

## Programmation de jeux vidéo III

420-V31-SF

Pondération 3-3-3

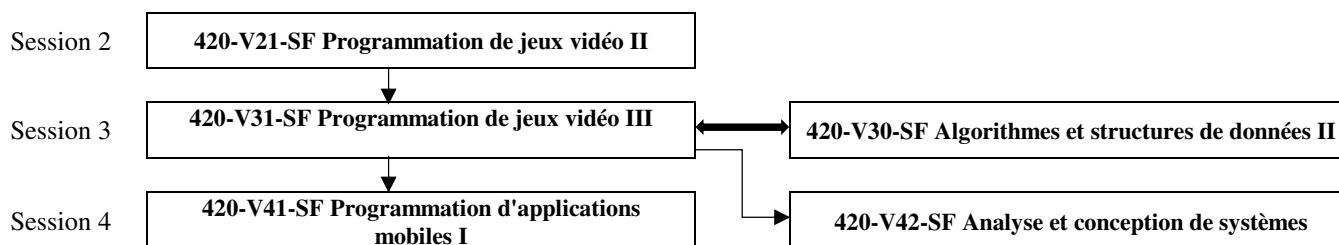
François Paradis [fparadis@cegep-ste-foy.qc.ca](mailto:fparadis@cegep-ste-foy.qc.ca), local C-242 #6661, CP 337  
Pierre Poulin [ppoulin@cegep-ste-foy.qc.ca](mailto:ppoulin@cegep-ste-foy.qc.ca), local C-240 #6669, casier 94

### Présentation générale du cours

Le cours « Programmation de jeux vidéo III » (420-V31-SF) destiné à la clientèle étudiante du programme « Techniques de l'informatique – Programmation de jeux vidéo » permet à l'étudiant d'approfondir ses connaissances en programmation objet. L'étudiant devra créer des diagrammes de classes à partir de situations textuelles simples. Il apprendra ou révisera également les techniques et concepts plus avancés tels l'héritage, le polymorphisme et l'encapsulation de données. L'étudiant devra compléter la programmation d'un jeu vidéo à partir d'une base de code lui étant fournie. Une attention particulière sera portée au découpage et à l'architecture d'une application dans un environnement graphique. Pour ce faire, des patrons de conceptions pertinents seront explorés et utilisés.

L'étudiant doit consolider les techniques reliées au développement de programmes vues dans les cours de la première année. Il doit maintenant élaborer des programmes d'une plus grande envergure.

### Situation du cours dans le programme



Préalable absolu : →

Cours en parallèle : ↔

Ce cours a comme préalable absolu le cours « 420-V21-SF Programmation de jeux vidéo II ».

Il sera donné en parallèle avec le cours « 420-V30-SF Algorithmes et structures de données II » avec laquelle il est arrimé.

Ce cours est un préalable absolu aux cours « 420-V41-SF Programmation d'applications mobiles I » et « 420-V42-SF Analyse et conception de systèmes ».

### Objectifs spécifiques du cours

Le cours vise spécifiquement à développer les éléments de compétences suivants :

Énoncé(s) de la compétence	Éléments de la compétence
016T - Appliquer une approche de développement par objets	1 Créer un modèle objet
	3 Procéder à la codification d'une classe
	4 Valider le fonctionnement d'une classe
	5 Générer la version exécutable du programme
0176 - Apporter des améliorations fonctionnelles à une application	1 Analyser les caractéristiques de l'application
	2 Analyser la nature des améliorations à apporter à l'application
	3 Ajouter et modifier des fonctionnalités à une application
	4 Valider le fonctionnement de l'application
017C - Concevoir et développer une application dans un environnement graphique	1 Établir le cadre général de l'application
	2 Préparer le travail de développement de l'application
	3 Modéliser l'application
	4 Programmer l'application
	5 Produire la documentation relative à l'application

## Stratégies pédagogiques et modalités de participation

Ce cours se déroule sous forme d'exposés magistraux, de démonstrations et d'ateliers en laboratoire pour un total de 90 heures en classe auxquelles s'ajoutent environ 45 heures de travail personnel. Le professeur intervient pour introduire et commenter les concepts et les notions nécessaires à la réalisation de chaque programme. Il agit régulièrement comme un guide accompagnant l'étudiant dans son cheminement.

