
B65 – PROJET SYNTHÈSE

Hiver 2014

Contenue du document

Présentation générale	1
Introduction.....	1
Forme du cours B65.....	1
Déroulement général du projet.....	1
Contraintes générales.....	2
Contraintes associées à la démarche du projet.....	2
Contraintes techniques générales	3
Présentation des treize livrables	4
Livrable 1 – Document de conception	4
Livrable 2 – Document de planification générale	5
Livrable 3, 4 et 5 – Documents de suivi de projet pour les <i>Sprints 2, 3 et 4 (3 livrables)</i>	5
Livrables 6 et 7 – Évaluation de l’avancement du projet (<i>2 livrables</i>)	6
Livrable 8 – Livraison finale du projet.....	6
livrable 9 – Manuel de l’usager.....	6
Livrable 10 – Fichier <i>Lisez_moi.txt</i>	7
livrable 11 – Site Web	7
Livrable 12 – Présentation	8
Livrable 13 – Vidéo de présentation	8
Les projets proposés.....	8
Projet générique 1 – Gestion de données	9
Présentation	9
Contraintes	10
Projet générique 2 – Jeu vidéo	11
Présentation	11
Contraintes	11
Projets spécifiques.....	12
Projets industriels	12
Annexe 1 – Outils de conception Conception UML	13
Qu’est-ce que UML.....	13
Diagramme de classes	13
La classe.....	14
L’héritage.....	14

L'association	14
L'agrégation et la composition	15
Exemple complet	15
Annexe 2 – Outils de conception Conception d'interface usager	17
Besoins applicatifs du projet	17
Ergonomie en informatique	17
Outils recommandés pour le cours	17
Annexe 3 – Outils de conception « <i>Design pattern</i> »	19

PRÉSENTATION GÉNÉRALE

INTRODUCTION

Tel que stipulé dans le plan de cours, ce projet vous donne l'opportunité de réaliser entièrement une application informatique et d'en assurer la mise en œuvre complète. Ainsi, à partir d'un devis présentant une idée générale et certaines contraintes spécifiques, vous allez devoir implémenter une application fonctionnelle en passant par plusieurs étapes de production typiques d'un tel projet.

Comme l'indique le titre du cours, ce projet se veut une synthèse de certains apprentissages que vous avez accumulés pendant votre DEC. Il est donc essentiel de mettre à profit tous les aspects techniques que vous avez acquis afin de rendre hommage à l'informaticien(ne) que vous êtes rendu(e)!

FORME DU COURS B65

Le cours *Projet synthèse* se déroule à raison de trois heures par semaine en classe et vous demande de produire une somme de travail significative à la maison.

Aucune présentation ou évaluation formelle de nouvelle matière est prévue. Ainsi, aucun cours magistral n'est à l'horaire et aucune évaluation autre que celles du projet n'est à faire. Néanmoins, tous les projets requièrent l'approfondissement de nouvelles connaissances et de nouvelles compétences. Ce sera à vous d'identifier et d'acquérir les éléments requis au succès de votre projet.

Le projet à réaliser est à mi-chemin entre un projet libre et un projet imposé. Plusieurs devis sont présentés à l'étudiant qui doit choisir lequel il réalisera. Chaque devis présente des objectifs spécifiques plus ou moins contraignants mais laisse une grande place à l'interprétation et à la créativité.

DÉROULEMENT GÉNÉRAL DU PROJET

Pour donner suite aux notions que vous avez acquises lors des cours précédents, le développement et la gestion du projet s'effectuera selon la méthode [Scrum](#) qui est une des méthodes [Agile](#). Dans cet esprit, la réalisation du projet est découpée selon ces quatre *Sprints* distincts :

- Sprint 1. Identification, conception et planification
- Sprint 2. Implémentation des éléments essentiels
- Sprint 3. Augmentation fonctionnelle
- Sprint 4. Livraison finale

Pour chaque étape, vous aurez plusieurs évaluations différentes. En tout, le cours comportera treize évaluations distinctes!

Le projet qui vous est demandé est exigeant pour le temps que vous avez de disponible. Ainsi, vous allez devoir gérer vos activités très efficacement afin d'éviter les pièges typiques de tels projets. D'ailleurs, vous devez vous y mettre dès la première semaine pour ne pas accumuler de retard. Finalement, ne jamais hésiter à poser des questions sur la gestion du projet ou tout autre aspect technique.

CONTRAINTES GÉNÉRALES

Chacun des projets proposés comportent des contraintes qui leur sont propres et qui sont présentées à même leur devis respectifs. Néanmoins, chaque projet présente des éléments communs et plusieurs contraintes sont appliquées également sur tous les projets (sauf pour les exceptions mentionnées localement dans les devis).

Les contraintes les plus significatives du projet sont sans contredit les méthodes de travail et le respect des échéanciers qui sont demandées. Tout au long du projet, une démarche serrée doit être suivie et plusieurs évaluations permettront de garder un rythme de travail adéquat.

CONTRAINTES ASSOCIÉES À LA DÉMARCHE DU PROJET

Le projet est divisé en 4 *Sprints* distincts :

Sprint 1 – Identification, conception et planification : Cette phase vise à réaliser la mise sur pied du projet. Après s'être assuré de bien comprendre le devis, l'étudiant doit définir les éléments du projet qui lui sont propres et faire la conception technique du projet ainsi que la planification des activités.

Sprint 2 – Implémentation des éléments essentiels : Cette phase permet de produire les assises techniques du projet. À la fin de cette étape, on doit avoir une version préliminaire **fonctionnelle** du projet. En se basant sur les parties essentielles, on doit mettre en place l'infrastructure de l'application tout en offrant quelques fonctionnalités de bases.

Sprint 3 – Augmentation fonctionnelle : Cette phase consiste à atteindre un avancement intermédiaire du projet. Comme pour le *Sprint* précédent, on doit déterminer des objectifs intermédiaires pour lesquels de nouvelles fonctionnalités seront mise de l'avant. Généralement, l'application devrait déjà contenir l'essentiel de tous les modules.

