582-555-SF
CONCEPTION ET DÉVELOPPEMENT D'UN JEU VIDÉO
TRAVAIL PRATIQUE #2
CRÉATION D'UN ÉDITEUR DE NIVEAU DE JEU
PONDÉRATION : 5%

OBJECTIFS PÉDAGOGIQUES

Ce travail vise à familiariser l'étudiante ou l'étudiant avec la conception et le développement d'une application dans un environnement graphique (compétence 017C). Plus spécifiquement, il vise le développement des éléments de compétence suivants :

- 017C - 1 - Établir le cadre général de l'application.
- 017C - 2 - Préparer le travail de développement de l'application.
- 017C - 3 - Modéliser l'application.
- 017C - 4 - Programmer l'application.
- 017C - 5 - Produire la documentation relative à l'application.

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

Le développement d'un engin de jeu sous-entend la création d'un ensemble d'outils permettant de faciliter la production de jeux. Un engin de jeu contient généralement plusieurs éditeurs qui permettent d'accélérer le travail des concepteurs et des artistes. Par exemple, un engin peut avoir un éditeur de « monde » ou un éditeur spécialisé dans l'animation ou dans les effets spéciaux.

L'objectif de ce travail est de réaliser un éditeur de jeu de "battle arena" qui permettra aux concepteurs de niveaux de créer et de modifier facilement les éléments d'une carte composée de plusieurs images juxtaposées. Ce travail vous permettra de développer des outils qui pourront être réutilisés ultérieurement dans ce cours.

SPÉCIFICATIONS DE L'APPLICATION

Votre application doit respecter les contraintes suivantes :

- Elle doit être implémentée en utilisant le patron de conception MVC.

- L'interface générale de votre application doit au minimum contenir :
  - Une barre de menu avec un menu « Fichier » permettant de :
    - Créer une nouvelle carte. Une fenêtre doit demander à l'utilisateur la taille de la carte en nombre de cases en hauteur et en largeur (Minimum 10 x 10).
    - Fermer la carte en cours d'édition (confirmer la sauvegarde de la carte avant de la fermer).
    - Ouvrir une carte existante (Format XML).
    - Sauvegarder la carte en cours d'édition (Format XML).
    - Valider la carte. L'application mentionnera si la carte est valide ou non.
    - Quitter l'application (et demander une confirmation à l'utilisateur si une carte est présentement en cours d'édition).
  - Un menu « ? » ayant un bouton « À propos » ouvrant une fenêtre avec vos noms.
  - Une barre de statut indiquant en temps réel la case surlignée (X, Y).
  - Une barre ou boite d'outils affichant toutes les tuiles disponibles dans l'application (voir plus bas).
    - Une partie de la barre / boîte d'outils permettra de gérer les étages.
  - Une zone d'édition centrale dans laquelle vous éditez votre carte en tant que tel (Panel recommandé).
  - Une zone de propriétés pour la tuile active dans la zone d'édition centrale.

- L'interface générale doit pouvoir se redimensionner efficacement.
  - La zone d'édition centrale doit prendre le maximum d'espace disponible

- La zone d'édition devra être comme suit :
  - Prendra place dans la partie centrale.
  - Si la carte est trop grande, des barres de défilement devront être affichées.
  - Avoir en arrière-plan une grille quadrillée où chaque case représente une tuile du niveau. La grille ne doit être affichée que lorsqu'une carte est ouverte.
  - Lorsqu'une tuile est sélectionnée dans la barre d'outils, un clic sur une case de la zone d'édition transformera la case en une case de ce type.
  - Les tuiles se trouvant sur la zone d'édition pourront être sélectionnées. Il doit y avoir une distinction entre un élément sélectionné et non-sélectionné.
  - Une tuile sélectionnée dans la zone d'édition pourra être :
    - Déplacée (par une opération de « drag & drop » avec la souris).
    - Pendant son déplacement, la case qu'elle « survole » devra être soulignée. On appelle cette case, la case active.
    - Placés dans la case active si la touche de souris est relâchée (confirmation demandé si la case n'était pas vide).
    - Supprimée (par exemple avec le bouton « Delete » du clavier, ou alors par une "tuile de suppression" qui ramène la case à l'état vide.).
  - Par rapport aux étages. Il sera possible d'afficher un étage à la fois, ou l'ensemble des étages (l'élément le plus élevé sera alors affiché)