Des exercices fréquents permettront à l'étudiant d'appliquer les méthodes et techniques enseignées. Les travaux pratiques seront directement en lien avec les jeux vidéo.

On développera l'autonomie et l'auto apprentissage des étudiants en insistant sur l'exploration de l'environnement de travail ainsi que sur l'utilisation des ressources documentaires disponibles.

## Stratégies d'évaluation

### Évaluation formative

Des exercices formatifs devront être réalisés et remis tout au long de la session. Ces exercices viseront principalement à aider la compréhension et l'application de la matière. Notez que ces exercices comptent pour **5% de la pondération du cours** et que l'évaluation de ces derniers portera principalement sur **l'assiduité<sup>1</sup> à la réalisation de ces exercices**.

Nature des évaluations	Pondérations	Dates approximatives	Contextes de réalisation
Trois travaux pratiques notés en cours de session valant chacun 10%, 10% et 25%	45%	Semaines 4, 9 et 15	Seul et/ou en équipe
Trois examens valant respectivement 15%, 20% et 15%	50%	Semaines 5, 10 et 15	Écrits réalisés individuellement

### Note de passage

Pour obtenir la note de passage d'un cours, l'étudiant doit obtenir une note finale minimale de 60%.

## Médiagraphie & environnement de développement

1. Livres de référence
  - [Dickheiser, Mike : C++ For Game Programmers](#)
  - [Nystrom, Robert: Game Programming Patterns](#)
2. Sites internet
  - Références sur le C++ : <http://www.cplusplus.com/>
  - Tutoriaux sur le langage C++
    - <https://openclassrooms.com/courses/programmez-avec-le-langage-c>
    - <http://www.cprogramming.com/tutorial/>
    - <http://www.learncpp.com/>
    - <https://isocpp.org/get-started>
  - Tutoriaux sur SFML
    - <https://openclassrooms.com/courses/creez-des-applications-2d-avec-sfml?status=published>
    - <https://github.com/SFML/SFML/wiki>

---

<sup>1</sup> **Définition :** Application continue à un travail, dans le but d'obtenir un produit fini de qualité.

### Environnement de développement obligatoire

- Microsoft Visual Studio 2015 (Community Edition suffisante)
- SFML 2.4.2 ( Visual C++ 14 (2015) - 32-bit ) : <https://www.sfml-dev.org/download/sfml/2.4.2/index-fr.php>
- Visual Leak Detector: <https://vld.codeplex.com/>
- GhostDoc: <http://submain.com/products/ghostdoc.aspx>
- Cahier de notes, papier et crayon

### Environnement de développement optionnel (Mais recommandé)

- Gamepad USB (Version Microsoft XBox recommandé)

## Calendrier des blocs et des remises

Le calendrier suivant est donné à titre indicatif seulement et peut être modifié en fonction des circonstances. En ce qui concerne la date de l'examen et celles de remise des travaux, il va de soi que les étudiants seront avisés conformément à ce que prévoit la Politique d'évaluation des apprentissages du Collège.

### Bloc 1 – Rappels et récapitulatifs; Introduction à la programmation C++ et à la programmation graphique

Énoncé de la compétence	Éléments de la compétence
016T - Appliquer une approche de développement par objets.	016T-1 Créer un modèle objet.
	016T-3 Procéder à la codification d'une classe.
	016T-4 Valider le fonctionnement d'une classe.
	016T-5 Générer la version exécutable d'un programme.
0176 - Apporter des améliorations fonctionnelles à une application.	0176-1 Analyser les caractéristiques de l'application.
	0176-2 Analyser la nature des améliorations à apporter à l'application.
	0176-3 Ajouter et modifier des fonctionnalités à une application.
	0176-4 Valider le fonctionnement de l'application.
017C - Concevoir et développer une application dans un environnement graphique.	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.