Sprint 4 – Livraison finale : C'est ici qu'on peaufine l'application, qu'on fait les tests de fonctionnalité et que l'on apporte certaines modifications. Aussi, il faut terminer la réalisation de tous les documents finaux qui sont à remettre.

Il est important de bien planifier le projet afin qu'il soit réalisable dans le temps disponible considérant toutes les autres contraintes que vous avez. Le calendrier des activités donne une excellente vue d'ensemble du déroulement global du projet.

<i>Semaine</i>	<i>Nbr de semaines</i>	<i>Description des activités</i>	<i>Remise s et évaluations</i>	<i>Date de remise des travaux</i> <i>Toujours au début du cours sans faute!</i>
1	4	Sprint 1 22 % Conception et planification	Document de conception 20,0 % Document de planification générale 2,0 %	28 février
2				
3				
4				
5	3	Sprint 2 6,5 % Implémentation des éléments essentiels	Projet : Implémentation initiale 5,0 % Suivi de projet : <i>Sprint 2</i> 1,5 %	28 mars et 4 avril
6				
7				
8	3	Sprint 3 6,5 % Augmentation fonctionnelle	Projet : Augmentation fonctionnelle 5,0 % Suivi de projet : <i>Sprint 3</i> 1,5 %	25 avril et 2 mai
9				
10				
11	5	Sprint 4 65 % Livraison finale	Projet : Version finale de l'application 45,0 %	lors de la dernière semaine (date à déterminer)
12			Suivi de projet : <i>Sprint 4</i> 1,5 %	
13			Manuel de l'utilisateur 4,0 %	
14			Fichier <i>LisezMoi.txt</i> 1,5 %	
15			Site Web 3,0 %	
			Présentation du projet 5,0 %	
			Vidéo de présentation 5,0 %	

CONTRAINTES TECHNIQUES GÉNÉRALES

- Les projets se font seuls et aucune équipe n'est permise. Néanmoins, l'entre aide constructive est encouragée et souhaitée.
- Outre le respect des contraintes spécifiées dans les devis, les étudiants ont pleine liberté pour réaliser le projet qui les intéresse.
- Les outils de développement sont laissés à l'entière discrétion de l'étudiant en autant qu'ils soient disponibles sur la plateforme du cégep (*IDE*, langage de programmation, environnement de développement, outils de documentation et autres). Néanmoins, malgré la flexibilité qui est offerte, le contexte de ce projet se prête mal à l'apprentissage de nouveaux outils informatiques. Le temps disponible est déjà très serré pour le projet à réaliser. Il est fortement suggéré d'utiliser les outils connus et pour lesquels on désire développer une plus grande expertise.
- La validation de l'approche avec l'enseignant est essentielle par une discussion avec ce dernier des détails du projet lorsqu'ils sont identifiés.
- Les échéanciers ne sont pas négociables.
- La réalisation de ce projet repose beaucoup sur les aptitudes à solutionner les problèmes qui ce présenteront. D'ailleurs, être capable d'identifier les problématiques et savoir où trouver les solutions sont des aptitudes essentielles qui doivent être développés afin d'être mieux préparé pour le marché du travail.
- Pour quelque raison que ce soit, il sera permis de changer de projet en cours de session. Néanmoins, deux aspects sont à considérer :
 - a. Toute la démarche de conception et de production des documents demandés pour le sprint 1 est à refaire.
 - b. Une pénalité est appliquée pour chaque semaine de retard engendré selon cette formule :

$$\text{Pénalité} = 2.5\% \times (\text{Semaine courante de la session} - 1)$$

PRÉSENTATION DES TREIZE LIVRABLES

LIVRABLE 1 – DOCUMENT DE CONCEPTION

Avant d’amorcer la réalisation technique du projet il est essentiel d’en définir les objectifs et plusieurs aspects. Après le choix du projet, il importe à l’étudiant de bien comprendre le devis demandé et de préciser son intention. Ainsi, la première étape en est une de conception et de planification.

Cette première étape est cruciale et permet, grâce à un effort de réflexion et d’abstraction, de déterminer quelle est la vision complète du projet. Comme tous les projets de développement, les objectifs qui seront déterminés ne sont pas définitifs ou immuables mais permettent de fixer une ligne directrice efficace. Ainsi, il est fort probable, voir inévitable, que plusieurs changements viendront en cours de développement. En fait, il tout à fait naturel, pour ce genre de projet, de faire évoluer le produit en fonction du développement des changements de contraintes ou d’objectifs.

Dans cet esprit, les premières semaines de la session doivent servir à réfléchir sur la nature et la forme du projet et à documenter le développement à venir. Cette étape vous demande de produire un document présentant une conception étayée, approfondie et relativement détaillée du projet mis de l’avant. Ce document de conception servira de référence pendant la production de l’application et d’élément de validation technique de l’approche choisie.

Par le fait même, ce document de conception devient le devis technique du développement entier du projet. L’usage d’un tel document joins à la méthode de gestion de développement *Agile*, permettra de mettre toutes les chances de son côté afin d’atteindre les objectifs que l’étudiants déterminera. De plus, il est important de valider l’interprétation que vous pouvez faire d’un devis. Cette validation avec les personnes qui l’ont rédigée permet d’éviter, dès l’amorce du projet, la production d’une application qui diffère des intentions initiales.

Ce projet de fin d’étude est une excellente occasion de mettre de l’avant les notions de conception orientée objet. D’ailleurs, on demande de faire un effort particulier afin d’utiliser adéquatement les notions d’encapsulation, d’héritage et de polymorphisme lorsqu’elles sont applicables. Le niveau d’abstraction mis de l’avant sera l’un des points techniques évalués les plus importants (qui se reflétera dans votre diagramme CRC ou UML).

