

420-V12-SF
PROGRAMMATION DE JEUX VIDÉO I
TRAVAIL PRATIQUE #3
LE JEU DE TETRIS
PONDÉRATION : 10%

OBJECTIFS PÉDAGOGIQUES

Ce travail vise à familiariser l'étudiante ou l'étudiant avec les objectifs suivants :

- Traduire l'algorithme dans le langage de programmation. (compétence 016S-2)
- Compiler le programme (compétence 016S-3)
- Valider le programme (compétence 016S-4)
- Procéder à la programmation de l'interface utilisateur. (compétence 016X-5)

De même, il permet à l'étudiante ou à l'étudiant d'appliquer les standards de conception, de documentation et de programmation.

MISE EN CONTEXTE ET MANDAT

Pour ce dernier travail, vous devez réaliser une version similaire au jeu bien connu de Tetris. Il existe une grande variété de jeux en ligne qui illustrent les mécaniques du jeu. Ces mécaniques ne seront donc pas énumérées dans ce document.

SPÉCIFICATIONS DE L'APPLICATION

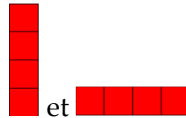
Votre jeu doit notamment respecter les contraintes suivantes :

- ☐ (C) Le titre du jeu doit apparaître dans la barre de titre
- ☐ (C) Le joueur doit pouvoir quitter le jeu à tout instant.
- ☐ (C) Au début de la partie, une nouvelle pièce générée aléatoirement est placée au centre de la surface de jeu et en haut de celle-ci.

Votre jeu doit minimalement supporter les pièces de type « carré »



Pour obtenir des points additionnels, vous devez supporter (dans l'ordre) les pièces suivantes :



et



- ☐ (C) Une pièce doit pouvoir être déplacée vers la gauche lorsqu'aucun bloc ne l'en empêche.
 - ☐ (C) Une pièce doit pouvoir être déplacée vers la droite lorsqu'aucun bloc ne l'en empêche.
 - ☐ (C) Une pièce doit pouvoir être déplacée vers le bas lorsqu'aucun bloc ne l'en empêche.
 - ☐ (C) Une pièce doit être déplacée vers le bas lorsqu'aucun bloc ne l'en empêche à chaque seconde environ. Le délai entre les mouvements verticaux peut varier selon le nombre de lignes complétées.
 - ☐ (C) Une attention particulière doit être apportée à l'esthétisme général de l'application.
 - ☐ (1) Une pièce ne peut pas être déplacée vers la gauche lorsqu'elle est située à l'extrême gauche de la surface de jeu ou lorsqu'une pièce figée l'en empêche.
 - ☐ (1) Une pièce ne peut pas être déplacée vers la droite lorsqu'elle est située à l'extrême droite de la surface de jeu ou lorsqu'une pièce figée l'en empêche.
 - ☐ (1) Une pièce ne peut pas être déplacée vers le bas lorsqu'elle est située en bas de la surface de jeu ou lorsqu'une pièce figée l'en empêche.
 - ☐ (1) Une pièce qui ne peut plus descendre doit être figée dans la grille de jeu.
-

- ☐ (1) Une nouvelle pièce est générée aléatoirement après qu'une pièce ait été gelée dans la grille de jeu.
- ☐ (1) Une ligne complète doit être retirée de la surface de jeu. En particulier, les blocs figés situés au-dessus de la ligne en question doivent être décalés vers le bas. En outre si plusieurs lignes sont complétées d'un seul coup, elles doivent toutes être retirées correctement avec les décalages appropriés réalisés.
- ☐ (1) Un système de pointage doit être mis en place. Le pointage doit être mis à jour lorsque cela est opportun (ex. lorsque une ou des lignes sont complétées).
- ☐ (1) Il doit être possible de configurer le jeu. Les paramètres configurables doivent pouvoir être modifiés dans un formulaire distinct du formulaire principal dont l'esthétisme aura été soigné. En particulier, le formulaire devra utiliser les contrôles les plus adaptés aux types de données à modifier. L'affichage du formulaire de configuration doit mettre le jeu en pause.

