

Tetris : nvs__2__58124__60131

nvs

20 juin 2024

Tetris : nvs__2__58124__60131 (58124 & 60131)

dépôt

`https://git.esi-bru.be/58124/dev4_tetris_d121_b-a`

`git@git.esi-bru.be:58124/dev4_tetris_d121_b-a.git`

modélisation

retours donnés lors de la défense orale par les étudiants.

métier / console

remise

.gitignore et taille du dépôt

- il y a un fichier `.gitignore`, mais pas celui *Uploaded 6/07/23, 16 :33* dans la formation dev4 de poESI : ko
- 2,43 Mo (2.553.732 octets) :
 - `.git` : ok
 - `html, latex` : ko

tag / commit ok

retard (void)

autre (void)

documentation

- ko :
 - pas de fichier pour **doxygen**
 - fichiers avec énumérations non documentés
- ok : toutes les classes, énumérations, fonctions documentées

fichier de configuration uniquement

- ko : dossiers **html**, **latex** dans le dépôt

rapport

- ko : le contenu du rapport n'est pas ce qui est attendu : il s'agit uniquement des écarts entre implémentation et analyse

format pdf ok

bogue non signalé (void)

écart / ajout non signalé (void)

autre (void)

rapport / code

avertissement restant

gcc signalé

(void)

non signalé

```
[2/126 1.7/sec] Building CXX object src/CMakeFiles/src.dir/Model/Board.cpp.obj
C:/[...]/nvs_2_58124_60131/rnvs_console/src/Model/Board.cpp: In member function 'void model::Board::set_shape_on_board(const model::shape)':
C:/[...]/nvs_2_58124_60131/rnvs_console/src/Model/Board.cpp:31:27: warning: comparison of integer expressions of different signedness: 'int'
31 |         for (int i = 0; i < positions.size(); ++i) {
    |                         ~^~~~~~
```

à régler éventuellement

```
[15/126 5.4/sec] Building CXX object src/CMakeFiles/src.dir/Model/Game.cpp.obj
C:/[...]/nvs_2_58124_60131/rnvs_console/src/Model/Game.cpp: In member function 'void model::Game::generateRandomBoard()':
C:/[...]/nvs_2_58124_60131/rnvs_console/src/Model/Game.cpp:96:22: warning: comparison of integer expressions of different signedness: 'int'
96 |     for(int i = 0; i < maximumShapes; i++){
    |                    ~^~~~~~
```

à régler éventuellement

cppcheck signalé

(void)

non signalé

src\Model\Shapes\Position.h: 26:5 [src\Model\Shapes\Position.h:26]

style : noExplicitConstructor

```
    Position(int x = 0, int y = 0);
    ^
```

Class 'Position' has a constructor with 1 argument that is not explicit.

à régler éventuellement

src\View\View.h: 47:5 [src\View\View.h:47]

style : noExplicitConstructor

```
    View(const model::TetrisInterface& game);
    ^
```

Class 'View' has a constructor with 1 argument that is not explicit.

à régler éventuellement

src\Model\TetrisInterface.h: 104:18 [src\Util\Observable.h:33] -> [src\Model\TetrisInterface.h:104]

```
style : missingOverride
    virtual void notify() = 0;
    ^
```

The function 'notify' overrides a function in a base class but is not marked with a 'override' specifier.

à régler

```
src\Controller\KeyBoardInput.cpp: 9:42 [src\Controller\KeyBoardInput.cpp:9]
performance : passedByValue
int inputNumber(int min, int max, string message) {
    ^
```

Function parameter 'message' should be passed by const reference.

à régler

```
src\Controller\KeyBoardInput.cpp: 28:39 [src\Controller\KeyBoardInput.cpp:28]
performance : passedByValue
std::string choiceDefaultBoard(string msg) {
    ^
```

Function parameter 'msg' should be passed by const reference.

à régler

```
src\Model\Level.h: 27:5 [src\Model\Level.h:27]
style : noExplicitConstructor
    Level(int level = 1);
    ^
```

Class 'Level' has a constructor with 1 argument that is not explicit.

à régler éventuellement

```
src\Model\Game.h: 62:5 [src\Model\Game.h:62]
style : noExplicitConstructor
    Game(int lvl, int row = 20, int col = 10);
    ^
```

Class 'Game' has a constructor with 1 argument that is not explicit.

à régler éventuellement

```
src\Model\Board.h: 58:10 [src\Model\Board.cpp:21] -> [src\Model\Board.h:58]
```

```
style (inconclusive) : functionConst
    bool is_cell_empty(shape::Position& pos);
    ^
```

Technically the member function 'model::Board::is_cell_empty' can be const.

à régler

```
src\Model\Board.h: 65:10 [src\Model\Board.cpp:25] -> [src\Model\Board.h:65]
style (inconclusive) : functionConst
    bool is_out_of_board(shape::Position& pos);
    ^
```

Technically the member function 'model::Board::is_out_of_board' can be const.

à régler

```
src\Model\Board.h: 106:10 [src\Model\Board.cpp:80] -> [src\Model\Board.h:106]
style (inconclusive) : functionConst
    bool is_row_full(int row);
    ^
```

Technically the member function 'model::Board::is_row_full' can be const.