Voici les sections que votre rapport de conception doit présenter :

Présentation générale du projet (retour sur le devis général en quelques lignes)	1.0 %
Présentation précise du projet (l’interprétation du devis et la touche personnelle de l’étudiant)	3.0 %
Aspects techniques de la conception :	
a. cas d’usage;	3.0 %
b. diagramme CRC étendu ¹ (Classes-Responsabilités-Collaboration) ou, préférablement, diagramme de classe UML (<i>Unified Modeling Language</i> , voir l’annexe sur le sujet) ou diagramme de base de données (c’est l’enseignant qui décide selon la nature du projet);	8.0 %
c. conception des interfaces usagers (voir l’annexe sur le sujet).	5.0 %
	20.0 %

¹ Au diagramme CRC présenté dans les cours précédents, ajouter une description précise des types de données pour chaque attribut (variables) et la signature complète des méthodes (fonctions) en plus d’inclure une légende claire.

Concernant le document à remettre, voici les contraintes à respecter :

- Le document doit être de format Microsoft Word (*.docx)
- Concernant la production des schémas :
 - pour les cas d'usage et le diagramme CRC ou UML, vous devez utiliser un logiciel de dessin vectoriel de type DIA (logiciel libre) ou Visio (disponible sur les ordinateurs du département);
 - pour les interfaces usagers, vous devez utiliser Balsamiq Mockups ou un autre outil du même calibre (voir l'annexe);
 - tous les schémas doivent être **inclus** dans le document principal et **ne doivent pas** être remis séparément.
- Le document doit être nommé : B65_S1_Conception_VotreNom.docx.

LIVRABLE 2 — DOCUMENT DE PLANIFICATION GÉNÉRALE

Ce document vise la documentation de la planification détaillée de chacun des *Sprints*. Cette planification doit présenter la liste des tâches principales nécessaires à la réalisation de chacune des étapes associées. Ces tâches doivent viser un découpage modulaire du projet de façon telle à ce que chaque module amène une fonctionnalité particulière. Pour chacune de ces tâches, vous devez indiquer :

Une description technique de haut niveau de la tâche	1.0 %
<i>vous pouvez faire référence à certains éléments de votre document de conception et vous devez ajouter les précisions techniques nécessaires (avec Excel, utilisez l'outil Nouveau commentaire disponible dans l'onglet Révision)</i>	
Une liste des tâches préalables à la tâche courante (tâches prédécesseurs);	0.2 %
Une cote de priorité	0.2 %
<i>1 à 3 où 1 étant le plus prioritaire – cette cote doit vous permettre d'établir quels éléments seront évincés si vous manquez de temps</i>	
Une cote associée à la difficulté technique de réalisation	0.2 %
<i>1 à 3 où 1 représente une plus grande difficulté pour vous</i>	
Une estimation du temps requis	0.4 %
	2.0 %

Un fichier Excel a été monté afin de simplifier la gestion de cette planification. L'usage de ce fichier est obligatoire pour tous. Votre document doit être nommé : B65_S1_Planification_VotreNom.xlsx.

LIVRABLE 3, 4 ET 5 — DOCUMENTS DE SUIVI DE PROJET POUR LES *SPRINTS* 2, 3 ET 4 (3 LIVRABLES)

À la fin des *Sprint* 2, 3 et 4, il est important de faire le bilan des activités et de produire un petit rapport d'avancement. Ce document est plutôt simple et reprend en grande partie les éléments qui ont été définis dans le document de planification. D'ailleurs, le fichier Excel mis à la disposition des étudiants contient les éléments nécessaires à la documentation du suivi. Pour chaque tâche, on doit retrouver :

Le temps investi pour chaque activité	0.5 %
L'avancement en % de chaque activité	0.5 %
Si une tâche n'est pas terminée, expliquer quelles ont été les raisons du retard ainsi que les moyens qui seront pris pour la terminer	0.5 %
Pour chaque livrable	1.5 %
Totalisant pour les 3 livrables	4.5 %

Pour chaque remise, le même document doit être réutilisé et mis à jour. Le document doit être nommé : `B65_S (2,3ou4)_Suivi_VotreNom.xlsx`.

LIVRABLES 6 ET 7 — ÉVALUATION DE L'AVANCEMENT DU PROJET (2 LIVRABLES)

Pour les *Sprints* 2 et 3, on doit démontrer l'avancement intermédiaire du projet. La démonstration de ces avancements se fait par une évaluation personnelle avec l'enseignant.

En cours de session, un calendrier sera établi et chaque étudiant aura une période de rencontre avec l'enseignant.

Ces évaluations sont courtes et visent une estimation de l'avancement des projets. Chaque étudiant est évalué sur l'avancement, le suivi fait et la compréhension de son projet et non sur la qualité du code ou tous autres éléments techniques tels que les éléments d'apparence graphique. Ces évaluations sont d'environ 10 minutes seulement et se font seul à seul.

De façon plus précise, la forme de l'évaluation se fait en trois parties :

1. une démonstration du travail réalisé par l'étudiant;
2. une discussion sur l'avancement du projet en consultant le document de suivi;
3. de deux à trois questions techniques sur les réalisations faites ou sur le travail à venir.

LIVRABLE 8 — LIVRAISON FINALE DU PROJET

Au dernier *Sprint*, l'étudiant doit remettre un fichier *zip* contenant tous les éléments techniques du projet:

- Un dossier contenant votre projet (le code source et tous les fichiers de projet);
- Un dossier contenant toutes les bibliothèques externes que vous avez utilisées;
- Un dossier contenant un exécutable fonctionnel de votre application (lorsque applicable);
- Dans la racine de votre fichier *zip*, ajouter le fichier `Lisez_moi.txt`.

Cette évaluation technique est la plus importante et la plus significative de la session. Chaque projet est évalué selon des critères généraux et des critères spécifiques pour chaque projet. Les critères spécifiques sont décrits dans chaque devis. Néanmoins, les critères communs qui s'ajoutent sont :

- qualité du code :
 - modularité;
 - réutilisabilité;
 - lisibilité;
 - documentation;
- Respect des contraintes générales énoncées préalablement.

Le fichier doit se nommer : `B65_S4_Projet_VotreNom.ZIP`.

