#ifndef PIECES\_H

#define PIECES\_H

#pragma once

#include <iostream>

#include <stdio.h>

#include <string>

class Piece {

public:

Piece(const string& type) : type(type) {} ///on construit une pièce uniquement à partir de son type en supposant qu'elle n'est pas sur le plateaudès qu'elle est créée

virtual ~Piece() {}

int getX() const { return x; }

int getY() const { return y; }

string getType() const { return type; }

bool estPlacee() const { return placee; }

void placerPiece(int x, int y) {

if (!placee) {

this->x = x;

this->y = y;

placee = true;

}

}

void deplacerPiece(int x, int y) {

if (placee) {

this->x = x;

this->y = y;

}

}

protected: ///pour donner accès à ces données dans les classes qui héritent de Piece

int x, y; ///coordonnées pour l'attribut position

string type; ///type de la pièce

bool placee = false; ///pour savoir si la pièce créée est en jeu ou non

};

class ReineAbeille : public Piece {

// parametrer les déplacements

};

class ReineAbeille : public Piece{

public:

void deplacerReine(int x, int y){

if((x == ReineAbeille.x || x == ReineAbeille.x + 1 || x == ReineAbeille.x - 1) && // Je vérifie que le Movement demandé est possible selon les règles

y == ReineAbeille.y || y == ReineAbeille.y + 1 || x == ReineAbeille.y - 1) &&

!estOcupe(x,y)){ // Il faut aussi rajouter une condition si il y a des cases occupées ou non

ReineAbeille.deplacerPiece(x,y); //La fonction deplacerPiecer vérifira si il y a une pièce adjacente

}

}

bool reineEntourer(){

return if(estOcupe(ReineAbeille.x+1,ReineAbeille.y) && //Vérifie si toute les cases autour de la reine sont occupé

estOcupe(ReineAbeille.x-1,ReineAbeille.y) &&

estOcupe(ReineAbeille.x,ReineAbeille.y+1) &&

estOcupe(ReineAbeille.x,ReineAbeille.y-1) &&

estOcupe(ReineAbeille.x+1,ReineAbeille.y+1) &&

estOcupe(ReineAbeille.x+1,ReineAbeille.y-1))

}

}

class Case{

int m\_x, m\_y;

bool m\_estOcupe;

public:

bool getEtat(){

return m\_estOcupe

}

}

class Araignee : public Piece {

// parametrer les déplacements

};

class Scarabee : public Piece {

// parametrer les déplacements

};

class Fourmis : public Piece {

// parametrer les déplacements

};

class Sauterelle : public Piece {

// parametrer les déplacements

};

class Moustique : public Piece {

// parametrer les déplacements

};

#endif