Il doit être possible de modifier les paramètres suivants :

- Le nombre de lignes de la surface de jeu
- Le nombre de colonnes de la surface de jeu
- L'activation et la désactivation des effets sonores du jeu.

En terminant, une modification des paramètres de jeu entraîne la fin de la partie courante. Les nouveaux paramètres doivent être pris en compte dans la prochaine partie.

- ☐ (2) Le jeu doit permettre la rotation des pièces.
- ☐ (2) La fin de partie doit être correctement détectée lorsqu'une nouvelle pièce ne peut être générée.
- ☐ (2) Le joueur doit être informé que la partie est terminée.
- ☐ (2) Il doit être possible de recommencer une partie (sans utiliser `Application.Restart()`)
- ☐ (2) À la fin de la partie, un formulaire distinct du formulaire principal doit afficher les statistiques de la dernière partie jouée. Le formulaire en question doit minimalement afficher les statistiques suivantes :
 - Nombre de pièces de chaque type
 - Pourcentage de pièces de chaque type
 - Temps de jeu

Une attention particulière doit être portée à l'esthétisme du formulaire de statistiques.

- ☐ Le jeu doit être programmé en utilisant les standards de programmation qui vous ont été communiqués en classe en début de session. En particulier, le code doit :
 - o Être correctement indenté et ne contenir qu'une seule instruction par ligne.
 - o Utiliser les structures de contrôles adéquates (`while`, `for`, `if`, etc).

- Être séparé adéquatement en fonctions / méthodes pour faciliter la lisibilité et la réutilisation.
- Utiliser des assertions aux endroits où cela est pertinent.

Vous devez également commenter à profusion les instructions, les déclarations des variables et les entêtes de méthodes.

Tests unitaires.

Votre jeu doit effectuer plusieurs tests unitaires pour valider le fonctionnement de certaines fonctionnalités. Pour les besoins du travail, vos méthodes de tests doivent être appelées à la toute fin de la méthode appelée lors du chargement du formulaire principal (événement Load).

Les tests suivants doivent minimalement être programmés :

- ☐ (1) Retrait d'une ligne complète seule dans la surface de jeu.
- ☐ (1) Retrait d'une ligne complète avec des blocs à décaler.
- ☐ (1) Retrait de deux lignes complètes consécutives
- ☐ (1) Retrait de deux lignes complètes non consécutives
- ☐ (1) Retrait de trois lignes complètes
- ☐ (1) Retrait de quatre lignes complètes.
- ☐ (2) Rotation d'un bloc seul placé au centre de la surface de jeu
- ☐ (2) Rotation d'un bloc seul placé à gauche de la surface de jeu.
- ☐ (2) Rotation d'un bloc seul placé à droite de la surface de jeu.
- ☐ (2) Rotation d'un bloc avec des blocs situés autour de lui dans la surface de jeu qui empêche la rotation.
- ☐ (2) Détection de la fin de la partie (partie terminée)
- ☐ (2) Détection de la fin de la partie (partie non terminée)

Document de remise.

- ☐ Vous devez compléter le document RemiseTP3.docx ainsi que la grille d'auto-évaluation qui s'y trouve.

CONTEXTE DE RÉALISATION ET DÉMARCHÉ DE DÉVELOPPEMENT

Ce travail pratique doit être réalisé **en équipe de 2 en fonction des mandats qui sont définis**. À ce titre, les numéros mentionnés entre parenthèses dans la section précédente identifient les mandats : **C** ➔ Mandat commun, **1** ➔ Mandat 1, **2** ➔ Mandat 2. Vous pouvez aussi référer à la grille d'évaluation pour avoir une description détaillée des mandats.

Les pages qui suivent résument les étapes du projet ainsi que les biens livrables devant être remis à la fin de chacune d'elles.

Étape 1: Analyse de la situation

Cette étape est partiellement rattachée au cours « Algorithmes et structure de données I ».

Vous devez aussi prendre connaissance de la demande et répartir entre les deux coéquipiers les deux mandats qui vous sont confiés. Veuillez noter que **la correction sera faite par mandat**, i.-e. les deux coéquipiers n'auront pas nécessairement la même note.