- L'édition de la carte se fera grâce à la barre / boîte d'outils. Il y aura plusieurs types de tuile. Seulement un outil à la fois peut être actif.
  - Il devra y avoir un outil de type « Tuile » pour chaque tuile mise à votre disposition. Un clic sur la grille entraînera la création de la tuile sur la carte à la position du clic.
  - Une section de la zone permettra de gérer les étages. Par vertu de simplicité, nous allons fixer le nombre d'étages possibles. 5 Serait un bon nombre, bien que le minimum requis soit de 3. Un slider / drop down list / etc, permet de sélectionner l'étage en cours, plus une section qui permet d'afficher le tout On voit le bloc le plus élevé pour chaque case.
  - La tuile ajoutée est sur l'étage en cours, sauf si on affiche le tout, dans quel cas on ajoute la toile au-dessus de l'objet le plus élevé déjà présent (empilement)

- La zone de propriétés est normalement vide. Elle se meuble de certains contrôles si une tuile dispose d'options particulières. L'utilisateur pourra ajuster les options selon les besoins.

- Certains objets pourront se déplacer selon un parcours précis. Nullement besoin de dessiner le parcours, mais ces objets auront une liste de position : des waypoints. Et des tracés de ligne seront illustrés entre ces points dans la zone d'édition. Une seule position implique que l'objet ne se déplacera pas. Un objet se déplace seulement sur l'étage où il se trouve.

- Validation de la carte. Pour être valide la carte doit
  - Avoir quatre "spawn points" (1 par joueur)
  - Au moins un spawn point pour la balle et un et un seul but (balle / but utilisé)
  - N'avoir aucune case vide (null) au premier étage
  - Respecter toutes les contraintes de chaque tuile particulière.

- Vous devez également développer un **mécanisme de gestion des erreurs** et du flot d'exécution du programme et informer l'utilisateur des erreurs survenues en cours d'exécution. De même, lorsqu'une erreur survient, vous devez fournir le maximum de renseignements à l'utilisateur.

TYPES DE TUILE (ET RÈGLES DE VALIDATION)

Notez bien que toutes les cases impliquent une case de plancher, sur lequel est déposé l'objet.

- Plancher vide
  - La case par défaut. Espace inoccupé (à distinguer d'une case vide, l'équivalent d'un « null »)

- *Bloc inaccessible*
  - Par-dessus lui doit se trouver un autre mur inaccessible, jusqu'au dernier étage

- Bloc accessible
  - Par-dessus lui DOIT se trouver un autre objet, peu importe le type.
  - Peut théoriquement être suspendu dans le vide

- But
  - Peut avoir une rotation ou non
  - 1, 2 ou 4 faces.
  - Déplaçable

- *Spawn point des joueurs*
  - Doit en avoir 4 pour avoir une carte valide

- Spawn point d'un bonus
  - Le bonus produit peut-être fixe ou faire partie d'une liste de bonus
  - Peut-être visible où mystérieux (style boite à bonus de Mario Kart)
  - Déplaçable
  - Propriétés : temps de respawn : random de i à j (si i et j sont égaux, il n'y a pas de random. j doit être >= à i)

- *Rampe*
  - Une rampe doit avoir au moins une autre rampe comme voisine directe et au plus deux.
  - Si a deux rampes voisines, ses deux voisines doivent être dans le même axe
  - A une propriété axe qui illustre la direction de la monté de la rampe.

- *Puit sans fond*
  - Aucune propriété (bin kin)
  - Seulement sur le premier plancher

- *Téléporteur*
  - Doit être voisin d'un mur ou une limite de la carte
  - Doit avoir une direction (haut / bas / gauche / droite). Dans cette direction se trouve un mur ou une limite de la carte.
  - Doit être lié à un et un seul autre téléporteur

- *Tourelle*
  - Type (Laser ou missile).
  - Points de vie (peut-être illimité / indestructible)
  - Déplaçable.

