

TECHNIQUES DE L'INFORMATIQUE

(420.B0)

Automne 2026

PLAN D'ÉTUDES

Pondération : 2 - 3 - 2

420-565-MT

Unités : 2,33

Intelligence artificielle

75 heures

Préalables : PA 420-345-MT Programmation et architecture
PA 420-355-MT Programmation de jeux

Adopté en assemblée départementale
le 20 août 2026

Nadine Giasson St-Amand

O-129

poste 2127

stamandnadine@cgmataane.qc.ca

Description du cours

L'étudiant exploite l'intelligence artificielle dans des jeux et simulations. À partir de données, l'étudiant utilise des algorithmes qui permettent à un ordinateur d'apprendre de façon autonome pour résoudre des problèmes de classification et de traitement de signal. Il optimise les paramètres pour obtenir les meilleures classifications. Il se familiarise avec les notions d'heuristique, d'apprentissage, de sensibilité et de spécificité.

Objectifs de formation fondamentale du projet éducatif

- La communication et la maîtrise de la langue
- La pensée autonome
- Le sens des responsabilités

Objectifs et standards

OBJECTIF		STANDARD	
ÉNONCÉ DE LA COMPÉTENCE		CONTEXTE DE RÉALISATION	
OOSW : Effectuer le développement d'applications de jeu ou de simulation		• Pour différents jeux ou simulations : jeux d'action, jeux de rôles, simulateurs de vol, simulateurs de procédés industriels, etc. • Pour de nouvelles applications et des applications à modifier. • À partir des documents de conception. • À l'aide de moteurs de jeu ou de simulation. • À l'aide de sons et d'images 2D et 3D. • À l'aide de procédures de suivi des problèmes et de gestion des versions.	
ÉLÉMENTS DE COMPÉTENCE		CRITÈRES DE PERFORMANCE	
1	Analyser le projet de développement de l'application.	1.1	Analyse juste des documents de conception.
		1.2	Détermination correcte des tâches à effectuer.
2	Préparer l'environnement de développement informatique.	2.1	Installation correcte des logiciels et des bibliothèques.
		2.2	Configuration appropriée du système de gestion de versions.
		2.3	Importation correcte du code source.

3	Générer des représentations de mondes réels ou imaginaires.	3.1 Choix et utilisation appropriés des éléments graphiques pour l’affichage et la saisie. 3.2 Intégration correcte des images 2D et 3D. 3.3 Adaptation des représentations en fonction du format d’affichage et de la résolution.
4	Programmer la logique du jeu ou de la simulation.	4.1 Programmation correcte des comportements des éléments graphiques et des périphériques. 4.2 Programmation correcte des effets visuels. 4.3 Intégration précise des sons. 4.4 Programmation correcte des interactions. 4.5 Application correcte des techniques d’internationalisation. 4.6 Application rigoureuse des techniques de programmation sécurisée. 4.7 Utilisation appropriée des moteurs de jeu ou de simulation.
5	Contrôler la qualité de l’application.	5.1 Application rigoureuse des plans de tests. 5.2 Revues de code et de sécurité rigoureuses. 5.3 Pertinence des correctifs. 5.4 Respect des procédures de suivi des problèmes et de gestion des versions. 5.5 Respect des documents de conception.
6	Participer à la mise en service de l’application.	6.1 Préparation appropriée de l’application en vue de son déploiement, de son exportation ou de son installation. 6.2 Déploiement, exportation ou installation corrects de l’application.
7	Rédiger la documentation.	7.1 Détermination correcte de l’information à rédiger. 7.2 Notation claire du travail effectué.

Activités d'apprentissage

- Utiliser un algorithme intelligent dans un moteur de simulation.
- Exploiter une librairie d'intelligence artificielle dans un programme.
- Implémenter manuellement un algorithme intelligent
- Exploiter des algorithmes pour classifier des données ou détecter des objets.
- Exploiter des algorithmes pour transformer et compresser des données.
- Exploiter des algorithmes pour prendre des décisions.

Univers de l'intelligence artificielle

Application

Applications de classification, de détection, d'optimisation, de prédiction et de création

Applications dans une simulation ou un jeu

Applications techniques : exploration d'arbre, recherche de chemin "pathfinding", parsing, etc.

