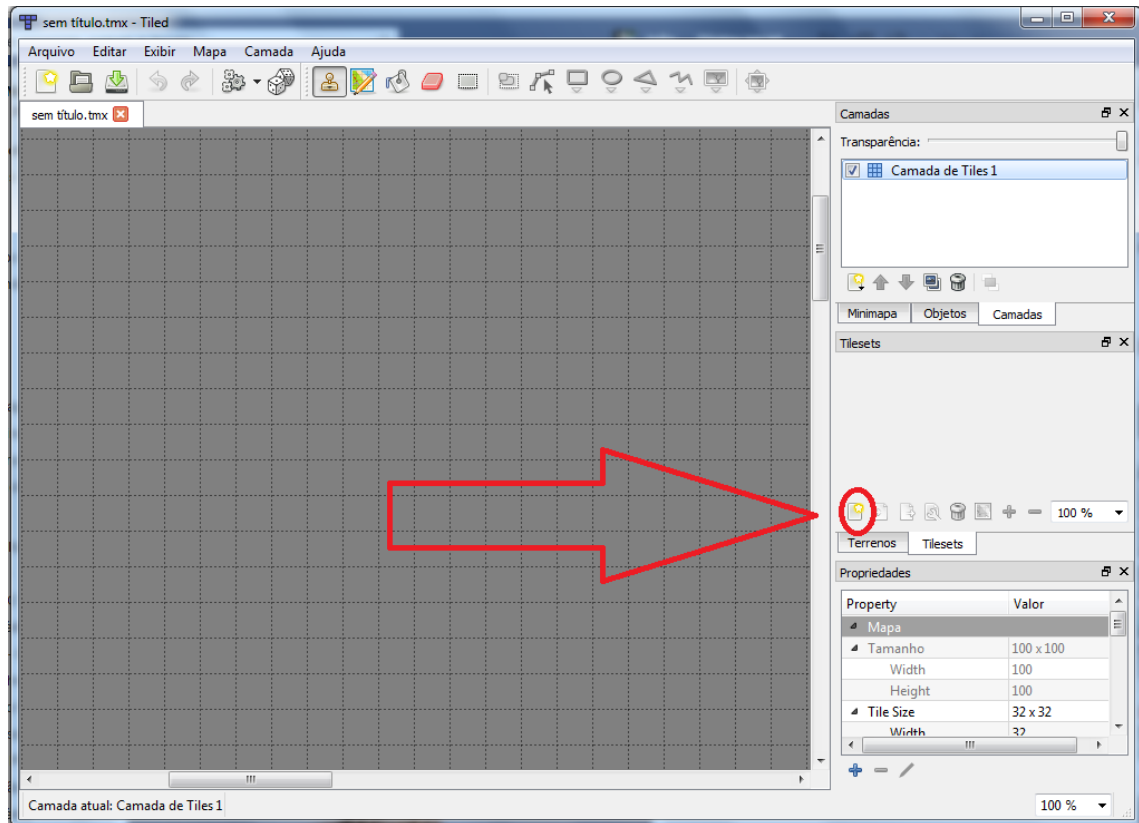
Primeiro, crie um novo arquivo. Pelo menu Arquivo->Novo. Irá surgir uma PopUp para que você escolha o tipo de mapa que desejar. Você deverá escolher a Orientação entre Ortogonal e Isométrica. Para maiores sobre Ortogonal/isométrico, leia esse artigo: (<http://www.significant-bits.com/a-laymans-guide-to-projection-in-videogames>).

Para uma primeira experiência, as configurações padrões são adequadas.

