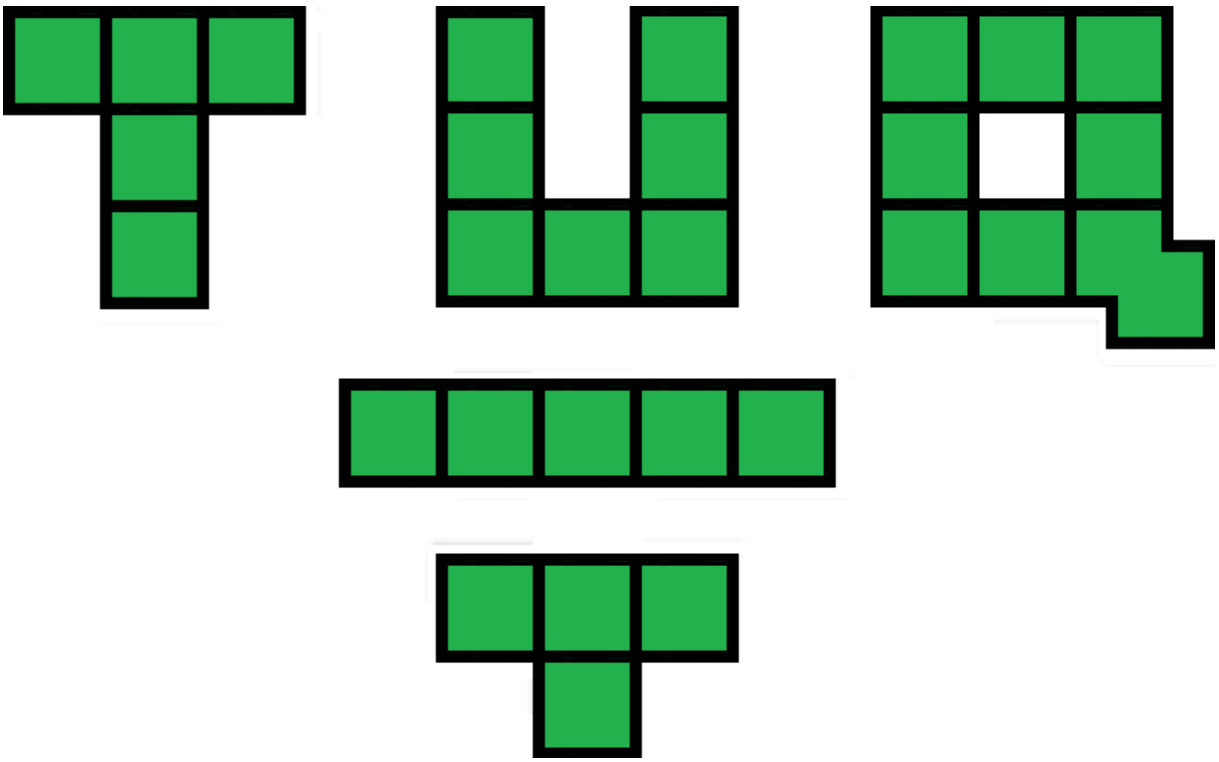
Informatique et Science du Numérique



Tuteurs : M.Guenego et M.Castagnetto

QUENTEL Thomas TS2
UDDIN Tanvir TS2

Lycée Jean Monnet 2018-2019

Sommaire :

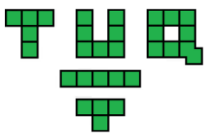
I. Règles du jeu:

Le TUQ se joue sur un quadrillage de 19x11. Des formes géométriques composées de carrés tombent depuis le haut de l'écran. Le joueur doit contrôler la chute de ces pièces :

- En les déplaçant sur les côtés
- En les faisant tourner sur elles-mêmes

Il n'est pas possible de ralentir la chute des pièces.
Le joueur doit ainsi empiler les pièces sur le bas de l'écran. Les pièces tombent une par une et fusionnent avec le reste des pièces déjà tombées une fois parvenues en bas. Le but du jeu est de former des lignes : lorsqu'un bloc termine une ou plusieurs lignes, les lignes en question disparaissent. Les pièces se trouvant au-dessus de la ligne disparue sont alors décalées vers le bas. En complétant des lignes, nous pouvons donc empêcher la pile de pièces d'atteindre le haut de l'écran : si cela se produit, la partie est alors terminée.
Le score augmente à chaque fois qu'une pièce touche la pile (+3 points) et lorsque nous complétons une ligne (+10 points).

