



## HLIN302 – Projet noté « 1010 le casse brique »

**Programmation impérative avancée**  
**Pascal GIORGI et Laurent IMBERT**

---

L'objectif de ce projet est la réalisation d'un logiciel en C++ permettant de jouer à un jeu de casse brique dans un terminal Unix. Le jeu choisi s'appelle 1010 ou ten ten, et voici une illustration graphique du jeu :



Le principe du jeu est très simple. Le plateau du jeu est constitué d'une grille  $10 \times 10$  où l'on peut déposer des pièces colorées ayant des formes prédéfinies (à la tetris). Les formes sont composées de plusieurs carrés connexes et l'objectif est de poser un maximum de pièces sur le plateau. Dès lors qu'une ligne ou une colonne du plateau est remplie par des carrés (sans se soucier de la couleur), ils sont supprimés et l'on gagne des points. Le jeu se termine dès qu'on ne peut plus poser les pièces proposées en bas de l'écran. Ces pièces sont données par paquet de trois lorsque les précédentes ont toutes été posées.

### Règles du jeu

L'objectif du jeu est donc de poser un maximum de pièces afin de récolter le plus de point. Il est bien évident que l'on pose des pièces uniquement sur les espaces libres du plateau de jeu, le chevauchement de pièces est

interdit. Le comptage des points se fait en fonction du nombre de lignes ou de colonnes que l'on supprime lorsque l'on pose une pièce. Le comptage du jeu initial étant complexe, nous allons proposer notre propre règle de calcul. Le nombre de point gagnés est fonction du nombre de lignes ou de colonnes que l'on détruit.

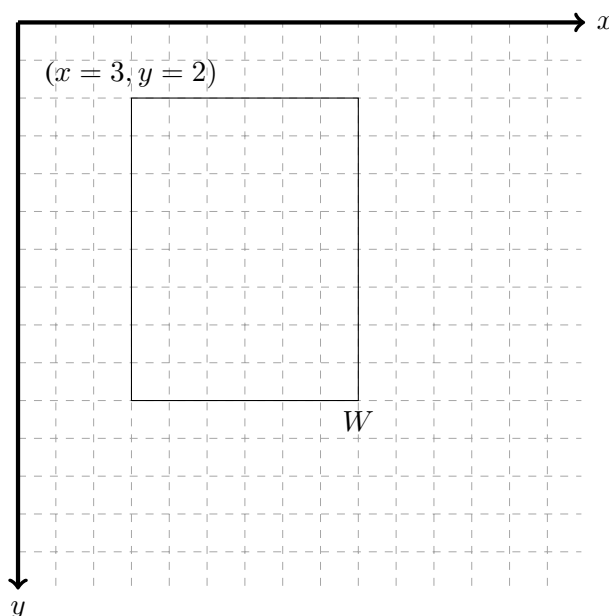
nbr lignes ou colonnes	1	2	3	4	5	6
nbr points	10	25	45	60	75	90

Par exemple, la destruction de 2 lignes rapporte 25 points. Si l'on arrive à détruire en même temps des lignes et des colonnes, on additionne l'ensemble des points et on multiplie le résultat par deux. Exemple : la destruction de 2 lignes et 1 colonne en même temps rapporte  $2 \times (25 + 10) = 70$  points. Le but du jeu est de battre le meilleur score qui est affiché en haut de l'écran à droite.

## Interface graphique en mode texte

Un des objectifs de ce projet est de fournir une version graphique de ce jeu en s'appuyant sur le mode texte de votre terminal Unix. Pour cela, vous devrez vous appuyer sur une bibliothèque permettant la manipulation de votre terminal. Cette bibliothèque se nomme `ncurses`<sup>1</sup> et elle est fournie librement dans toutes les distributions Unix. Pour utiliser cette bibliothèque, il vous suffit d'inclure le fichier `ncurses.h` et compiler votre programme en ajoutant l'option `-lncurses`.

Comme la prise en main de cette bibliothèque n'est pas évidente, nous vous fournissons une classe C++ permettant une gestion simplifiée. La classe `Window` définit des objet permettant de gérer des fenêtres dans votre terminal. Par exemple, si votre terminal est de dimension  $15 \times 15$ , la déclaration `Window W(3, 2, 6, 8)` permet de définir une fenêtre de taille  $6 \times 8$  dont le coin supérieur gauche est positionné à la coordonnée  $(x = 3, y = 2)$ . Attention, la coordonnée du coin supérieur gauche de votre terminal est à la position  $(x = 0, y = 0)$  et l'axe des  $x$  est orienté vers la droite alors que l'axe des  $y$  est orienté vers le bas. La figure ci-dessous illustre la construction de notre exemple.



À partir d'un objet `Window` il est possible d'écrire du texte ou un caractère à une position donnée dans la fenêtre. Pour cela, il faut utiliser la méthodes `print(x, y, s)` où  $(x, y)$  représente la position où l'on souhaite écrire dans la fenêtre (la position  $(0, 0)$  correspond au coin supérieur gauche de la fenêtre) et  $s$  représente le caractère ou la chaîne de caractère que l'on veut afficher.