Contexte d'apprentissage	Activités d'apprentissage
<ul style="list-style-type: none"><li>• Individuellement.</li><li>• Exercices formatifs régulièrement</li></ul> <p><u>À partir</u></p> <ul style="list-style-type: none"><li>- de concepts de programmation</li><li>- de cas concret d'utilisation</li></ul> <p><u>À l'aide</u></p> <ul style="list-style-type: none"><li>- d'un ordinateur</li></ul>	<p><b>Contenus :</b> Des formatifs tout au long du bloc 1 Un quiz formatif préparatoire à l'examen Révision des concepts en programmation Caractéristiques du langage C++ UML : Conception partie 1 Documentation de code. Programmation graphique Division du code (Entrées-traitement-affichage) Algorithmes de déplacement 2D Introduction à l'héritage en C++</p> <p><b>Évaluations sommatives :</b> Travail pratique en lien avec les jeux vidéo Examen écrit</p>

## Bloc 2 – Héritage, polymorphisme et patrons de conceptions (partie 1)

Énoncé de la compétence	Éléments de la compétence
016T - Appliquer une approche de développement par objets.	016T-1 Créer un modèle objet.
	016T-3 Procéder à la codification d'une classe.
	016T-4 Valider le fonctionnement d'une classe.
	016T-5 Générer la version exécutable d'un programme.
0176 - Apporter des améliorations fonctionnelles à une application.	0176-1 Analyser les caractéristiques de l'application.
	0176-2 Analyser la nature des améliorations à apporter à l'application.
	0176-3 Ajouter et modifier des fonctionnalités à une application.
	0176-4 Valider le fonctionnement de l'application.
017C - Concevoir et développer une application dans un environnement graphique.	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.

Contexte d'apprentissage	Activités d'apprentissage
<ul style="list-style-type: none"> <li>• Individuellement.</li> <li>• Exercices formatifs régulièrement</li> </ul> <p><u>À partir</u></p> <ul style="list-style-type: none"> <li>- de concepts de programmation</li> <li>- de cas concret d'utilisation</li> </ul> <p><u>À l'aide</u></p> <ul style="list-style-type: none"> <li>- d'un ordinateur</li> </ul>	<p><b>Contenus :</b> Des formatifs tout au long du bloc 2  Un quiz formatif préparatoire à l'examen  Héritage (suite)  Polymorphisme et transtypage  Méthodes virtuelles  Gestion de source  UML : Conception partie 2  Introduction au patron de conception : Singleton  Patron de conception : Commande  Patron de conception : MVC</p> <p><b>Évaluations sommatives :</b>  Travail pratique en lien avec les jeux vidéo  Examen écrit</p>

### Bloc 3 – Héritage multiple, interfaces et patrons de conception (partie 2)

Énoncé de la compétence	Éléments de la compétence
016T - Appliquer une approche de développement par objets.	016T-1 Créer un modèle objet.
	016T-3 Procéder à la codification d'une classe.
	016T-4 Valider le fonctionnement d'une classe.
	016T-5 Générer la version exécutable d'un programme.
0176 - Apporter des améliorations fonctionnelles à une application.	0176-1 Analyser les caractéristiques de l'application.
	0176-2 Analyser la nature des améliorations à apporter à l'application.
	0176-3 Ajouter et modifier des fonctionnalités à une application.
	0176-4 Valider le fonctionnement de l'application.
017C - Concevoir et développer une application dans un environnement graphique.	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.

Contexte d'apprentissage	Activités d'apprentissage
<ul style="list-style-type: none"> <li>• Individuellement.</li> <li>• Exercices formatifs régulièrement</li> </ul> <p><u>À partir</u></p> <ul style="list-style-type: none"> <li>- de concepts de programmation</li> <li>- de cas concret d'utilisation</li> </ul> <p><u>À l'aide</u></p> <ul style="list-style-type: none"> <li>- d'un ordinateur</li> </ul>	<p><b>Contenus :</b> Des formatifs tout au long du bloc 3 Un quiz formatif préparatoire à l'examen Héritage multiple et interfaces Patron de conception : Observateur Patron de conception : Fabrique Patron de conception : Composite Librairies</p> <p><b>Évaluations sommatives :</b> Travail pratique en lien avec les jeux vidéo (Travail commun avec le cours 420-V30-SF) Algorithmes et structures de données II) Examen écrit</p>