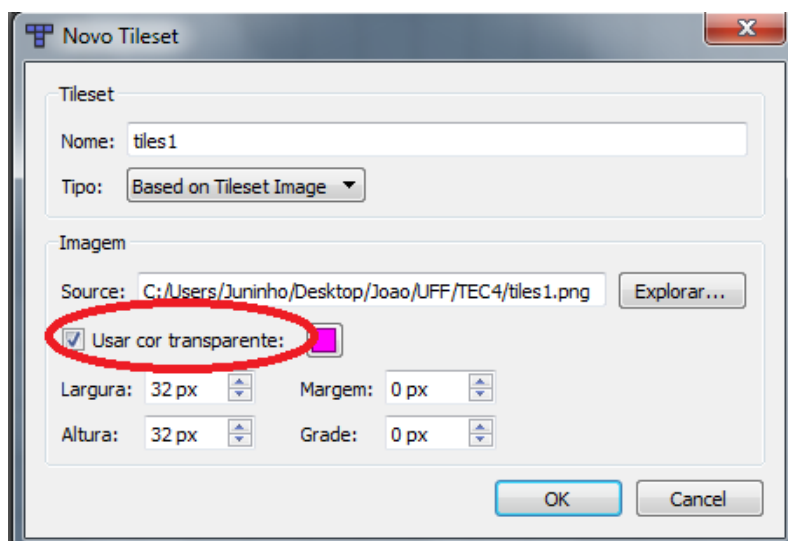


5- Importando um TileSet

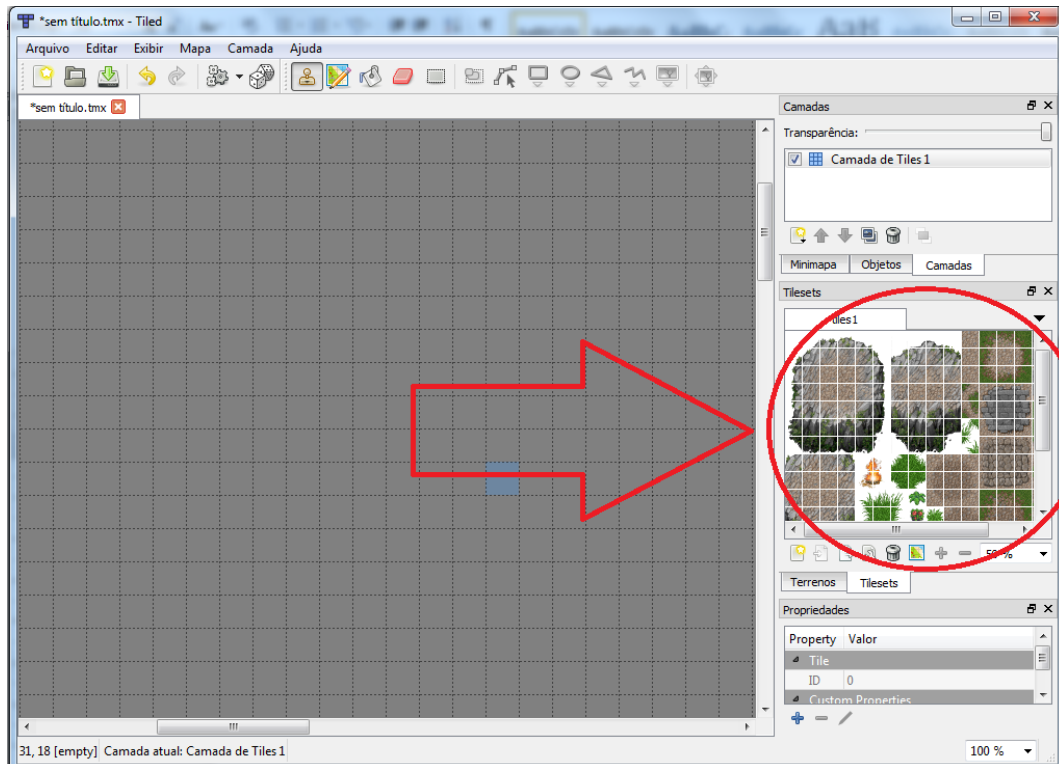
Um mapa do tipo TiledMap é um mapa dividido em NxM divisões. No caso de um mapa ortogonal, são NxM quadrados. Cada Tile pode ter propriedades próprias, assim podemos representar melhor objetos do mundo como fogo, árvores, rochas, etc. Um TileSet é um arquivo que possui um conjunto de modelos de Tiles. O TileSet usado nesse tutorial pode ser baixado por esse link(<http://www.atmanx.com/UFF/cursosDX/LostGarden.rar>). No programa Tiled, no painel de Tilesets, clique no ícone de Novo Tileset.



Selecione o seu arquivo de TileSet, eu recomendo que marque a opção **usar cor transparente**.