LIVRABLE 9 — MANUEL DE L'USAGER

Chaque projet doit être accompagné d'un petit document d'utilisation. Ce texte vise à clarifier une utilisation simple de toutes les fonctions disponibles par un usager non technique. Ce que doit contenir ce manuel :

Présentation générale de l'application	0.5 %
Présentation des fonctionnalités et caractéristiques de l'application	1.5 %
<i>À ces listes, on doit préciser les limitations et contraintes s'il y en a</i>	
Présentation d'un cas d'utilisation simple	1.0 %
Présentation des interfaces usagers	1.0 %
<i>Ne présenter que les principaux s'il y en a plusieurs</i>	
	4.0 %

Ce document doit être nommé : `B65_S4_Manuel_VotreNom.docx`

LIVRABLE 10 – FICHIER *LISEZ_MOI.TXT*

Le fichier `Lisez_moi.txt` doit expliquer simplement et clairement la procédure complète d'installation du projet selon deux points de vue :

- utilisateur technique désirant installer tous les éléments du projet pour son développement;
- utilisateur non technique désirant installer uniquement ce qui est nécessaire à l'utilisation de l'application.

La procédure d'installation décrite doit être exhaustive tel que destinée à une personne qui a très peu de connaissance informatique.

Il est important de remettre un fichier `Lisez_moi.txt` clair et complet car sans ce fichier le travail ne sera pas corrigé et une note de zéro sera automatiquement attribuée pour la remise finale.

Ce fichier vaut 1.5 % de la session et doit être nommé : `B65_S4_LisezMoi_VotreNom.docx`

LIVRABLE 11 – SITE WEB

Chaque étudiant doit monter un petit site web, accessible directement à partir de sa page personnelle du CVM. Ce site doit servir de plateforme promotionnelle du projet. Sans se lancer dans une élaboration trop ardue, ce site est évalué sur la pertinence des informations qui y seront disponibles et sur la présentation de ces dernières.

En arrivant sur ce site, on doit y retrouver ces informations :

Présentation sommaire du projet	0.25 %
Dans un but de promotion, quelques captures d'écran avec explications (au moins 3)	0.75 %
Présentation de tous les outils technologiques utilisés pour le développement	0.50 %
<i>IDE, langages de programmation, librairies, moteur de base de données, etc.</i>	
Temps total investi (réaliste)	0.25 %
Les coordonnées de l'étudiant	0.25 %
La présentation générale du site sera aussi évaluée	1.00 %
	3.0 %

Sans être obligatoire, on peut rendre disponible l'exécutable de son projet. Dans ce cas, il importe de joindre à son projet, les deux fichiers de documentation (*Manuel de l'utilisateur* et `Lisez_moi.txt`).

Finalement, la remise se fait par un fichier compressé contenant tous les éléments du site Web. Ce fichier doit être nommé : B65_S4_SiteWeb_VotreNom.ZIP.

LIVRABLE 12 – PRÉSENTATION

Une présentation du projet est à faire à la fin de session. La forme n'étant pas encore déterminée, des précisions seront apportées en cours de session.

Cette présentation vaut pour 5 % de la session.

LIVRABLE 13 – VIDÉO DE PRÉSENTATION

Chaque étudiant doit réaliser une courte vidéo destiné à la présentation de son projet. La vidéo attendue est simplement une capture d'écran dynamique pendant la réalisation de tâches spécifiques de son application. Ainsi, vous devez utiliser un outil comme *SnagIt* qui permet de réaliser de telle vidéo. Les contraintes de réalisation sont :

- vidéo de 1.5 à 2.5 minutes totalisant **moins de 25 Mo**;
- la vidéo doit présenter de façon efficace et claire un ou plusieurs cas d'usage pertinent du projet;
- il n'est aucunement obligatoire de couvrir tous les aspects de l'application mais plutôt l'essentiel – une vidéo trop longue et trop détaillée n'est pas vraiment intéressante et passerait à côté de l'objectif;
- la vidéo doit être accompagnée d'information contextuelle selon l'un de ces deux choix :
 - idéalement, entendre l'étudiant présenter chaque aspect au moment où la vidéo se déroule;
 - voir, en « overlay » des textes explicatifs de ce qui est présenté.

Cette vidéo compte pour 5 % et le fichier de remise doit être nommé : B65_S4_Video_VotreNom.MPG.

LES PROJETS PROPOSÉS

Tous les projets proposés sont construits de façon à suggérer une problématique particulière et certaines contraintes de réalisation tout en laissant une place à l'interprétation et à la créativité de l'étudiant. Cependant, on peut regrouper ces projets en trois formes différentes qui présentent des thèmes plus ou moins précis et, par conséquent, laissent place à une liberté variée. Ces trois formes sont :

- i. **Les projets génériques** : on parle ici de projet sans application concrète prédéfinie. Les devis sont très larges et donnent une place importante à la créativité de l'étudiant.
- ii. **Les projets spécifiques** : correspondant à des mandats plus concrets dont les devis sont plus précis et mieux définies. Ces projets, plus contraignants sur les objectifs, offrent tout de même une bonne place à la créativité de l'étudiant.
- iii. **Les projets industriels** : sont de vrais projets de l'industrie accompagnés de devis précis et d'objectifs concrets. Ces projets laissent aussi place à la créativité des étudiants mais amènent plus de contraintes techniques particulières. Finalement, des opportunités spécifiques s'offrent selon les projets.

PROJET GÉNÉRIQUE 1 – GESTION DE DONNÉES

PRÉSENTATION

La gestion de données est probablement le champ d'application le plus vaste du monde informatique. Quel que soit le domaine applicatif, l'importance d'une manipulation adéquate des données est cruciale et le succès du produit développé en dépend.

Considérant tous les aspects associés à la gestion de données, la complexité des systèmes modernes s'y rattachant peut rapidement devenir impressionnante. On parle souvent des modules de :

- acquisition;
- transformations diverses telles que : conversion, conformité, standardisation, normalisation et autres;
- transport de l'information;
- stockage, archivage;
- base de données;
- manipulations diverses;
- synthétisation;
- interprétation;
- présentation;
- ...