## Compétences détaillées

Critères de performance		Habilités	Contenus
016T-1	<ul style="list-style-type: none"> <li>1 - Établissement complet des domaines d'application.</li> <li>2 - Détermination des classes pertinentes.</li> <li>4 - Détermination des services de chacune des classes.</li> <li>5 - Détermination des attributs pertinents compte tenu de chacune des classes.</li> <li>6 - Représentation graphique appropriée du modèle objet.</li> </ul>	<ul style="list-style-type: none"> <li>1.1 1.2 Décomposer un problème sous la forme de classes.</li> <li>1.3 Concevoir les relations entre les classes.</li> <li>1.4 Déterminer les méthodes des classes.</li> <li>1.5 Déterminer les propriétés des classes.</li> </ul>	<ul style="list-style-type: none"> <li>Classes</li> <li>Types de relations composition vs agrégation, utilisation</li> <li>Méthodes des classes</li> <li>Propriétés des classes</li> </ul>
016T-3	<ul style="list-style-type: none"> <li>1 - Préparation appropriée de l'environnement de programmation.</li> <li>2 - Déclaration et définition de la classe respectant les règles de syntaxe et de sémantique du langage.</li> <li>3 - Exploitation judicieuse des possibilités du langage dans l'application des principes d'encapsulation, d'héritage et de polymorphisme.</li> <li>4 - Application rigoureuse des standards de nomenclature et de documentation reconnus propres à l'entreprise.</li> <li>5 - Repérage et correction des erreurs de compilation de la classe.</li> </ul>	<ul style="list-style-type: none"> <li>3.1 Configurer et utiliser un environnement de développement.</li> <li>3.2 Programmer chacune des classes en respectant les particularités du langage choisi.</li> <li>3.3 Appliquer les principes d'encapsulation en respectant les particularités du langage choisi.</li> <li>3.4 Respecter les standards de nomenclature et de documentation prescrits.</li> <li>3.5 Générer le programme et interpréter les messages fournis par l'environnement de développement afin de corriger les erreurs de génération.</li> </ul>	<ul style="list-style-type: none"> <li>Syntaxe et particularités du langage de programmation</li> <li>Principe d'encapsulation</li> <li>Accessibilité des propriétés et des méthodes</li> <li>Normes de programmation objets (par exemple, une classe par fichier et nomenclature des fichiers)</li> <li>Types d'erreurs de génération</li> </ul>
016T-4	<ul style="list-style-type: none"> <li>1 - Préparation correcte des jeux d'essai nécessaires à la vérification du bon fonctionnement de tous les services de la classe.</li> <li>2 - Création d'un environnement de test approprié.</li> <li>3 - Interprétation juste des résultats.</li> <li>4 - Débogage approprié de la classe.</li> </ul>	<ul style="list-style-type: none"> <li>4.1 Déterminer les méthodes à tester.</li> <li>4.2 Concevoir et programmer les tests appropriés pour chacune de ces méthodes.</li> <li>4.3 4.4 Choisir et utiliser les fonctionnalités les plus pertinentes de l'environnement de développement pour repérer et corriger les erreurs.</li> </ul>	<ul style="list-style-type: none"> <li>Tests unitaires</li> <li>Principales fonctionnalités du débogueur (par exemple, types de points d'arrêt, pile des appels, fenêtre des espions)</li> </ul>
016T-5	<ul style="list-style-type: none"> <li>1 - Utilisation des mécanismes de génération propres à l'outil employé.</li> </ul>	<ul style="list-style-type: none"> <li>5.1 Utiliser un outil logiciel pour générer le fichier exécutable.</li> </ul>	