à régler

```
src\Model\Board.cpp: 70:31 [src\Model\Board.h:98] -> [src\Model\Board.cpp:70]
style (inconclusive) : funcArgNamesDifferent
    void Board::move_line(int line) {
        ^
```

Function 'move_line' argument 1 names different: declaration 'row' definition 'line'.

à régler éventuellement

```
src\Model\TetrisInterface.h: 73:18 [src\Util\Observable.h:18] -> [src\Model\TetrisInterface.h:73]
style : missingOverride
    virtual void Register(util::Observer* obs) = 0;
    ^
```

The function 'Register' overrides a function in a base class but is not marked with a 'override' specifier.

à régler

```
src\Model\TetrisInterface.h: 79:18 [src\Util\Observable.h:25] -> [src\Model\TetrisInterface.h:79]
```

```
style : missingOverride
    virtual void Unregister(util::Observer* obs) = 0;
    ^
```

The function 'Unregister' overrides a function in a base class but is not marked with a 'override' specifier.

à régler

```
src\Model\Game.cpp: 71:20 [src\Model\Game.cpp:71]
style : constVariableReference
    for(Position & pos : positions) {
    ^
```

Variable 'pos' can be declared as reference to const

à régler

```
src\Model\Level.cpp: 37:54 [src\Model\Level.cpp:36] -> [src\Model\Level.cpp:37]
style : duplicateExpression
    _timetoLevel[1] = chrono::duration<double>(number/60);
    ^
```

Same expression on both sides of '/'.
^

sans avis

```
src\Model\Shapes\Shape.cpp: 5:8 [src\Model\Shapes\Shape.cpp:5]
warning : uninitMemberVar
Shape::Shape() : rotation_state{0} { offset = Position(0, 0); }
    ^
```

Member variable 'Shape::id' is not initialized in the constructor.

à régler

```
src\Model\Shapes\Shape.cpp: 5:38 [src\Model\Shapes\Shape.cpp:5]
performance : useInitializationList
Shape::Shape() : rotation_state{0} { offset = Position(0, 0); }
    ^
```

Variable 'offset' is assigned in constructor body. Consider performing initialization in initialization list.

à régler

```
src\Model\Shapes\Shape.cpp: 33:73 [src\Model\Shapes\Shape.cpp:33]
```

```
performance : passedByValue
void Shape::set_cell_status(std::map<int, std::vector<shape::Position>> cell_status) {
    ~
```

Function parameter 'cell_status' should be passed by const reference.

à régler

```
src\Model\Shapes\Shape.cpp: 45:5 [src\Model\Shapes\Shape.cpp:45]
style : useStlAlgorithm
    for(const Position& block_pos : cell_status) {
    ~
```

Consider using std::any_of algorithm instead of a raw loop.

à régler éventuellement

```
src\View\View.h: 25:10 [src\View\view.cpp:34] -> [src\View\View.h:25]
performance (inconclusive) : functionStatic
    void print_next_shape(const model::shape::Shape& shape);
    ~
```

Technically the member function 'view::View::print_next_shape' can be static (but you may consider moving to unnamed namespace).

à régler éventuellement

```
src\View\View.h: 32:10 [src\View\view.cpp:50] -> [src\View\View.h:32]
performance (inconclusive) : functionStatic
    void print_board(const std::vector<std::vector<int>>& board, const model::shape::Shape& curr_shape);
    ~
```

Technically the member function 'view::View::print_board' can be static (but you may consider moving to unnamed namespace).

à régler éventuellement

```
src\View\View.h: 39:10 [src\View\view.cpp:30] -> [src\View\View.h:39]
performance (inconclusive) : functionStatic
    void print_stat(const int& score, const int& level);
    ~
```

Technically the member function 'view::View::print_stat' can be static (but you may consider moving to unnamed namespace).

à régler éventuellement

```
src\View\View.h: 57:10 [src\View\view.cpp:19] -> [src\View\View.h:57]
```

```
performance (inconclusive) : functionStatic
    void gameIsFinish();
    ~
```

Technically the member function 'view::View::gameIsFinish' can be static (but you may consider moving to unnamed namespace).

à régler éventuellement

```
src\Model\Game.cpp: 36:0 [src\Model\Game.cpp:36]
style : unusedFunction
const double Game::get_tick() const {
    ~
```

The function 'get_tick' is never used.

sans avis

```
src\Model\Game.cpp: 187:0 [src\Model\Game.cpp:187]
style : unusedFunction
void Game::Unregister(Observer* obs) {
    ~
```

The function 'Unregister' is never used.

sans avis

code source

portabilité

casse noms fichiers ok

séparateur / ok

c++ standard ok

si pas std : portabilité (void)

bonnes pratiques

déclarations anticipées si possible ko :