II. Visuel du jeu :

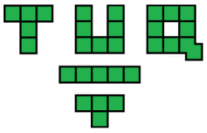


LEVEL:

FACILE

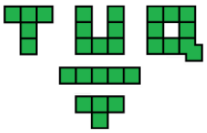
MOYEN

DIFFICILE



VOTRE PSEUDO:
Bill_25_

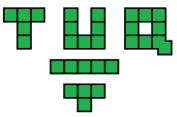
RETOUR VALIDER

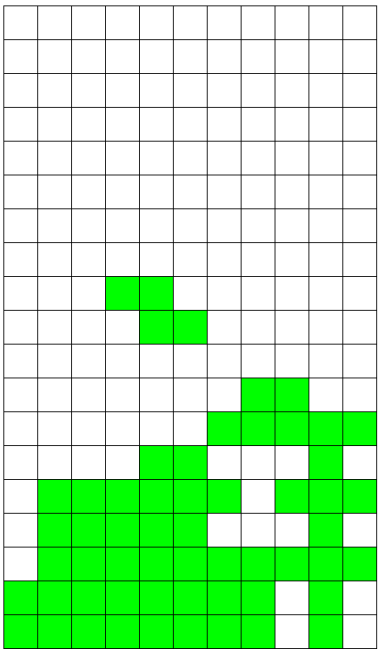


MUSIQUE:

Journey's Reflection
Days Past
Canon in D Major
Ambient Orchestral Piece

RETOUR





JOUEUR: Bill_25
LEVEL: 0
LIGNE: 1
SCORE: 58
TOP SCORE: 75

PERDU !

JOUEUR: Bill_25

LIGNE: 1

SCORE: 76

TOP SCORE: 76

REJOUER QUITTER

III. Cahier des charges :

-Un écran d'accueil

-Un choix de niveau

-Une interface de jeu

-Une interface en cas de défaite

-Système de score

-Sauvegarde de score avec un pseudo

-Rendre le code le plus lisible et le plus clair possible

Ce projet a été inspiré du célèbre et ancien jeu Tetris conçu par Alekseï Pajitinov en 1984. Pour faire ce projet, nous avons utilisé le langage Processing, basé sur le langage Java. Nous avons aussi utilisé quelques logiciels pour faire un peu de graphisme ; Paint et Photofiltre.

IV. Répartition des tâches :

Date	Objectif 1	Personne	Fait	Objectif 2	Personne	Fait
19 décembre 2018	Faire apparaitre une pièce aléatoirement à la position 4	Thomas		Faire le design autour (Titre, Level, Lignes…)	Tanvir	
6 février 2019	Gestion de la descente des pièces	Thomas			Tanvir	
27 février 2019	Gestion des déplacements des pièces	Thomas		Fonction pour vérifier si la pièce touche le bas	Tanvir	
6 mars 2019	Afficher la pièce suivante dans le petit cadre	Thomas		Fonction pour vérifier si la pièce touche le bas	Tanvir	
6 mars 2019	Gestion du score	Thomas			Tanvir	
13 mars 2019	Faire le GAME OVER quand une pièce en ligne 1	Thomas		Gestion des rotations des pièces	Tanvir	
20 mars 2019	Création de plusieurs niveaux	Thomas		Gestion des rotations des pièces	Tanvir	
20 mars 2019	Création d'un pseudo	Thomas			Tanvir	
20 mars 2019	Gestion de sauvegarder le top score avec pseudo	Thomas			Tanvir	
27 mars 2019	Descente rapide des pièces (Flèche du bas)	Thomas		Gestion des rotations des pièces	Tanvir	
20 avril 2019	Musique	Thomas		Gestion des rotations des pièces	Tanvir	
15 mai 2019	Mettre les commentaires manquants	Thomas		Simplification du programme	Tanvir	

V. Moyen de communication :

La communication au sein de l'équipe s'est fait principalement à l'école pendant les intercours ou pendant le cours d'ISN ou bien par mail si nous avions beaucoup de chose à dire et sinon par Snapchat si c'est juste pour avoir quelques informations. Nous avons pris quelques fois du temps en dehors des cours via Discord pour faire des mises au point et s'entraider sur certains points.