0176-1	<p>1 - Examen complet de tous les documents disponibles sur l'application.</p> <p>2 - Examen des fonctionnalités de l'application et de leurs relations.</p> <p>3 - Examen des caractéristiques de l'application au regard de la programmation et des données.</p> <p>4 - Examen complet des exigences de l'entreprise.</p>	<p>Afin de comprendre le fonctionnement interne de l'application et les choix technologiques et conceptuels sous-jacents, examiner, en consultation avec les gens impliqués :</p> <p>la documentation externe et dans le code; l'organisation des données; l'architecture du code; les tests effectués; les fonctionnalités de l'application, notamment en se substituant au rôle de l'utilisateur.</p>	<p>Exploiter un langage de programmation structurée</p> <p>Appliquer une approche de développement par objets</p> <p>Développer des modèles conceptuels selon l'approche structurée</p> <p>Créer et exploiter des bases de données</p>
0176-2	<p>1 - Détermination des fonctionnalités à ajouter à l'application.</p> <p>2 - Détermination des modifications à apporter aux fonctionnalités existantes.</p> <p>3 - Vérification de la faisabilité des changements dans l'environnement de l'application.</p> <p>4 - Mise en évidence des répercussions des changements sur les programmes et les données.</p>	<p>2.1 2.2 Déterminer les fonctionnalités à ajouter à l'application et les modifications à apporter aux fonctionnalités existantes.</p> <p>2.3 Déterminer la faisabilité des modifications à apporter.</p> <p>2.4 Évaluer l'impact des changements sur l'organisation des données et l'utilisation de l'application.</p>	<p>Réinvestissement des compétences de programmation, d'analyse et de la compétence 016V (élément 4) : Appliquer une approche client</p>
0176-3	<p>1 - Planification réaliste des activités clés.</p> <p>2 - Développement et modification des algorithmes appropriés.</p> <p>3 - Conception et modification correcte des structures de données appropriées.</p> <p>4 - Mise en place correcte des structures de données.</p> <p>5 - Conversion appropriée des données existantes.</p> <p>6 - Production et modification correcte des interfaces graphiques appropriées.</p> <p>7 - Programmation appropriée des nouvelles fonctionnalités.</p> <p>8 - Modification appropriée des programmes touchés par les changements.</p> <p>9 - Exploitation judicieuse des ressources existantes.</p>	<p>Planifier les activités de développement et en faire le suivi.</p> <p>Utiliser les techniques de développement et de programmation pour apporter les améliorations souhaitées à l'application, en s'assurant de préserver l'intégrité des données existantes.</p>	<p>Outils de gestion de sources</p> <p>Planification et suivi des activités de développement (par exemple, logiciel « Pivotal Tracker »)</p>

0176-4	<p>1 - Vérification rigoureuse des programmes portant sur les nouvelles fonctionnalités et de leur intégration dans l'application.</p> <p>2 - Vérification rigoureuse des programmes modifiés et de leur bon fonctionnement dans l'application.</p> <p>3 - Mise à jour soignée et complète de la documentation associée à l'application.</p> <p>4 - Présentation des propositions pour approbation en cours d'exécution et à la fin des travaux.</p>	<p>Utiliser les techniques pour valider le fonctionnement de l'application avec les améliorations apportées, notamment à l'aide de tests.</p> <p>Développer l'habitude d'obtenir l'approbation des modifications de la part des utilisateurs.</p> <p>Mettre à jour la documentation.</p>	<p>Tests unitaires et d'intégration</p> <p>Assurer la qualité d'une application</p>
017C-1	<p>1 - Précision de l'idée directrice en collaboration avec les personnes participant au projet.</p> <p>2 - Recherche et analyse des produits similaires sur le marché.</p> <p>3 - Détermination des caractéristiques fonctionnelles de l'application.</p> <p>4 - Établissement des exigences matérielles propres à l'application.</p> <p>5 - Examen des contraintes imposées par l'environnement graphique et par les outils de développement.</p> <p>6 - Représentation de l'interface de l'application conforme aux exigences de l'ergonomie et de l'esthétisme.</p> <p>7 - Présentation du projet pour approbation.</p> <p>8 - Communication efficace avec toutes les personnes participant au projet.</p>	<p>Établir les fonctionnalités de l'application.</p> <p>Identifier et tenir compte des particularités de la plateforme cible (par exemple, grosseur et résolution de l'écran, puissance de calcul, quantité de mémoire).</p>	<p>Établir les fonctionnalités de l'application</p> <p>Recherche des particularités des plateformes</p>
017C-2	<p>1 - Établissement d'un calendrier des activités réaliste.</p> <p>2 - Utilisation appropriée des méthodes et des outils de planification.</p> <p>3 - Appropriation de l'environnement de développement de l'application.</p>	<p>Préparer le travail de développement de l'application.</p>	<p>Environnements de développement : langages et technologies utilisées, etc.</p> <p>Planifier et gérer des activités de travail</p> <p>Outils d'aide à la gestion de projet</p> <p>Diagrammes de planification (ex. Gantt)</p>
017C-3	<p>1 - Application appropriée d'une méthode d'analyse.</p> <p>2 - Production de modèles justes en fonction de la méthode d'analyse retenue.</p> <p>3 - Production des documents techniques appropriés.</p>	<p>Développer des modèles conceptuels selon l'approche structurée.</p> <p>Modéliser l'application selon les méthodes propres à l'environnement graphique.</p>	<p>Planifier et gérer des activités de travail.</p> <p>Patrons de conception (MVC, MVVM et autres)</p>