La fenêtre utilise un système d'affichage de couleur : une couleur de texte et une couleur de fond. Il est possible de changer cette couleur via la méthode `setCouleurFenetre`. De même, on peut changer la couleur de la

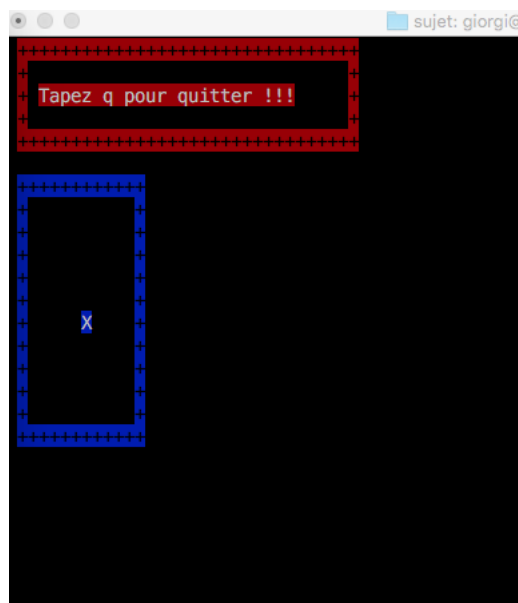
1. <https://fr.wikipedia.org/wiki/Ncurses>

bordure de la fenêtre via la méthode `setCouleurBordure`. On peut également spécifier une couleur `c` à la méthode `print(x, y, s, c)` pour écrire dans une couleur choisie. Enfin, on peut supprimer l'ensemble des textes écrits dans une fenêtre en utilisant la méthode `clear`. Le code de la classe `Window` est disponible dans l'espace projet sur l'ENT / Moodle du cours : fichiers `window.h` et `window.cpp`. Le fichier header comporte des commentaires vous permettant d'utiliser correctement cette classe et les couleurs d'affichage. Si vous le souhaitez, vous pouvez améliorer cette classe pour qu'elle gère plus d'options d'affichage de la bibliothèque `ncurses`.

**ATTENTION** : pour que votre programme fonctionne correctement avec la classe `Window` il faut que votre programme ressemble à celui-ci :

```
1 #include "window.h"
2
3 int main() {
4     startProgramX();
5     myprogram();
6     stopProgramX();
7     return 0;
8 }
```

Vous devrez remplacer l'appel à la fonction `myprogram()` par la fonction principale de votre jeu. Vous trouverez un exemple de programme utilisant la classe `Window` dans l'espace projet sur l'ENT / Moodle du cours. Rappel, pour compiler il vous faudra ajouter l'option `-lncurses`. Ce programme vous montrera également comment vous pouvez interagir avec les actions au clavier. Voici un rendu graphique du programme :



## Cahier des charges

Vous devez écrire le code nécessaire pour jouer au jeu 1010 dans un terminal Unix tel que décrit précédemment. En particulier, vous devez vous appuyer sur les concepts vus en cours, TD, TP, et particulièrement celui des classes. Il vous faudra donc fournir plusieurs classes permettant de modéliser le jeu : pièces, plateau de jeu, menu, score, ... Pour l'affichage, vous devez obligatoirement utiliser la classe `Window` ou sinon directement la bibliothèque `ncurses`.

## Les graphismes du jeu

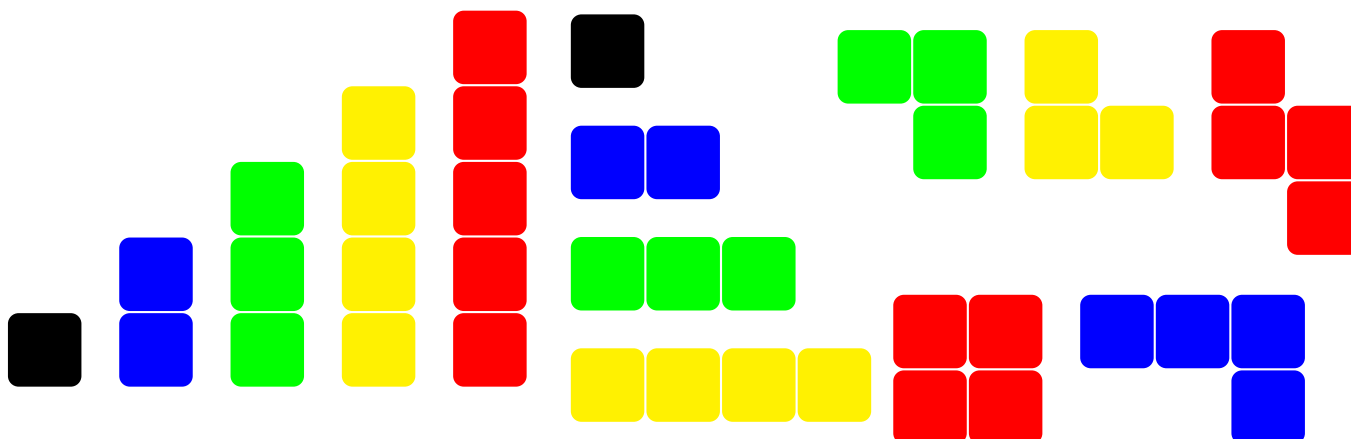
Le jeu doit comporter au moins 3 zones d'affichage correspondant à la grille du jeu, l'espace d'apparition des trois pièces à placer et l'espace score/meilleur score.