VI. Ma participation au projet :

Je me suis beaucoup occupé de la partie brute du projet. C'est-à-dire sur comment nous pouvions faire pour créer le quadrillage, descendre les pièces, les déplacer, ou bien les pivoter avant de faire la moindre ligne de code avec bien entendu régulièrement des idées de Tanvir que j'organisais par la suite.

Apparition des pièces :

Tout d'abord, suite à la création d'un quadrillage de 11x19 en utilisant une double liste remplie de 0, j'ai fait apparaître une pièce aléatoirement en haut du quadrillage. Pour cela, j'ai fait des petites doubles listes pour chaque pièce avec des 0 et des 1. Les 0 signifient du vide et les 1 du dur. J'avais un très grand quadrillage de 11x19 et des petits quadrillages de 4x4 et j'ai donc dû gérer la différence de taille en plaçant au bon endroit la pièce. Les différentes pièces sont d'abord dans une double liste puis dans un dictionnaire afin de faire de l'aléatoire par la suite. Pour l'aléatoire j'ai utilisé le « random » proposé par Processing qui renvoie un nombre. Ce nombre je le convertis ensuite en chaîne de caractère afin d'avoir le nom de la pièce (par exemple pieceLHaut). Puis avec le nom de la pièce je retrouve la double liste via le dictionnaire créé en amont.

La descente des pièces :

Dans la descente des pièces, je me suis juste occupé de faire en sorte que la pièce descende sans jamais s'arrêter. Pour cela, j'ai fait un scan du bas vers le haut et de droite à gauche en vérifiant si la case de la ligne au-dessus est un 1 ; si c'est un 1, alors il devient un 0 et la case où l'on se trouve devient un 1.

Le déplacement des pièces :

Puis est venu le déplacement des pièces vers la gauche et la droite. J'ai tout d'abord essayé de faire un scan du quadrillage de haut en bas et de droite à gauche (pour aller vers la gauche) et de gauche à droite (pour aller vers la droite) en réutilisant le même système que pour la descente des pièces. Mais je me suis vite rendu compte d'un problème ; si nous prenons la pièce L par exemple et que nous voulons la mettre complètement à droite, elle va se transformer en J car le L est formé de cette manière :

```
1  0
1  0
1  1
```

Donc si nous faisons comme pour la descente des pièces, les 1 vont venir prendre la place des 0 même sur la bordure du quadrillage. Cela est donc aussi problématique pour le J, le T, le S et le Z. C'est aussi problématique pour arrêter les pièces en bas du quadrillage. Nous avons donc décidé avec Tanvir de faire des « cases fantômes », ces cases fantômes vont être des cases où les 1 ne pourront pas aller mais elles restent tout de même des cases vides. Nous avons donné le numéro 2 à ces cases.

Pièce suivante :

Par la suite, j'ai mis en place une sauvegarde de la pièce qui suit pour pouvoir l'afficher dans un petit cadre. J'ai décidé de faire des images avec Paint pour les afficher afin de faire une petite pause sur les listes et programmer d'une autre façon.

La gestion du score :

Dans la gestion du score, j'ai tout d'abord fait un scan de bas en haut et de droite à gauche du quadrillage pour vérifier si une ligne est remplie de onze 9 (Le 9 symbolise les cases qui ne bougent plus, cela a été fait par Tanvir). Si la ligne est remplie alors tous les 9 de la ligne redeviennent des 0 et tous les 9 des lignes au-dessus deviennent des 1 afin de faire descendre les lignes. A chaque ligne complétée, nous ajoutons simplement 1 au nombre de ligne réussi ainsi que 10 au score total. Il y a aussi la possibilité de monter son score en arrivant simplement à poser une pièce, ce qui ajoute 3 points pour chaque pièce.

Lorsque le joueur atteint des intervalles de score, la vitesse de la descente augmente en modifiant le frameRate().