Concepts

Définition de l'intelligence artificielle

Espace de solution : paramétrisation, dimensions, projection

Recherche de minimum ou maximum, minimum ou maximum local

Heuristique & complexité

Concepts mathématiques

Matrice de confusion, spécificité et sensibilité

Equation du perceptron

Fonctions d'évaluation et Fonction d'activation

Calculs de rétropropagation

Convolution & Filtres

Corrélogramme & Matrice d'information mutuelle

Outils

Écosystème technologique

journal de code (notebook)

Librairies de science des données

Plateformes collaboratives

Pipeline : une chaîne de transformation

APIs & greffons

Modèles pré-entraînés

Algorithmes

Foules et essais

Boids : règles d'alignement, cohésion et séparation (pseudo-gradient)

Descente de gradient & refroidissement

Intégration dans une simulation 3D

Algorithmes génétiques

Population de solutions

Mutation et recombinaison

Sélection et élitisme

Exploitation dans une simulation
Ajustement des paramètres
Problèmes multi-objectifs

Machines d'état

Diagramme et notation
Applications des machines d'états
Programmation de machines d'état
Agents et coopération

Réseaux neuronaux

Réseau neuronal

Format des données et attributs
Neurone
Fonction d'activation
Structure du réseau neuronal
Couches de neurones
Entraînement

Structures de réseau

Perceptron
Réseau multi-couches
Réseau convolutif
Hyperparamètres

Mise en pratique

Échantillonnage
Préparation des données
Programmation d'un réseau neuronal
Entraînement
Évaluation des résultats

Calendrier des activités pédagogiques et d'évaluation

Moment	Contenu	Évaluation	
Introduction	Exploration et bases		Feuilles-synthèses 10% - tous les sujets
Semaines 1-2	Essaims et gradients	Laboratoires 10%	
Semaines 3-4-5	Algorithmes évolutifs	Laboratoires 10%	
Semaines 6-7	Agents et machines d'états	Laboratoires 10%	
PRÉPARER PROJET INTELLIGENT			Savoirs-lire 10% - algos évolutifs - réseaux neuronaux
Semaines 8-9-10	Réseaux neuronaux	Laboratoires 10%	
Semaines 11-12-13-14-15	Projet Intelligent	Projet Intelligent 40%	

* Le nombre de semaines de chaque module est un estimé et peut varier selon les besoins.

Méthodes et moyens pédagogiques

L'apprentissage s'articule autour de mini-projets intégrateurs réalisés en laboratoires qui servent à mettre en pratique les différentes techniques et notions du cours.

Les **méthodes** et **moyens** mis en place pour guider cet apprentissage sont :

- Un **Calendrier** des **Rencontres** hebdomadaires à présence obligatoire pour présenter la nouvelle matière et à réaliser les laboratoires
- Un **Espace virtuel** de liaison : soit un espace de discussion et un outil de vidéoconférence qui servent à faire des suivis et à fournir l'aide demandée entre les cours.
- Des documents mis à la disposition des participants classés par Modules dans un **Site**.

Tout au long du cours :

- *En support aux apprentissages sont présentés des **diaporamas**, des **articles**, des **vidéos**, des **exemples versionnés**, des **échantillons de code**, des **procédures**.*
- *Le participant est invité à réaliser un ou plusieurs **exercices** sous forme de travail dirigé.*
- *Ensuite, le participant expérimente la nouvelle technique en l'appliquant à son **laboratoire**.*
- *Enfin, le participant intègre et développe certains concepts en les appliquant à son **projet intelligent** qui est un projet intégrateur du cours. Occasionnellement, des veilles technologiques sont réalisées.*

Pour sa part, le participant devra adopter les pratiques suivantes :

=> L'étudiant devra **fournir un effort personnel** pour réaliser ses travaux de manière autonome. Il devra surtout prendre connaissance des notes de cours et expérimenter les échantillons et les exercices avant de poser des questions. Il devra suivre les étapes Recherche - Test - Question.

=> L'étudiant devra **versionner son code et son travail d'une manière régulière**, selon les instructions du guide de versionnement, et participer au processus **d'intégration continue**. Les ateliers **doivent être remis PENDANT la séance, AVANT de quitter la classe**, même s'ils ne sont pas terminés, afin de compter.

=> L'étudiant devra **rendre compte de son avancement dans les projets**, selon le format de la mêlée agile, il devra s'assurer de compléter ses parties de projet en temps utile pour ne pas compresser les étapes suivantes, et dans la même optique il devra signaler toute difficulté ou retard en temps utile. **Le travail se réalise d'abord en classe**. Les absences pour 'travailler sur le projet' ne sont pas acceptées. Le travail sur chaque partie du projet doit se versionner **en classe** et être complété à la maison.

=> L'étudiant devra **expliquer et même ajuster son code** à la demande pour valider chaque évaluation. Si l'étudiant est incapable d'expliquer le fonctionnement d'un code inscrit dans son projet ou s'il est incapable de le modifier au besoin, ce code ne sera pas considéré comme le sien et tombera sous la règle anti-plagiat.

Pour participer à ce cours, l'étudiant doit se procurer du matériel et des logiciels :

=> L'étudiant devra **se procurer le matériel nécessaire** :

- Ordinateur portable, écouteurs, micro & clé USB
- Accès internet & Serveur VPS
- Logiciels gratuits indiqués pour chaque projet

=> L'étudiant devra **maintenir la configuration d'ordinateur indiquée** en ce qui a trait aux logiciels requis ou interdits pour chaque chapitre du cours et utiliser les outils recommandés afin d'optimiser son apprentissage. Le dual-boot avec Linux est obligatoire pour les cours de session 5 incluant le cours Intelligence Artificielle.

