

582-525-SF
PROGRAMMATION OBJET AVANCÉE EN JEUX VIDÉO
TRAVAIL PRATIQUE #2
PATRONS DE CONCEPTION ET JEU
PONDÉRATION : 20%

OBJECTIFS PÉDAGOGIQUES

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

- Créer et modifier un modèle objet (compétences 016T-1 et 016T-2)
- Utiliser des modèles de conception (compétence 016T-1)
- Procéder à la codification d'une classe (compétence 016T-3)
- Générer la version exécutable du programme (compétence 016T-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

Ce travail pratique a d'abord pour but de mettre en pratique les concepts de patron de conception au travers la réalité d'un jeu interactif en temps réel.

De plus vous aurez une réflexion au niveau de la conception du jeu. Ce projet est un projet semi-libre. Vous pouvez le faire tel que décrit ou vous pouvez rebrasser complètement le concept, en autant que certaines contraintes de base soient respectées.

Pour ce faire, vous devez récupérer du code existant, en comprendre la structure et le fonctionnement, identifier les modifications ou améliorations à apporter puis programmer et tester ces dernières.

SPÉCIFICATIONS DE L'APPLICATION

- Le jeu à développer à la base est un jeu de labyrinthe style Pac-Man (plus Mouse Trap en fait : <https://www.youtube.com/watch?v=rYnYXJNjGmo>). Il s'agit en fait d'un TP semi-libre. Vous devriez suivre le concept de base, mais plusieurs options vont s'offrir à vous. Vous pouvez même dans une certaine limite, changer le design du jeu, en autant que les contraintes de base restent.

- Gru, le méchant de service, se déplace dans le labyrinthe. Il a un objectif à accomplir qui peut-être :
 - Ramasser tous les dollars qui trainent dans le labyrinthe.
 - Ramasser une série d'objets qui apparaissent au hasard dans le labyrinthe
 - Ramasser des objets qui apparaissent un à un dans le labyrinthe. Un objet apparaît à une position au hasard une fois que le précédent est ramassé.
 - Vaincre un certain nombre de policiers.
 - Plusieurs de ces objectifs à la fois
 - Etc.
- Une fois le ou les objectifs atteints, le vaisseau de Gru apparaît quelque part dans le labyrinthe. Vous devez le rejoindre pour gagner le niveau. Une fois le niveau complété, le suivant devrait être un peu plus difficile que le précédent (Option : si le vaisseau est touché par un policier, il respawn immédiatement ailleurs, peut-être pas immédiatement).
- Des policiers partent des quatre boîtes sur chaque côté du labyrinthe. À la base ils vont pourchasser Gru. Vous pouvez décider de vous limiter à quatre policiers et les faire "resawner" quelques instants après qu'ils aient été vaincus, ou alors en faire "spawner" de manière régulière et théoriquement illimitée, à votre choix.
- Les policiers font perdre une vie à Gru s'ils entrent en collision avec lui.
- Les policiers vont avoir un comportement stratégique pour attraper Gru, en voici un exemple (Mais vous pouvez implémenter le vôtre)
 - À la base le policier se promène au hasard dans une direction. Il ne peut pas aller en direction opposée à la sienne (comme les fantômes de Pac-Man) sauf s'il frappe un cul de sac. Il peut voir Gru sur l'avant et les côtés (L'avant étant la direction vers où il se déplace).
 - S'il voit Gru, il fonce vers lui.
 - S'il le perd de vue, il va vers la dernière position où il a vu Gru.
 - S'il ne voit pas Gru une fois la dernière position atteinte, il recommence à se promener au hasard.
 - S'il croise un autre policier, il lui communique la position de Gru ou la dernière position où il a été vu.
 - Le policier nouvellement informé ne peut pas changer immédiatement de direction et va même continuer son chemin un certain temps (ou prendre un dernier virage au hasard), mais va par la suite tenter de rejoindre lui aussi la dernière position de Gru.
 - NOTEZ BIEN : vous pouvez faire changer la vitesse des policiers, les faire tirer sur Gru, etc.

- À quelques endroits dans le labyrinthe se trouvent des armes. À la base elles fonctionnent comme les super-pilules de Pac-Man (Le contact vainc le policier, et ceux-ci ont peur de Gru). Mais au lieu de se déclencher immédiatement, Gru peut les accumuler et les déclencher au besoin. Vous pouvez trouver une autre fonction à l'arme de Gru, mais ça doit changer l'État des policiers.
- Gru a aussi trois minions (trois est une simple suggestion) avec lui par niveau. Il peut les relâcher dans le labyrinthe. Le problème est que les Minions ont un comportement absolument imprévisible. Vous pouvez les faire agir comme bon vous semble et leur durée de vie est assez limitée, mais ils devraient être dangereux à la fois pour Gru et pour les policiers. Un exemple : Minion1 est joyeux et inoffensif, Minion2 est potentiellement dangereux et Minion3 est ultraviolent. En fait, ça peut-être le même Minion qui change d'humeur au cours de son existence. Utilisez les Minions avec créativité, mais vous devrez expliquer comment vous le ferez.
- Les policiers peuvent réagir aux Minions, mais ne sont pas obligés de le faire si vous ne le souhaitez pas. (en fait ils ne savent même pas c'est quoi ces bibittes jaunes-là, et surtout parce que ça peut introduire un bon niveau de complexité que de faire interagir deux systèmes d'État ensemble) mais ça reste une option...
- Remarquez le téléporteur : Gru est venu l'installer avant de lancer ses forfaits. Il peut l'utiliser pour se sortir d'un mauvais pas, seulement, il n'a aucune idée où ça va le mener exactement. À vous de voir si les AI peuvent s'en servir ou non (à la base ce n'est pas requis).
- Élément supplémentaire. Vous pouvez ajouter des règles de jeu qui ne sont pas précisées ici. Soyez créatifs.
- Vous devez implémenter un tableau de jeu avec un score et des vies pour Gru. La partie se termine quand Gru n'a plus de vie. Il est possible de mettre le jeu en pause.
Bref faites en sorte que votre jeu soit le plus complet possible.