La sauvegarde des pseudos et des scores :

Pour sauvegarder les pseudos et les scores, j'ai utilisé un fichier JSON incluant un dictionnaire avec les pseudos et le meilleur score de chaque pseudo. Pour laisser l'utilisateur entrer son pseudo c'est simplement par le biais d'une fonction qui vérifie sur quelle touche l'utilisateur appuie.

La musique :

Pour la musique, j'ai utilisé la librairie Minim avec la possibilité de l'utilisateur à choisir entre 4 musiques libres de droits ; il fallait juste mentionner les auteurs.

VII. Licences :

Les musiques ont été prises d'Internet avec comme licence BY qui nous autorise à partager et à adapter sous condition d'attribuer (Créditer, Intégrer un lien vers la licence et Indiquer s'il y a eu des modifications : ce qui n'est pas le cas). (https://creativecommons.org/licenses/by/3.0/deed.fr)

Pour notre projet nous avons mis comme licence BY SA comme demandé par le jeu Tetris ; Donc avec le droit de partager, d'adapter, de vendre sous conditions de réutiliser la même licence et d'attribuer. (https://creativecommons.org/licenses/by-sa/4.0/)

VIII. Les difficultés rencontrées :

Au départ, nous étions partis sur une simple liste avec une taille de 208 pour faire le quadrillage. Mais dès que nous avons commencé à essayer de faire apparaître quelque chose dedans, nous avons rapidement vu que c'était très problématique. En effet, nous ne distinguions pas les lignes des colonnes et nous ne savions pas vraiment où faire apparaître quelque chose dans le quadrillage. Donc nous avons vite abandonné cette idée. Suite à un bon moment de réflexion, nous nous sommes rappelés qu'il était possible de faire des dictionnaires en Python et nous nous sommes dit que ça devait exister aussi en Processing. Nous avons donc fait des recherches sur le site de Processing et nous sommes tombés sur les doubles listes qui nous ont bien plus. Nous avons donc fait un quadrillage avec une double liste de la forme {{0,x9,0},x17,{0,x9,0}} afin de séparer les lignes. Cela a été très utile pour pouvoir faire des scans du quadrillage ligne après ligne en utilisant des boucles for() avec en général i pour le numéro de la ligne et j pour le numéro de la colonne.

Puis j'ai eu un autre gros problème lors des déplacements des pièces comme cité plus haut avec les « cases fantômes ».

Le 20 mars, j'ai eu un problème très surprenant qui m'a pris plusieurs jours à résoudre avec les professeurs. Il s'agit d'une fonction qui permet de vérifier si l'utilisateur a déjà joué auparavant pour récupérer son meilleur score. J'avais fait pour cela une condition if (nomJoueur == pseudos[i]) dans une boucle for pour parcourir la liste de chaîne de caractères « pseudos » incluant tous les pseudos déjà existants ; et nomJoueur est la variable qui contient le pseudo de l'utilisateur. Cependant cette condition fonctionne dans le cas où la liste n'a pas été créée via un fichier JSON. Lorsque la liste a été remplie par le chargement d'un fichier JSON il faut utiliser la méthode equals() de la classe str() : nomJoueur.equals(pseudos[i]).

IX. Perspectives :

Pour améliorer le jeu, nous pourrions y ajouter un peu de couleur, améliorer la fluidité en simplifiant un maximum le programme. Nous pouvons aussi ajouter un système de sauvegarde en ligne pour faire un classement à travers le monde.

X. Bilan Personnel :

Ce projet m'a permis tout d'abord d'apprendre un nouveau langage (le Processing) plutôt sympathique pour faire des petites choses visuelles assez rapidement (ça me change du Python). Je me suis senti quand même assez limité avec ce langage car il reste un langage simplifié. De plus, nous n'avions pas vraiment le temps d'apprendre réellement ce langage (Approfondir la POO m'aurais bien intéressé pour voir la différence de philosophie entre Python et Processing. Je pense que je vais l'approfondir par la suite). Le projet m'a aussi permis de me rapprocher de Tanvir en travaillant en équipe dans une bonne humeur et réussir à lui faire confiance. Ce projet a réussi à me stimuler mentalement avec la résolution de différents problèmes et avec les réflexions que nous avons pu faire pour rendre ce projet possible.