Étape 2: Mise au point et validation des algorithmes

Cette étape est rattachée au cours « Algorithmes et structure de données I ».

Étape 3: Programmation de l'application

Cette étape est rattachée au cours « Programmation de jeux vidéo I ».

Vous devez programmer les algorithmes conçus lors des deux premières étapes dans le projet de base fourni sur LEA.

Attention. Puisque la correction est individuelle, **chaque section de code que vous réaliserez devra être clairement identifiée**. Par exemple, si j'ajoute la méthode `CalculerPointage`, il faut que j'inscrive le code suivant.

```
// <ppoulin>
public int CalculerPointage
{
    return [...];
}
// </ppoulin>
```

Identification claire de la section de code produite à l'aide de l'initial de votre prénom et de votre nom de famille.

Notez qu'à des fins de correction, vous pourrez être convoqués à une rencontre visant à expliquer les parties de code que vous avez réalisées. Vous devez donc être en mesure de bien comprendre toutes les parties (fonctionnalités, code, etc) relatives à votre mandat.

Biens livrables :

- Code source de l'application documenté (selon les standards établis dans le cours), notamment en utilisant le /// pour documenter les fonctions / méthodes. Code des méthodes pour les tests unitaires.

Méthode:

- Vous devez remettre votre code source dans **une seule archive .zip par équipe** identifiée en fonction des noms des deux coéquipiers. Par exemple, pour l'équipe formée de Pierre Poulin et François Paradis, l'archive devrait être nommée TP3_PPoulinFParadis.zip.
- Cette archive doit être déposée sur LÉA avant le 7 décembre 2016 à 7h59.

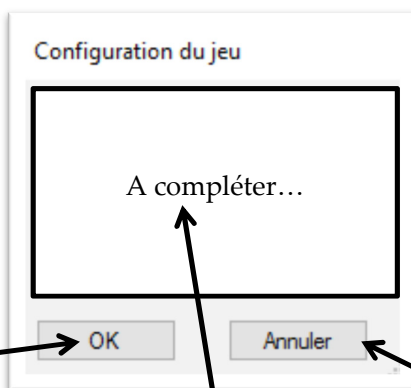
MODALITÉS D'ÉVALUATION

Vous trouverez dans le document `RemiseTP3.docx` la grille de correction qui sera utilisée pour le travail. **Cette grille indique la pondération accordée à chacune des parties du projet.**

- Tous les **biens livrables** de chacune des étapes devront être **remis à temps** et selon les modalités spécifiées.
- Nous rappelons que la **présence à tous les cours (de corps et d'esprit)** est **FORTEMENT RECOMMANDÉE**.
- Je vous recommande aussi de profiter de toutes les périodes de disponibilité qui vous sont offertes pendant la semaine, notamment celles où je suis présent en laboratoire.

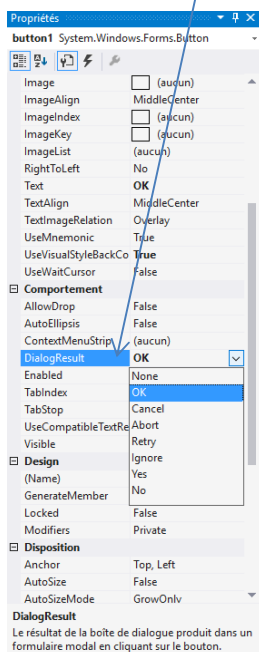
ANNEXE 1 FONCTIONNALITÉS DEVANT ÊTRE OFFERTES PAR LE JEU

Options devant être configurées pendant la partie



Si le joueur clique sur OK pour fermer le dialogue, les changements doivent être appliqués et le jeu ré initialisé.

Note : Il vous est possible de faire en sorte qu'un bouton de votre formulaire se comporte comme un bouton OK d'une boîte de message standard en spécifiant le « DialogResult » associé au contrôle :



Si le joueur clique sur Annuler pour fermer le dialogue, les changements ne doivent pas être appliqués et le jeu ne doit pas être réinitialisé.

Pour configurer le jeu, utilisez les contrôles que vous connaissez et qui vous semblent le plus appropriés pour le type de données à éditer.