0174C-4	<p>1 - Utilisation appropriée des bibliothèques.</p> <p>2 - Codification appropriée des éléments du modèle.</p> <p>3 - Utilisation appropriée des outils de création des éléments d'interface.</p> <p>4 - Validation correcte du fonctionnement de l'application.</p> <p>5 - Production complète et archivage de toute l'information relative aux programmes.</p>	Développer les programmes.	Développer les programmes.
017C-5	<p>1 - Modification appropriée de toute l'information relative à l'application.</p> <p>2 - Création appropriée de l'aide en ligne.</p> <p>3 - Rédaction claire et complète des instructions d'utilisation de l'application.</p>	Produire la documentation relative à l'application.	Produire la documentation relative à l'application.

## Calendrier détaillé

Semaines	Heures consacrées	Activités pédagogiques	Évaluations	Blocs du cours
1 21/08/2017	6	Mise en place du cours 420-V31-SF Installations et mise en place de l'environnement. C++, Le langage, Historique, particularités Retour et révisions de concepts connus sous l'angle C++		Bloc 1
2 28/08/2017	6	Documentation du code Introduction à SFML en C++ Séparation du code en programmation graphique Trigonométrie et algorithmes de déplacement 2D Algorithmes de collision		
3 04/09/2017	6	Éléments de C++, partie 1 Introduction à l'héritage en C++ UML, Conception et exercices, partie 1 Présentation du TP1		
4 11/09/2017	6	<b>Travail pratique #1</b>	<b>Sommative 10%</b>	
5 18/09/2017	1	Révision pour l'examen #1	Formative	
	2	Travail pratique #1		
	3	<b>Examen #1</b>	<b>Sommative 15%</b>	
6 25/09/2017	6	Héritage : concepts avancés et polymorphisme Héritage et design : séparation des concepts. Gestion des erreurs et tests unitaires		Bloc 2
7 02/10/2017	6	Patrons de conception : Singleton, Commande et MVC Interfaces : base sur les machines à états		
09/10/2017	Semaine de relâche			
8 16/10/2017	6	Éléments de C++, partie 2. Introduction à l'animation 2D UML, Conception et exercices, partie 2 Introduction à la gestion de source Présentation du TP2		Bloc 2
9 23/10/2017	6	<b>Travail pratique #2</b>	<b>Sommative 10%</b>	
10 30/10/2017	2	Travail pratique #2		
	1	Révision pour l'examen # 2	Formative	
	3	<b>Examen #2</b>	<b>Sommative 20%</b>	

Semaines	Heures consacrées	Activités pédagogiques	Évaluations	Blocs du cours
11 06/11/2017	6	Héritage multiple Patrons de conception: observateur		Bloc 3
12 13/11/2017	6	Patrons de conception: composite et fabriques Librairies Outils et techniques de débogage Présentation du TP3		
13 20/11/2017	14	<b>Travail pratique #3</b>	<b>Sommative 25%</b>	
14 27/11/2017				
15 04/12/2017				
	1	Révision pour l'examen final	Formative	
	3	<b>Examen final</b>	<b>Sommative 15%</b>	

---

## Politique d'Évaluation des Apprentissages (PEA)

---

[http://www.cegep-ste-foy.qc.ca/fileadmin/documents/notre\\_cegep/politiques\\_et\\_reglements/5.9\\_POLITIQUE\\_EVALUATION\\_DES\\_APPRENTISSAGES.pdf](http://www.cegep-ste-foy.qc.ca/fileadmin/documents/notre_cegep/politiques_et_reglements/5.9_POLITIQUE_EVALUATION_DES_APPRENTISSAGES.pdf)