Ce projet vous propose de mettre de l'avant une application de votre choix basée sur la gestion des données. Cette application doit être centrée sur les données qu'elle manipule et un usage pertinent que l'utilisateur peut en faire. Vous pouvez considérer des domaines aussi variés que :

- la gestion de données personnelles :
 - gestion de fichiers personnels;
 - gestion de musiques ou de média;
 - ...
- la gestion d'entreprise :
 - gestion de feuille de temps;
 - gestion d'horaire;
 - gestion de projet;
 - gestion de ressources diverses;
 - ...
- la gestion de données financières :
 - données financières personnelles;
 - investissements divers;
 - ...
- la gestion de capteurs d'observation :
 - données industrielles;
 - données d'infrastructure;
 - ...

CONTRAINTES

La description de ce projet est très générale. Néanmoins, le projet qui doit être réalisé doit être très précis lui. En d'autre mot, il est crucial que le projet soit bien cerné avant d'en commencer sa conception et son développement. D'ailleurs, une validation du concept ou de l'idée doit être validée avec l'enseignant.

Le contexte de gestion de données présenté fait référence à une application faisant une manipulation des données et non pas seulement à une application de stockage des données. Le projet à réaliser doit inclure plusieurs aspects de la gestion de données tout en considérant un logiciel applicatif complet.

Plusieurs contraintes significatives sont à considérer pour ce projet :

- le projet doit être pertinent et présenter un intérêt significatif;
- la complexité du projet doit être à la hauteur des attentes du cours (sans être trop simple, il ne doit pas être trop ambitieux non plus);
- le projet mis de l'avant doit rester respectueux, éthique et exempt d'élément de mauvais goût;
- le projet doit être centré sur les données qu'il manipule et les besoins de l'utilisateur;
- l'application doit inclure tous les modules suivants :
 - acquisition de données pertinente et élaborée (soit par un module d'acquisition externe ou par une interface usager);
 - un module d'édition et de suppression des données;
 - stockage des données par :
 - une structure de données personnalisée pertinente;
 - base de données;
 - un module d'exportation/importation des données selon un format simple;
 - module d'interprétation des données (rapport, synthèse d'information, ...) permettant une visualisation de ces dernières;

PROJET GÉNÉRIQUE 2 – JEU VIDÉO

PRÉSENTATION

Même si le développement d'un jeu vidéo peut sembler simpliste et ludique, les jeux les plus impressionnants du marché sont d'une complexité rarement égalee dans d'autres projets informatiques.

Ce projet vous propose de développer le jeu vidéo de votre choix. Vous pouvez considérer un petit jeu vidéo tel que :

- jeu d'arcade;
- jeu d'action;
- jeu de stratégie;
- jeu de rôle (*RPG – Role Playing Game*);
- jeu de société;
- ...

Le jeu développé peut être à tour de rôle ou en temps réel.

CONTRAINTES

La description de ce projet est très générale mais, en contre partie, le projet qui doit être mis de l'avant doit être très précis. En d'autre mot, il est crucial que le projet soit bien cerné avant d'en commencer sa conception et son développement. D'ailleurs, une validation du concept ou de l'idée doit être validée avec l'enseignant.

Plusieurs contraintes significatives sont à considérer pour ce projet :

- le jeu doit être ludique et projet doit être pertinent et présenter un intérêt significatif;
- la complexité du projet doit être à la hauteur des attentes du cours (sans être trop simple, il ne doit pas non plus être trop ambitieux);
- le projet mis de l'avant doit rester respectueux, éthique et exempt d'élément de mauvais goût;
- le scénario du jeu doit être suffisamment élaboré et présenter une certaine originalité;
- le jeu doit inclure tous les modules suivants :
 - un « *splash screen* »;
 - une fenêtre d'ajustement des paramètres du jeu;
 - présenter au moins trois éléments de base pouvant interagir les uns avec les autres d'une façon ou d'une autre;
 - une fenêtre principale présentant une grande surface de jeu et tous les contrôles nécessaires;
 - un système de pointage pertinent;
 - l'intégration d'au moins un algorithme particulier;
 - ...

PROJETS SPÉCIFIQUES

Plusieurs projets différents ont été présentés aux sessions précédentes. Tous ces projets peuvent maintenant servir de base à titre de projet spécifique. Cependant, tous ces projets peuvent être considérés comme étant des particularités des projets génériques déjà présenté. Néanmoins, trois se démarquent par leur aspect plus particulier :

- simulation d'un écosystème;
- traitement d'image;
- vision artificielle.

Le document de la session précédente est mis à votre disposition à titre de référence sur ces devis. Les pages 12 à 100 sont les plus pertinentes.

PROJETS INDUSTRIELS

Cette session, un seul projet industriel est proposé :

1. Développement d'un outil de d'étiquetage – Collineo

QU'EST-CE QUE UML

L'acronyme UML vient de « *Unified Modeling Language* » qui veut dire langage de modélisation unifié. Ce langage a été créé afin de normaliser les nombreux outils de conception de projet existants auparavant. Il consiste à créer des graphiques à base de pictogrammes illustrant différents éléments conceptuels d'un système. Initialement développé par des chercheurs du domaine du génie logiciel et destiné aux éléments de conception orientée objet, cette méthode de conception est maintenant utilisée par plusieurs branches du génie (électrique, mécanique, industriel, etc).

La dernière révision d'UML (datant de mai 2010) propose 13 types de diagrammes permettant d'élaborer la conception d'un projet. Ces 13 diagrammes sont dépendants hiérarchiquement et se complètent mutuellement de façon à permettre la modélisation d'un projet tout au long de son cycle de vie (que ce soit de sa conception, sa réalisation, son implémentation, son déploiement et même sa maintenance). À titre de référence, voici la liste des 13 diagrammes :

- Les 6 diagrammes structurels (statique)
 - Diagramme de classes
 - Diagramme d'objets
 - Diagramme de composants
 - Diagramme de déploiement
 - Diagramme des paquetages
 - Diagramme de structure composite
- Les 3 diagrammes comportementaux
 - Diagramme des cas d'utilisation
 - Diagramme état-transition
 - Diagramme d'activité
- Les 4 diagrammes d'interaction (dynamique)
 - Diagramme de séquence
 - Diagramme de communication
 - Diagramme global d'interaction
 - Diagramme de temps

Il est important de comprendre que l'utilisation du langage UML n'est pas une méthode. Ce qui implique que son utilisation est laissée au besoin de son utilisateur. Par exemple, plusieurs méthodologies utilisent partiellement le langage. D'ailleurs, il est courant de retrouver en industrie seulement les diagrammes les plus utilisés : le diagramme des cas d'utilisation et le diagramme de classes notamment.