- dans Position.h, #include "../Direction.h" inutile
- dans Shapes.h, #include "Shape.h" inutile
- dans Board.h, #include "../Shapes/Position.h" inutile
- dans Board.h, #include "../Shapes/Shape.h" inutile
- dans Bag.h, #include "Shapes/Shape.h" inutile
- dans TetrisInterface.h, #include "../Shapes/Shape.h" inutile
- dans Observable.h, #include "Observer.h" inutile
- dans View.h, #include "../Model/TetrisInterface.h" inutile
- dans View.h, #include "../Model/Shapes/Shape.h" inutile
- dans Controller.h, #include "../Model/TetrisInterface.h" inutile
- dans Controller.h, #include "../View/View.h" inutile

include inutile ou dans .h plutôt que .cpp ko :

- dans TetrisInterface.h, #include "../Direction.h" inutile
- dans TetrisInterface.h, #include "../Util/Observer.h" inutile

absence de using namespace dans .h ok

autre (void)

gestion de la mémoire ok

tests unitaires (ceci n'est pas demandé)

- tests mis en oeuvre avec catch2

classes métier

absence d'interaction avec monde extérieur (cout, fstream) ok

plateau taille du plateau

- défaut : ok
- configurable : ok

plateau prérempli

- ko : option possible, mais il existe des pièces flottantes au dessus du vide

mouvements des pièces translations \leftarrow g, \rightarrow d, \downarrow b

ok

drop

ok

rotations

ok

pièces 7 types de pièces

ok

paquet + recreation du paquet mélangé quand vidé

ok

apparition de la pièce courante

- ko : pas au milieu si taille customisée

score choix du niveau initial

ok

changement de niveau toutes les 10 lignes

ok

effacement des lignes

ok

calcul du score

— ok, drop compris

fin de partie quand joueur perd

ok

quand score atteint

— ko : pas implémenté

quand nombre de lignes effectuées

— ko : pas implémenté

quand temps écoulé

— ko : pas implémenté

détection effective de la fin

— ko : détectée par le contrôleur

méthodes

complètes : 1 méthode / 1 action de jeu ko :

— la détection de fin n'est pas automatique, c'est le contrôleur qui s'en occupe dans `void Controller::start()`

impossibilité de tricher / mettre le jeu dans un état non valide (bibliothèque) ko :

— `TetrisInterface::setShapeOnBoard(int drop_value = 0)` est public : possibilité de modifier le plateau sans vérification des règles

— `TetrisInterface::set_score(int score)` est public : possibilité de modifier le plateau sans vérification des règles

— `TetrisInterface::next_level()` est public : possibilité de modifier le plateau sans vérification des règles

— `TetrisInterface::next_shape()` est public : possibilité de modifier le plateau sans vérification des règles

pour régler ces problèmes : déclarer ces méthodes **private**

mvc & observer

- ko :
 - les méthodes d'`Observable` sont virtuelles pures : pq ?
 - comme la détection de fin n'est pas réalisée automatiquement par les méthodes métier, la vue ne s'occupe pas de l'affichage de fin de partie dans la méthode `update()`, mais c'est le contrôleur qui déclenche `void View::gameIsFinish()`

autre (void)

exécutable

fiabilisation lectures clavier ok

tour par tour

- ok, sauf si mouvement vers le bas : dans ce cas, pas de chute

convivialité ok

affichages

- affichage du plateau : ok
- affichage du score : ok
- affichage du niveau : ok
- affichage du nombre de lignes effacées : (void)
- affichage de la brique suivante : ok

autre (void)

gui

remise

.gitignore et taille du dépôt

- `.gitignore` : ok
- taille : 72,8 Mo (76.345.029 octets) :
 - `.git` : 61,3 Mo (64.331.168 octets)

- commits : 130

tag / commit

- ko : doxyfile ajouté après tag gui

retard (void)

autre (void)

documentation

- ok : classe graphique bien documentée
- ko :
 - chemin absolu pour Source : C:/Users/Berin/OneDrive/Bureau/dev4_tetris_d121_b-a/DEV4_TETRIS_D121_58124_60131
 - chemin absolu pour Destination : C:/Users/Berin/OneDrive/Bureau/dev4_tetris_d121_b-a

fichier de configuration uniquement ok

rapport

format pdf ok

bogue non signalé (void)

écart / ajout non signalé (void)

autre (void)

code source

gestion de la mémoire ko :

- new QTimer, sans parent ni delete
- new QMovie, sans parent ni delete

métier : chute de la pièce active

(pseudo-)temps réel

- ok : avec `QTimer`
- ko : déplacement vers le bas intempestif : vestige de l'application console où chaque mouvement est suivi d'un déplacement vers le bas : ici il faut retirer ce déplacement

accélération

- ok : intervalles comme dans table 1 de l'énoncé

contrôleur

menu de configuration ok

respect des règles ok

convivialité

- ok : contrôle au clavier bien réactif

vue

design pattern observer

- presque ok : la fin de partie n'est pas gérée par la partie observateur `void MainWindow::update()` mais par le contrôleur et `void MainWindow::auto_increment()` qui est invoqué à chaque *tic* du `QTimer`

convivialité

- ok :
 - il y a un texte d'aide, un *gif* animé
 - affichage du niveau, du score et de la brique suivante, pas du nombre de lignes effacées

autre (void)

examen

retours donnés aux étudiants lors de l'examen oral.