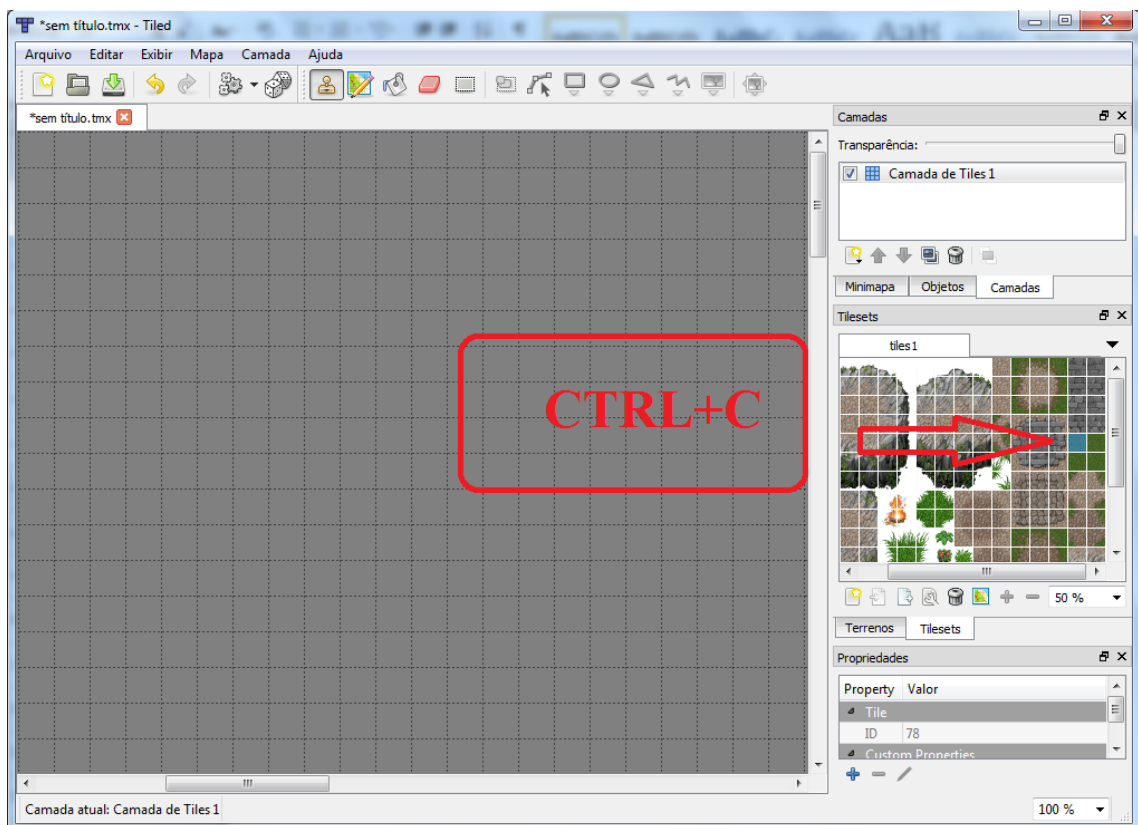


Pronto, seu TileSet foi importado e aparece aqui.

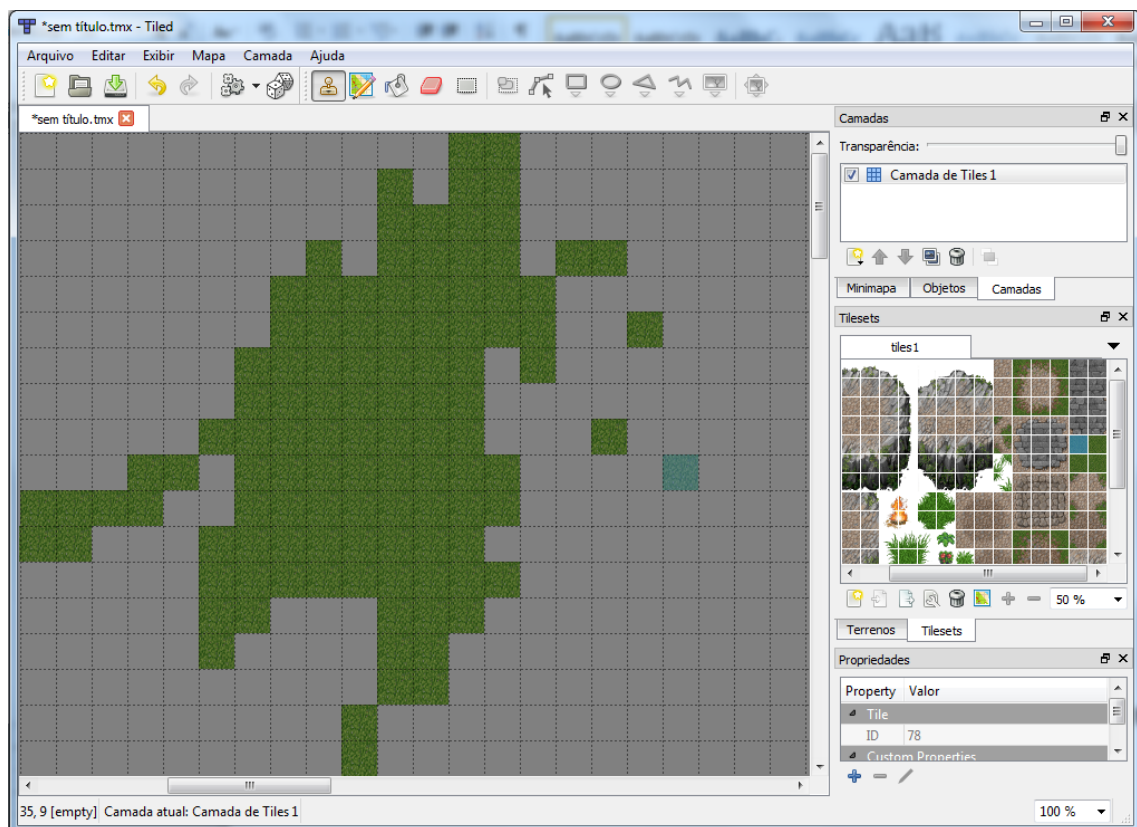


6- Criando um mapa simples.