Dans le cadre du projet, vous n'aurez qu'à fournir le diagramme de classes.

DIAGRAMME DE CLASSES

Le diagramme de classes est un schéma utilisé pour présenter les classes et les interfaces des systèmes ainsi que les différentes relations entre celles-ci. Une classe, comme vous le savez certainement, décrit les responsabilités, les attributs et le comportement d'un ensemble d'objet de même type. De cette

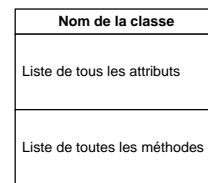
façon, elles permettent de modéliser un programme pour ainsi découper une tâche complexe en plusieurs sous-tâches plus simples.

Les classes peuvent être liées entre elles grâce au mécanisme d'héritage qui permet de mettre en évidence les relations de parenté. Plusieurs autres types de relations sont possibles entre des classes. Chacune de ces relations est représentée par un pictogramme spécifique dans le diagramme de classes.

Dans le cadre du cours, nous n'aborderons qu'une partie de ce qu'offre le diagramme de classes. Nous mettrons l'accent sur trois types de relations simplifiées : l'héritage, l'association et l'agrégation.

LA CLASSE

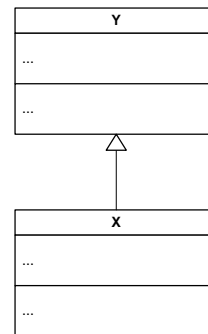
La classe est l'élément de base et est représentée par un pictogramme rectangulaire divisé en trois parties distinctes. La première partie possède le nom de la classe, la deuxième partie possède la liste de tous les attributs (incluant les types) et la troisième partie possède la liste de toutes les méthodes (incluant la signature détaillée de la fonction). Habituellement, on ajoute un code pour indiquer si chaque attribut et méthode est privé, protégé ou public.



L'HÉRITAGE

C'est le principe de division par généralisation et spécialisation. D'ailleurs, parfois on nomme le concept d'héritage comme étant le concept de généralisation ou de spécialisation dépendamment du point de vue. Néanmoins, ce type de relation implique toujours une classe parent et une classe enfant pour laquelle, la classe enfant hérite des attributs et méthodes (publics et protégés) de la classe parent.

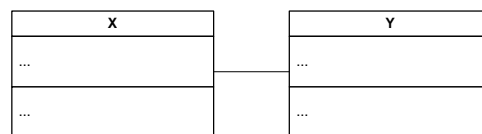
Le pictogramme pour indiquer cette relation est la flèche vide pointant vers le parent. De cette façon, le pictogramme se lit ainsi, la classe enfant X hérite de la classe parent Y.



L'ASSOCIATION

Ce type de relation implique une connexion sémantique entre deux classes. Par exemple, une première classe peut modifier le contenu d'une deuxième classe simplement en passant cette dernière comme paramètre dans une de ses méthodes. Un autre cas, une méthode de la première classe retourne une nouvelle instance d'une deuxième classe.

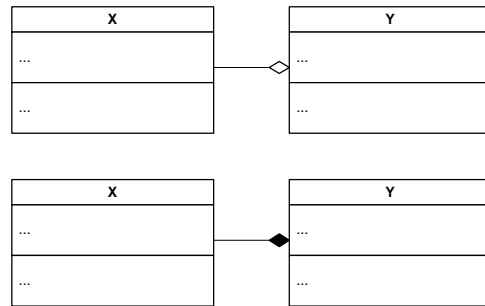
Dans tous ces cas, il y a association entre ces deux classes. Mais ni l'une ni l'autre ne va obligatoirement utiliser ou nécessiter l'autre pour exister. Le pictogramme utilisé ici est simplement une ligne.



L'AGRÉGATION ET LA COMPOSITION

C'est une relation de subordination entre deux classes. On dit qu'une première classe regroupe d'autres classes ou que l'instance de la première classe utilise une instance d'une deuxième classe. On peut spécialiser la notion d'agrégation lorsque le cycle de vie de la deuxième classe dépend de la première classe. On parle alors de composition.

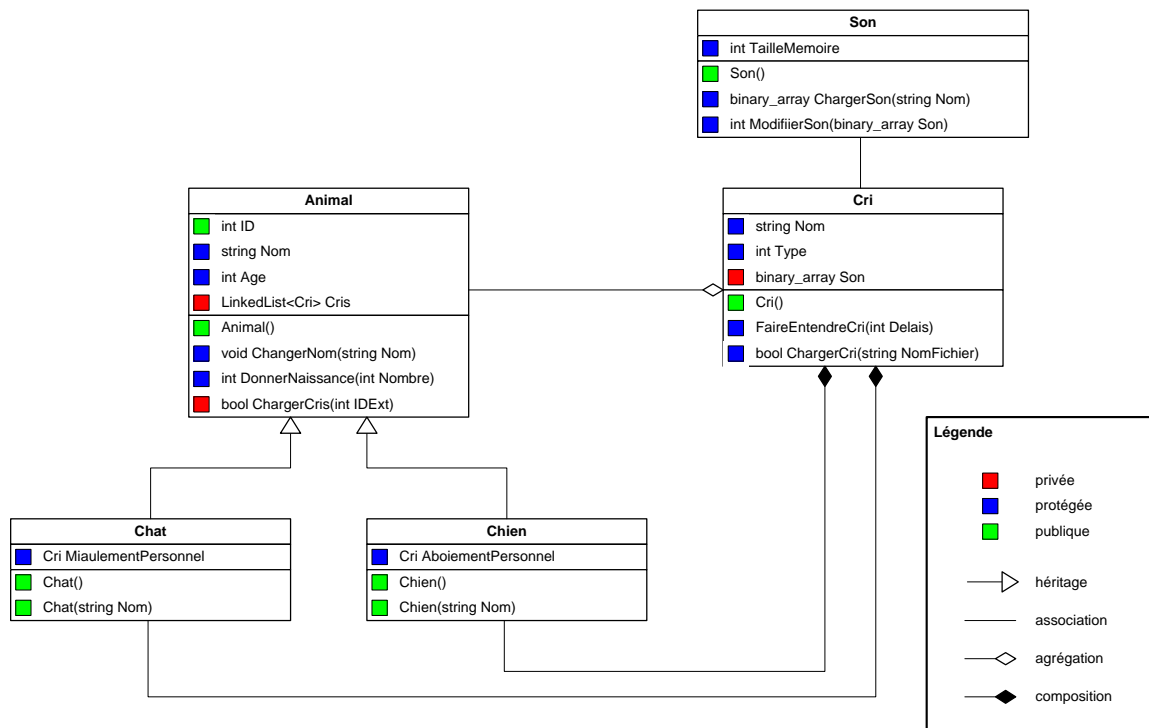
Le pictogramme utilisé est une ligne terminée par un losange vide (pour l'agrégation) ou plein (pour la composition).