## Le plateau de jeu

Il sera constitué par une grille  $10 \times 10$  dans laquelle on pourra placer des pièces de jeu.

## Les pièces de jeu

Les pièces du jeu comportent au maximum 5 carrés connexes. Les pièces qu'il faudra placer seront choisies aléatoirement parmi l'ensemble des pièces suivantes :



Dans votre jeu, les carrés pourront être représentés par un caractère (e.g. 'X'), un ensemble de caractères ou tout autre forme graphique intelligible. Les pièces comportent un carré "pivot" permettant de positionner la pièce : lorsqu'on demande de positionner la pièce aux coordonnées (x,y), on placera le carré pivot de la pièce à cette position.

## Travail à réaliser

Le travail à réaliser pourra se faire en plusieurs étapes. Plus les étapes seront réalisées meilleure sera la note finale du projet.

(étape 1) Le jeu doit être jouable avec les caractéristiques suivantes :

- les pièces sont définies par une classe. L'ensemble des pièces aléatoires peut être changer facilement.
- les 3 pièces à placer sont choisis aléatoirement et s'affichent dans la zone prévue à cet effet.
- On peut déplacer les pièces dans le plateau de jeu, en respectant les règles du jeu. Le placement des pièces se fera à minima par une saisie au clavier des coordonnées ou l'on souhaite disposer la pièces.
- le plateau est mis à jour si des lignes ou des colonnes doivent être détruites.
- le score est mis à jour.
- la partie se termine si plus aucune pièce n'est positionnable dans le plateau.

(étape 2) les options suivantes doivent être ajoutées au jeu

- le jeu permet le passage de paramètres au lancement du programme pour connaître l'aide, la version du jeu et le nom des auteurs
- on doit pouvoir passer un fichier de configuration au programme qui donne l'ensemble des pièces avec lesquelles on veut jouer. Il faudra donner un formalisme pour décrire une pièce dans un fichier. Cette option doit permettre de jouer avec d'autres pièces que celles utilisées par défaut.
- on doit pouvoir sauvegarder les 5 meilleurs scores dans un fichier et les afficher dans le jeu. Il faudra prévoir la saisie des noms des joueurs.

(étape 3) les options avancées suivantes doivent être possible

- il est possible de déplacer les pièces à positionner directement au clavier avec les flèches et valider la position par appuie sur la touche entrée.

- il est possible de sauvegarder une partie dans un fichier et la reprendre plus tard
- (étape 4) — il est possible de défaire des actions du jeu. Cela signifie que l'on doit pouvoir annuler des positionnements de pièces, et également la destruction de lignes/colonne correspondant. Cette option doit permettre de revenir en arrière dans le jeu, jusqu'au tout début. Aide : il va falloir sauvegarder l'ensemble des actions qui sont effectuées.



Concernant votre organisation, le travail est à réaliser en groupe de 4 étudiants. Vous devrez saisir la composition de votre groupe de travail sur l'espace projet de l'ENT / Moodle du cours site du cours au plus tard le 30 novembre 2016 à 23h59. Les retardataires verront leur note de projet pénalisée de 2 points.

Enfin, vous devrez rendre sur l'espace projet de l'ENT/ Moodle du cours –au plus tard le 15 janvier 2017 à 23h59– une archive compressée (au format **tar.gz** ou **tar.bz2** uniquement) contenant un répertoire portant le nom de votre groupe (e.g. groupe-12) et dans lequel on trouvera<sup>2</sup> :

- les sources de votre programme ;
- un fichier texte<sup>3</sup> mentionnant vos noms et prénoms ;
- un fichier texte<sup>4</sup> expliquant comment compiler (en ligne de commande) et utiliser votre programme<sup>5</sup> ;
- un rapport<sup>6</sup> au format **pdf** (10 pages maximum, annexes comprises, hors pages liminaires, police 11 points pour le corps, marges à 2cm) mentionnant vos noms et prénoms et décrivant votre analyse, vos choix, vos tests, les remarques et commentaires, ...

Bon Courage. . .

---

2. Le non-respect de ces consignes sera indubitablement sanctionné.

3. Typiquement, un fichier `Auteurs.txt`.

4. Typiquement, un fichier `LisezMoi.txt`.

5. L'utilisation d'un `Makefile` est fortement conseillée.

6. Le rapport devra être écrit en français et respecter au mieux les règles typographiques françaises.