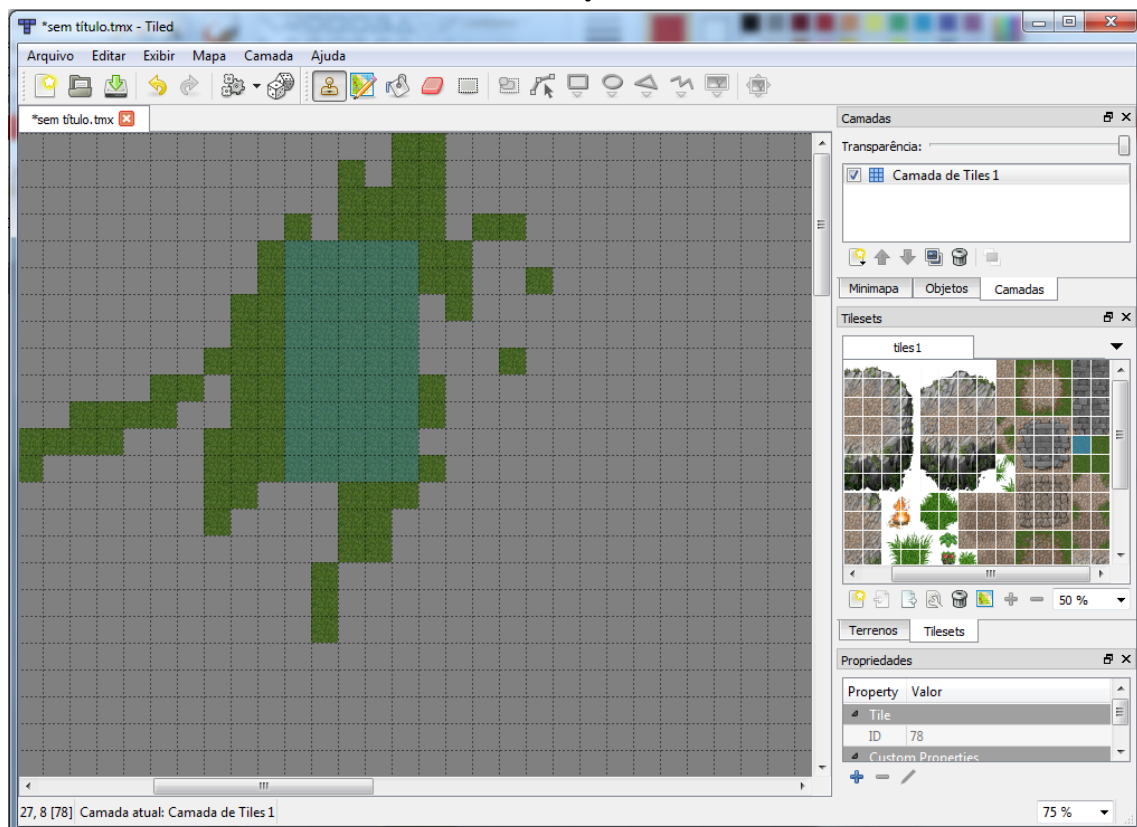
Clique em um Tile de seu TileSet, e aperte ctrl+C



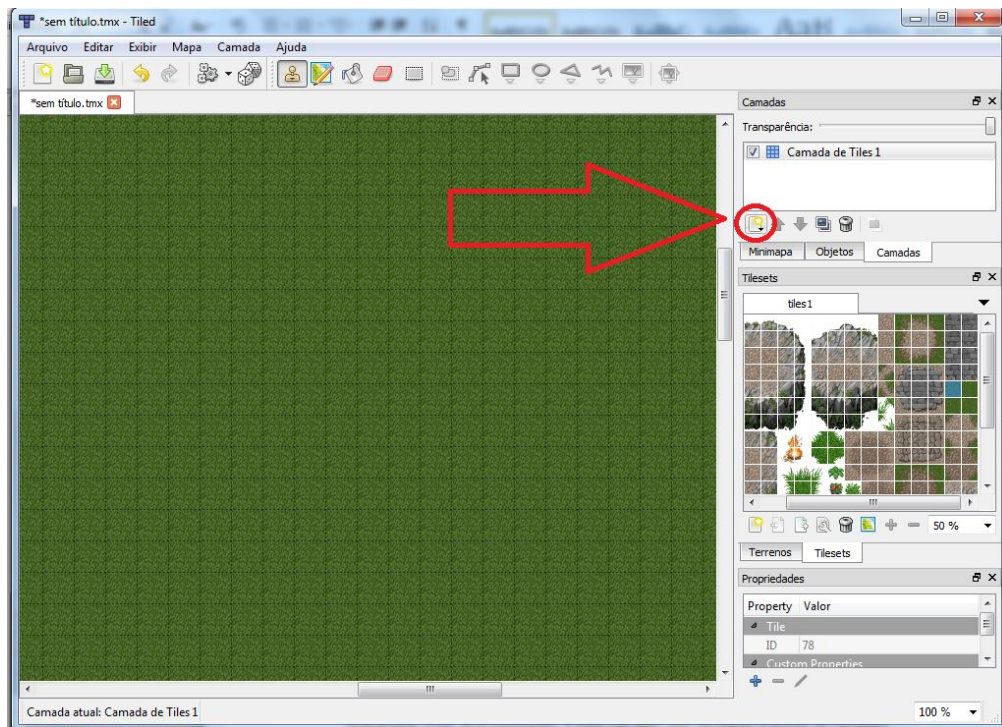
Em seguida clique no mapa, e assim preencha-o com o tile selecionado.