CONTRAINTES AU NIVEAU DES PATRONS DE CONCEPTION

- Les IA des personnages devront être déterminées par le Patron de conception États. Vous devez avoir deux IA différentes (le concept de base prévoit cela, si vous implémentez votre concept à vous, vous devez avoir tout de même deux systèmes d'états)
- Utilisez le patron de conception « Commande ». Votre "button mapping" devra venir d'un fichier (texte ou XML, peu importe), d'une classe de service ou d'un menu. Il serait donc simple pour le joueur de re-mapper ses boutons. Vos contrôleurs/claviers/souris vont entrer des données dans une structure de commande et c'est cette structure de commande qui va interagir avec gru
- Il serait de bon ton d'utiliser les patrons de conception précédemment étudiés. Le Patron Observateur pourrait être encore fort utile.

ASSURANCE-QUALITÉ

- Les tests unitaires ne sont pas requis pour ce projet, quoi que tout travail non-requis mais tout de même effectué pourra être considéré.
- Votre code doit compiler sans erreur ni avertissement en niveau 4 (le niveau que vous avez par défaut). Il se peut que du code qui vous est fourni ne respecte pas ces critères. Vous devrez alors apporter les modifications nécessaires pour éliminer les avertissements.
- Vous pouvez également utiliser des assertions pour sécuriser votre code et effectuer les validations de base.
[http://msdn.microsoft.com/en-us/library/system.diagnostics.debug.assert\(v=vs.110\).aspx](http://msdn.microsoft.com/en-us/library/system.diagnostics.debug.assert(v=vs.110).aspx)

CONTEXTE DE RÉALISATION ET DÉMARCHE DE DÉVELOPPEMENT

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

Étape 1: Exploration du code récupéré

Lecture et compréhension du code récupéré. Ce travail ne nécessite pas l'écriture d'une grande quantité de code. Un des défis qu'il présente est d'intégrer le code que vous devrez produire dans du code déjà existant et d'implémenter correctement les patrons de conception.

Aucun bien livrable.

Étape 2: Conception de l'application

Vous devrez joindre à votre projet un document de conception comprenant les sections suivantes

1- Design de votre jeu. Quels sont les éléments optionnels que vous choisissez si vous partez du concept de base. Sinon en quoi votre design change du concept original. Si vous suivez exactement le concept de base, voici les choix que vous aurez à faire.

- Objectif(s) de Gru (et mode de fuite finale)
- Mode de spawning des policiers
- Comportement (États) des policiers.
- Arme de Gru
- Minions
- Téléporteur
- Éléments supplémentaires et fonctionnalités diverses

2- Diagramme de classe de votre projet.

3- Description des comportements de Gru, des policiers et des minions et choix des états

4- Diagrammes d'états.

Biens livrables :

- Document de conception (joindre avec le projet final).

Méthode:

- Validation en classe avec le professeur.

Étape 3: Programmation de l'application

Programmation des fonctionnalités précédemment identifiées. Les fonctionnalités que vous devez livrer consistent en deux parties : La partie est fonctionnelle d'un bout à l'autre, avec les patrons de conception correctement implémentés.

Vous devez faire en sorte que votre jeu soit de la plus grande qualité possible.

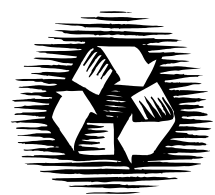
Biens livrables :

- Projet avec code source correctement documenté. (GhostDoc va vous sauver beaucoup de temps, avec cet outil, vous n'avez plus d'excuse). La description (summary), paramètres (param), retour (return) et références (see) doivent être inclus. (<https://msdn.microsoft.com/en-us/library/5ast78ax.aspx>) Application fonctionnelle et Tests unitaires complets
- Sur LEA pour **le lundi 14 septembre 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.

- Tous les **biens livrables** de chacune des étapes devront être **remis à temps** et selon les modalités spécifiées.
- 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 3

Codification des classes (016T-3)

Critères d'évaluation	
- Déclaration et définition des classes : clarté, qualité et efficacité du code. Application rigoureuse des standards de nomenclature et documentation du code selon les standards .Net	10%
- Exploitation judicieuse des possibilités du langage dans l'application des principes d'encapsulation, d'héritage et de polymorphisme et <u>utilisation pertinente des patrons de conception</u>	20%
- Qualité du document de conception.	10%

Génération de la version exécutable du programme (016T-5)

Critères d'évaluation				
Application fonctionnelle. Une partie peut-être jouée d'un bout sans bug et sans erreur, selon les spécifications demandées.			25%	
Comportements / États des Personnages non-joueurs bien implémentés et efficaces			25%	
Finition : score, écran de titre, menu, nouveau niveau après en avoir complété un, progression de la difficulté. Annonce du Game over			10%	
Amélioration du projet en place et situations exceptionnelles. Qualité générale du projet. Ajout supplémentaires. Fun factor remarquable.			Jusqu'à 10% de bonus discrétionnaire.	
TOTAL			100%	