EXEMPLE COMPLET

Cet exemple montre une petite conception incomplète de classe reliée à une application faisant l'usage de différents animaux. Cet exemple présente 5 classes distinctes toutes reliées d'une façon ou d'une autre.

Voici le diagramme de classes associé :



Les classes *Chat* et *Chien* hérite de la classe *Animal* de façon évidente.

La composition entre la classe *Cri* et les classes *Chat* et *Chien* vient du fait que chacune des classes possesseurs (*Chat* et *Chien*) possèdent une variable de type *Cri* (*MiaulementPersonnel* et *AboiementPersonnel*). On remarque que l'instanciation d'un objet *Cri* se fera en même temps que l'instanciation de la classe possesseur. Ainsi, la classe *Cri* « naîtra » et « mourra » en même temps que la classe *Chat* par exemple.

L'agrégation entre la classe *Animal* et *Cri* représente le fait que la classe possesseur (*Animal*) peut posséder une ou plusieurs instances de la classe *Cri*. Lorsque la classe *Animal* est instanciée, il est possible qu'aucune classe *Cri* ne soit instanciée. Il est même possible que les objets de la classe *Cri* incluses dans la liste de *Animal* ne soient pas libérés lorsque l'instance de la classe *Animal* soit libérée.

L'association entre la classe *Cri* et *Son* vient du fait que la classe *Cri* utilise dans une de ses fonctions (*ChargerSon*) une classe externe pour faire le travail. C'est-à-dire que la classe *Son* sera utilisée par la classe *Cri* pour faire une tâche spécifique. Néanmoins, aucune instance de la classe *Son* n'est possédée par la classe *Cri*.

Voici maintenant un exemple de code illustrant ces propos :

```
public class Animal
{
    public int ID;
    protected String Nom;
    protected int Age;
    private LinkedList<Cri> Cris = new LinkedList<Cri>();           // Agrégation

    public Animal() {};
    public void ChangerNom(String Nom) { this.Nom = Nom };
    protected int DonnerNaissance(int Nombre) { ... }
    private boolean ChargerCris(int IDExt) { ... }
}

public class Cri
{
    protected String Nom;
    protected int Type;
    private binary array Son;

    public Cri() {};
    protected void FaireEntendreCri(int Delais) { ... };
    protected boolean ChargerCri(String NomFichier) { ... }       // Association
}

public class Son
{
    protected int TailleMemoire;

    public Son() {};
    protected binary_array ChargerSon(String Nom) { ... };
    protected int ModifierSon(binary_array Son) { ... }
}

public class Chat extends Animal                                // Héritage
{
    protected Cri MiaulementPersonnel;                          // Composition

    public Chat() {};
    public Chat(String Nom) { ... };
}

public class Chien extends Animal                              // Héritage
{
    protected Cri AboiementPersonnel;                            // Composition

    public Chat() {};
    public Chat(String Nom) { ... };
}
```

ANNEXE 2 — OUTILS DE CONCEPTION | CONCEPTION D'INTERFACE USAGER

Il est fréquent de voir une grande négligence au niveau de la conception des interfaces usagers lors d'un développement informatique de petite envergure ou pratiqué par des novices. Cependant, cette partie de la conception est fondamentale pour trois aspects :

- permet de valider les besoins du client avec ce dernier;
- aide à mieux définir les besoins applicatifs du projet;
- essentiel à l'ergonomie générale de l'application.

BESOINS APPLICATIFS DU PROJET

Généralement, la conception d'un projet informatique amène une description plus ou moins détaillée des constituants du projet et des différents cas d'usage. Ces cas d'usage, visant systématiquement les tâches de haut niveau, amènent parfois à des lacunes significatives au niveau de l'interaction finale avec l'utilisateur.

Plusieurs méthodes et outils de conception permettent de réduire ces omissions. L'une des méthodes les plus fréquentes, les plus efficaces et les plus simples est la conception des interfaces usagers.

Il est important de comprendre que la conception des interfaces usagers a un impact important dès les premiers instants du processus de conception. Outre l'aspect ergonomie qui est abordé plus loin, cette phase permet de mieux situer le projet dans son ensemble, de définir tous les cas d'usage au niveau de son utilisation et finalement d'avoir une réflexion plus profonde sur la finalité du projet.

Cet effort de réflexion est parfois plus important qu'il n'y paraît et apporte beaucoup au processus de conception.