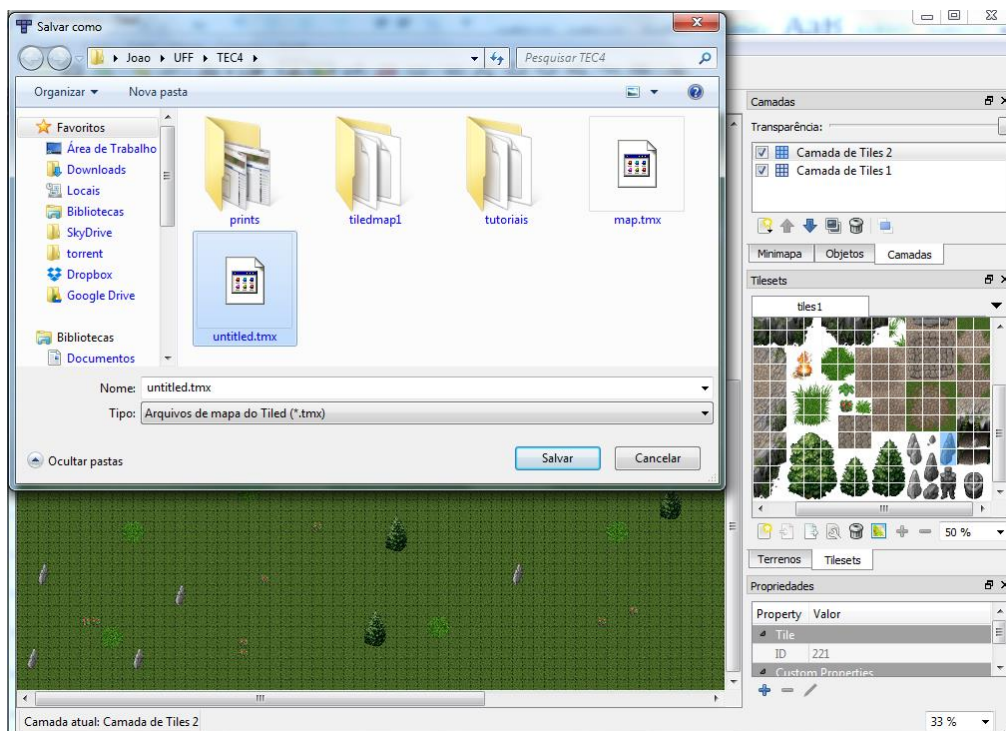
Você pode selecionar vários Tiles do Mapa, para copia-los. Basta manter pressionado o botão direito do mouse e arrastar a área de seleção.



Se você desenhar algo no mapa, como o terreno, e tentar desenhar algo em cima, como uma árvore, o tile será substituído, e você não quer isso. No painel a direita de camadas, crie uma nova camada de Tiles.