<u>Notez que tout ce qui est en italique à la contrainte qu'il doit avoir sous lui un autre objet (ne peut pas être suspendu dans le vide.</u>

Pour les différents types, faites-vous une courte liste, pas besoin d'être exhaustif.

Notez que pour la simplicité, nous ne ferons pas de validation sur le fait qu'une trajectoire d'objet déplaçable coupe un mur ou non, mais le fait de le voir devrait forcer l'utilisateur à se corriger.

CONTEXTE DE RÉALISATION ET DÉMARCHE DE DÉVELOPPEMENT

Ce travail pratique doit être réalisé **en équipe de 2 personnes**.

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 : Conception de l'application

Dans cette étape, vous devez :

- Préciser l'idée directrice du projet avec le professeur.
- Au besoin : rechercher et analyser des produits similaires que vous connaissez.
- Déterminer les caractéristiques fonctionnelles de l'application demandée.
- Examiner les contraintes imposées par l'environnement et par les outils de développement.
- Représenter l'interface de l'application et ce, de manière conforme aux exigences de l'ergonomie et de l'esthétisme (Papier, Photoshop, Winforms directement)
- Présenter votre projet pour approbation.
- Identifier les principales classes et autres éléments de votre application ainsi que les liens entre eux.

Biens livrables :

- Aucun. Validation formative en classe avec le professeur.

Méthode:

- Maquette d'interface réalisée à la main ou à l'aide d'un programme approprié. UML

Étape 2 : Programmation de l'application

Dans cette étape, vous devez programmer l'application et produire les interfaces utilisateurs réalisés à l'étape précédente. Plus spécifiquement, vous devez :

- Utiliser de manière appropriée les bibliothèques et autres fonctionnalités propres au langage C#.
- Coder les classes et autres éléments du modèle.
- Utiliser des outils de création des éléments d'interface.
- Valider correctement le fonctionnement de l'application.
- Produire et archiver toute l'information relative aux programmes (code source + documentation si nécessaire).

Biens livrables :

- Code source de l'application.

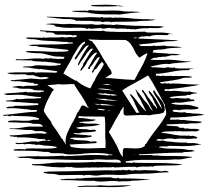
Méthode:

- Sur LEA, **Le mardi 6 octobre à minuit**

MODALITÉS D'ÉVALUATION

Vous trouverez à la fin de l'énoncé la grille de correction qui sera utilisée pour le travail. **Cette grille indique la pondération accordée à chacune des parties du projet**.

- Nous rappelons que la **présence à tous les cours** (**de corps et d'esprit**) est **OBLIGATOIRE** et est **ÉVALUÉE**. Toute absence non justifiée par un motif raisonnable[1] sera pénalisée.
- Dans l'éventualité où vous récupériez du code existant ailleurs (internet, MSDN, etc), vous devez clairement indiquer la source ainsi que la section de code en question. Tout travail plagié ou tout code récupéré d'une source externe et non mentionnée peut entraîner la note zéro (0) pour l'ensemble du travail.



Grille d'évaluation du travail pratique #1

Étape 2 (100%)	
<u>Interface utilisateur de l'application</u> Critères d'évaluation	
- Ergonomie, esthétisme et fluidité de l'interface	**15%**
<u>Codification de l'application</u> Critères d'évaluation	
- Déclaration et définition des classes : clarté, qualité et efficacité du code. Documentation et commentaires	**10%**
- Utilisation appropriée du patron de conception MVC.	**20%**
<u>Validation du fonctionnement de l'application</u> Critères d'évaluation	
- Création d'une nouvelle carte vide et ouverture d'une carte existante. Fermeture et sauvegarde de la carte en cours d'édition.	**10%**
- Menu, barre / boite d'outils et zone / dialogue de propriétés efficace	**10%**
- Ajout, déplacement et retrait des tuiles	**10%**
- Propriétés des tuiles, mécanique d'étages de la carte et paths des objets mobiles	**20%**
- Validation de la carte	**5%**
Total	**100%**