ERGONOMIE EN INFORMATIQUE

L'ergonomie informatique (ou ergonomie des interfaces) est une branche de l'ergonomie, qui a pour objectif d'améliorer l'interaction homme-machine, la facilité d'utilisation et d'apprentissage des produits interactifs. Cette pratique cherche à concevoir ou modifier des interfaces afin qu'elles soient en adéquation avec les caractéristiques physiologiques, perceptives et cognitives de leurs utilisateurs potentiels. Elle s'appuie sur différentes méthodes de conception et d'évaluation permettant d'obtenir le logiciel ou le site web le mieux adapté aux utilisateurs visés selon l'application spécifique.

Tout au long de la démarche ergonomique, la conception et l'évaluation du produit s'attachent à prendre en compte les besoins et attentes de l'utilisateur final ; elle vise à intégrer la composante humaine, le "point de vue utilisateur", dans le processus de conception d'un produit informatique : la conception centrée sur l'utilisateur.

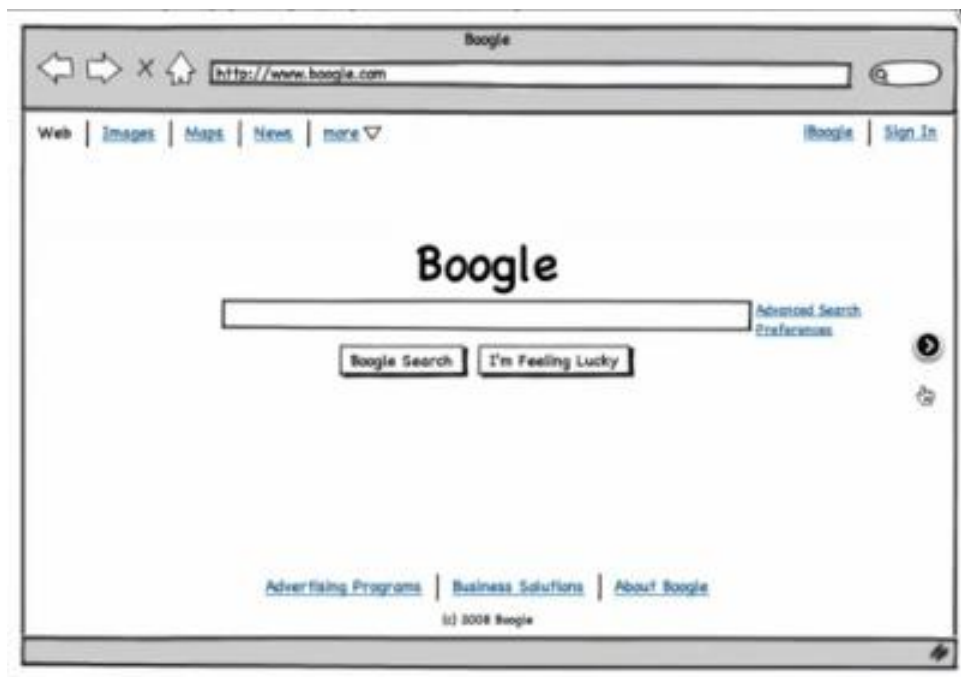
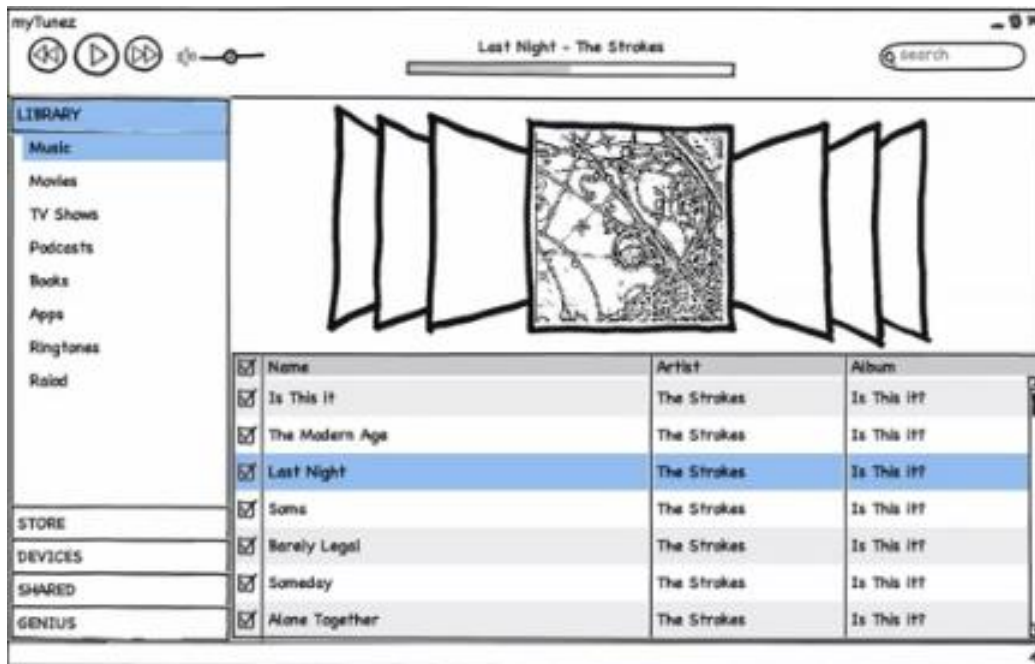
Beaucoup de choses peuvent être dites sur ce sujet car il correspond à une branche significative du génie logiciel et de ses dérivées.

OUTILS RECOMMANDÉS POUR LE COURS

Il existe plusieurs outils de conception d'interface. Des outils plus traditionnels tels que Visio permettent de créer simplement des prototypes efficaces. Néanmoins, il existe depuis quelques années un outil d'une très grande qualité qui est de plus en plus reconnu par l'industrie : *Balsamiq Mockups*.

Ce logiciel est très simple d'utilisation et facile à apprendre. De plus, il existe une version « *Web based* » dont le département possède une licence académique. Pour qu'un étudiant puisse l'utiliser dans le cadre d'un projet particulier, vous devez demander à l'enseignant qu'il vous invite en lui donnant une adresse courriel viable.

Voici quelques exemples de conception avec cet outil :



ANNEXE 3 – OUTILS DE CONCEPTION | « *DESIGN PATTERN* »

À venir...