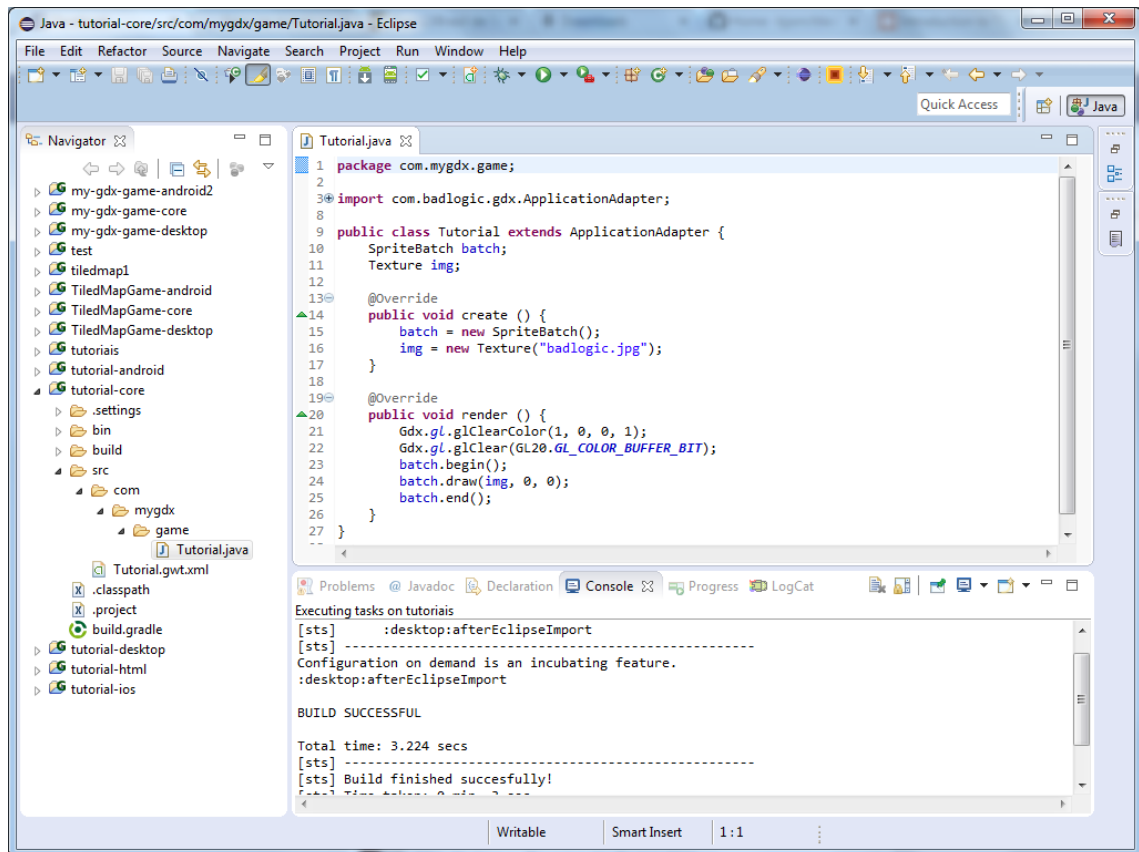


Então, já pode adicionar árvores, flores, etc. Lembre-se de que sempre que precisar que tiles se sobreponham, você pode precisar adicionar novas camadas. Aqui está um exemplo simples de um mapa criado dessa maneira, para avançarmos para o próximo passo. Apenas termine seu mapa e salve o arquivo .tmx

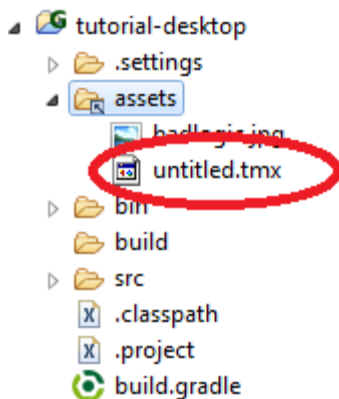


7- Importando seu mapa para um projeto do LibGDX no Eclipse.

Atenção: Para o próximo passo é necessário que você saiba como criar um novo projeto LibGDX e importa-lo no Eclipse. Caso precise de ajuda, há um tutorial detalhado nesse link: (<http://www.atmanx.com/UFF/cursoDX/Introducao%20ao%20Desenvolvimento%20de%20Jogos%20com%20LibGDX.pdf>)



Copie o Arquivo de mapa para a pasta Assets do projeto. Pode ser para qualquer versão do projeto, seja Desktop, Android, IOS ou Html.



Na classe principal da versão Core do projeto, faça a classe estender a classe Game, apague todo o resto do código gerado pelo LibGDX.

```
import com.badlogic.gdx.ApplicationAdapter;

public class Tutorial extends Game {

    @Override
    public void create () {

    }

    @Override
    public void render () {

    }

}
```

Agora para o método Render, chame `super.render()`, pois já faz boa parte do que precisaremos. Para o método Create iremos construir um objeto de uma nova classe, que irá gerenciar a construção do nosso Mapa. Chamarei essa classe de Play.

```
package com.mygdx.game;

import com.badlogic.gdx.ApplicationAdapter;

public class Tutorial extends Game {

    @Override
    public void create () {
        this.setScreen(new Play());
    }

    @Override
    public void render () {
        super.render();
    }

}

public class Play implements Screen {

    @Override
    public void show() {

    }

    @Override
    public void render(float delta) {

    }

    @Override
    public void resize(int width, int height) {

    }

    @Override
    public void pause() {

    }

    @Override
    public void resume() {

    }

    @Override
    public void hide() {

    }

    @Override
    public void dispose() {

    }

}
```

Feito isso teremos nossa classe play, já podemos trazer o Mapa pra ela. Vamos criar uma variável para o mapa, e vamos carregar o arquivo que criamos dentro do método Show().

```
public class Play implements Screen {

    private TiledMap map;
    private OrthogonalTiledMapRenderer renderer;

    @Override
    public void show() {
        map = new TmxMapLoader().load("untitled.tmx");
        renderer= new OrthogonalTiledMapRenderer(map);
    }

}
```

Pronto, nosso mapa já será renderizado, porém temos que fazer mais ajustes nessa classe. Como a maior parte desses ajustes não são referentes ao TiledMap, e sim a programação para LibGdx, não irei dar uma explicação técnica para cada linha de código.

8- Criação de uma câmera:

Vamos criar uma câmera para nosso jogo

```
public class Play implements Screen {  
  
    private TiledMap map;  
    private OrthogonalTiledMapRenderer renderer;  
    private OrthographicCamera camera=new OrthographicCamera();
```

E atualiza-la no método **resize**.

```
@Override  
public void resize(int width, int height) {  
    camera.viewportWidth=width;  
    camera.viewportHeight=height;  
    camera.update();  
}
```

Por fim escrevemos o método render.

```
@Override  
public void render(float delta) {  
    Gdx.gl.glClearColor(0, 0, 0, 1);  
    Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);  
    renderer.setView(camera);  
    renderer.render();  
}
```

Se você fez tudo certo, seu código estará assim:

```
public class Play implements Screen {

    private TiledMap map;
    private OrthogonalTiledMapRenderer renderer;
    private OrthographicCamera camera=new OrthographicCamera();

    @Override
    public void show() {
        map = new TmxMapLoader().load("untitled.tmx");
        renderer= new OrthogonalTiledMapRenderer(map);
    }

    @Override
    public void render(float delta) {
        Gdx.gl.glClearColor(0, 0, 0, 1);
        Gdx.gl.glClear(GL20.GL_COLOR_BUFFER_BIT);
        renderer.setView(camera);
        renderer.render();
    }

    @Override
    public void resize(int width, int height) {
        camera.viewportWidth=width;
        camera.viewportHeight=height;
        camera.update();
    }

    @Override
    public void pause() {
    }

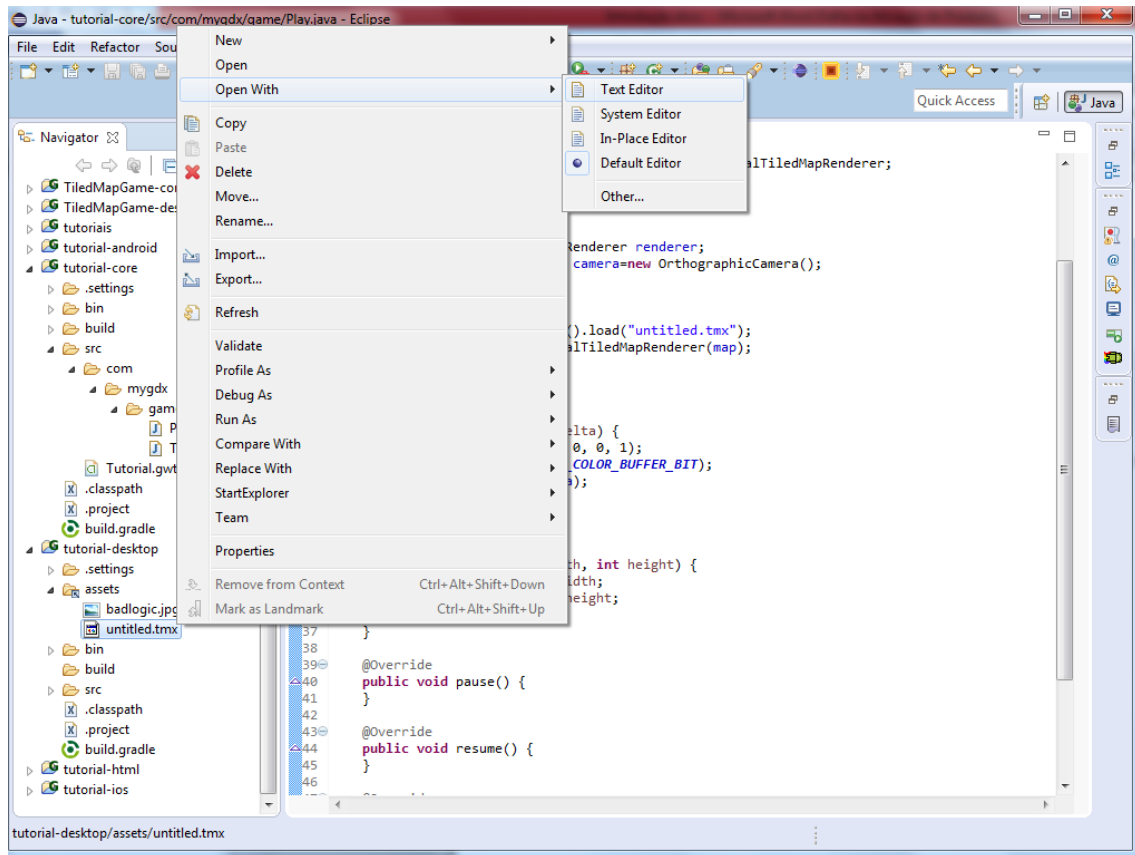
    @Override
    public void resume() {
    }

    @Override
    public void hide() {
    }

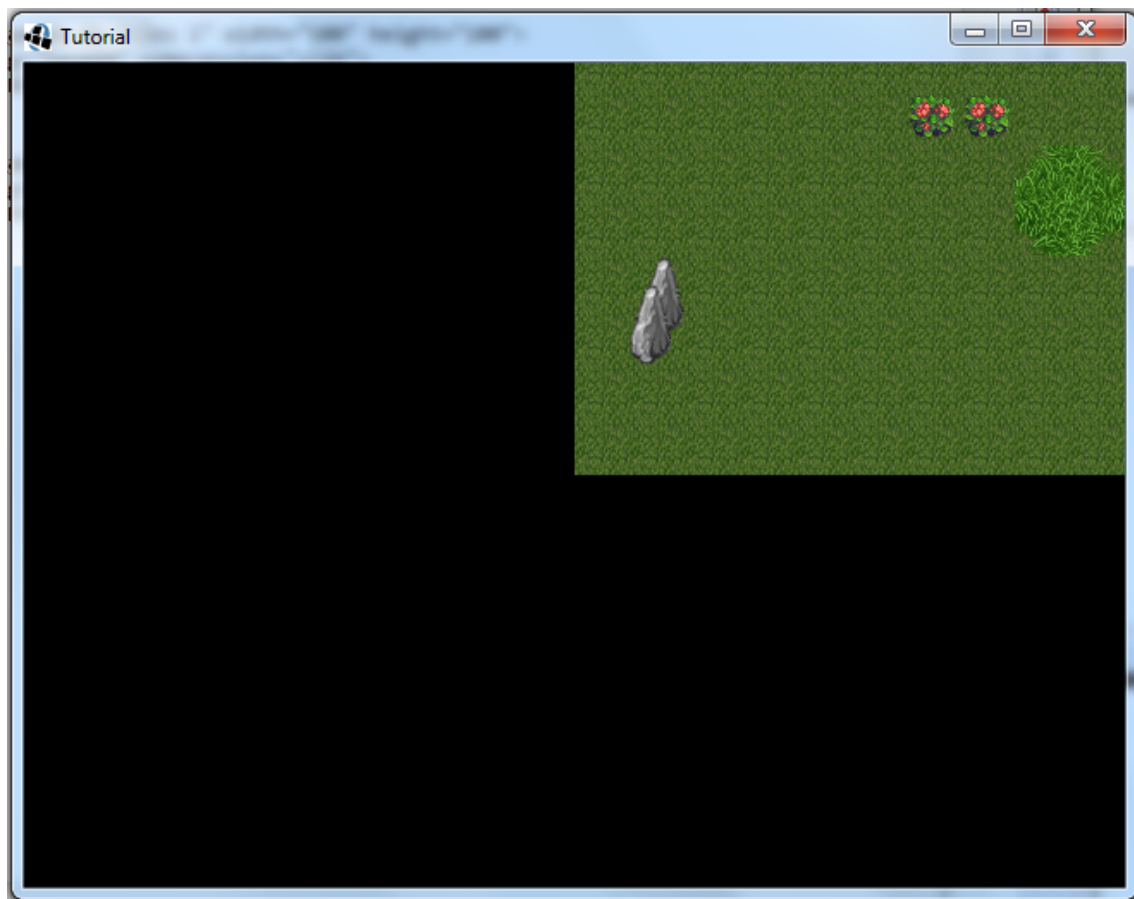
    @Override
    public void dispose() {
    }
}
```

9- Edição das propriedades do arquivo tmx.

Ainda é necessário fazer uma ultima edição nas propriedades do arquivo tmx. Pelo Eclipse, edite o arquivo com um editor de texto.



As informações do arquivo TMX estarão escritas em texto. Nessa parte destacada na imagem abaixo está o caminho para o Tileset utilizado. Tenha certeza de que o caminho está relativo a pasta do programa, sem nada do tipo ~~C:/Users/Fulano/....~~. Na verdade, nenhuma pasta que não seja nomes de packages do seu projeto deve aparecer no caminho. O exemplo abaixo está correto. Obs: Caso você use múltiplos TileSets, deverá observar o caminho de todos.



A câmera está descentralizada, basta editar as coordenadas iniciais dela, mas o mapa já está renderizado, isso que importa. Até mais!