En résumé les exposés théoriques et démonstrations seront suivis d'exercices pratiques effectués en laboratoire. La compétence sera perfectionnée par l'application dans un projet intégrateur au choix.

Sommaire des activités d'évaluation

Moyens	Moments	Pondérations
Savoirs-Lire et Feuilles-synthèses	Semaines 1 à 10	20%
Laboratoires Gradients	Semaines 1 & 2	10%
Laboratoires Évolatif	Semaines 3-4-5	10%
Laboratoires Agents	Semaines 6 & 7	10%
Laboratoires NeuralNet	Semaines 8-9-10	10%
Projet Intelligent (EFC)	Semaines 11 à 15	40%

Matériel obligatoire

=> L'étudiant devra **se procurer le matériel nécessaire** :

Un compte **Google Colaboratory**

- Incluant un upgrade de GPU (environ 15\$ par mois) lorsque exigé par le laboratoire.

Un **dual-boot** (Windows-KUbuntu) doit être installé sur l'ordinateur de l'étudiant

- Ce dual-boot doit être initié en début de session AVANT d'installer des logiciels pour les différents cours.
- Chaque système doit bénéficier d'au moins 100G de partition.

Une **extension de mémoire** externe d'au moins 128G

- Une clé USB ou une carte SD ou un disque SSD externe.

Une **unité VPS** (Virtual Private Server) selon les besoins des projets

- Le budget est d'environ 40\$ pour le serveur qui sera requis pour la partie projet et disponible pour certains labos.

Modalité des laboratoires

Les laboratoires se réalisent d'abord en classe et se complètent en devoir à la maison. Cependant la présence active à chaque séance de chaque laboratoire est obligatoire pour recevoir les points associés. La présence active signifie programmer et faire des commit tout au long de la séance. Toute absence non-justifiée à un laboratoire entraîne la note de 0. La remise des laboratoires se fait normalement à l'intérieur d'une fenêtre de temps de 7 jours suivant la séance,

cependant si les activités se suivent séquentiellement il faut compléter la remise avant la séance de laboratoire suivante. La rétention de code n'est pas permise sur le poste de travail ou dans une branche.

Description de l'épreuve finale de cours

Le Projet Intelligent (40%) constitue l'épreuve finale de cours.

Le projet Intelligent est un projet individuel qui permet à un étudiant d'exploiter un algorithme intelligent dans un projet de son cru ou un projet existant. Toutes les technologies et tous les algorithmes abordés dans le cours peuvent être choisis pour réaliser le projet. Le sujet, les technologies exactes et notamment les librairies, ainsi que l'architecture doivent être approuvés par le professeur suite à une proposition écrite.

Ensuite, en suivant le processus agile, l'étudiant réalise plusieurs itérations de son projet en se fixant des objectifs sous forme de ticket. Le projet s'accompagne de petites fiches à remplir qui guident les étapes du projet peu importe la technologie et l'application choisie. Chaque étape fait l'objet d'une remise intermédiaire et d'une démonstration de l'avancement.

Les quatre étapes du projet sont (1) Inspiration (2) Données (3) Apprentissage (4) Test.

Le projet se réalise en utilisant l'approche de l'intégration continue, c'est-à-dire en versionnant le code et les autres fichiers du projet au fur et à mesure de l'avancement.

- Le code devra suivre les Normes de programmation indiquées dans le cours, il devra notamment être sémantique et conforme au lexique du **dossier fonctionnel***. On doit respecter ces règles tout au long du processus (dans chaque commit).
- Des remises intermédiaires datées sous forme de TAG sont prévues dans chaque projet. Elles doivent être réalisées aux semaines indiquées afin de maintenir la qualité du projet.
- Suffisamment de temps doit être réservé pour les tests d'analyse des résultats et les nouveaux essais.

La réalisation du projet se réalise en laboratoire (présentiel) lors de toutes les périodes-contact afin d'échanger l'information, permettre le suivi du projet par le professeur et partager ses idées lors de la résolution de problèmes. Les absences non justifiées, incluant les retards ou longues pauses, ainsi que le manque de participation en-dehors des cours au projet seront pénalisés au prorata du temps de projet manquant. On facilitera ainsi les suivis, l'étalement du projet dans sa période de cogitation nécessaire, et la rencontre des critères d'intégration continue. On évitera ainsi que le projet ne soit réalisé en une période de temps réduite.

* Dossier fonctionnel :

Le dossier fonctionnel est soit l'énoncé sommaire fourni par l'enseignant (dans le cas d'un projet imposé), soit le dossier fonctionnel formel fourni par l'étudiant (dans le cas des projets au choix) et celui-ci est rédigé en langue française.

Intelligence artificielle

Algorithmes



Stuart Jonathan Russell, Peter Norvig

Intelligence artificielle

Pearson Education France, 10 décembre 2010

<https://books.google.ca/books?id=DWTIFWSGxJMC&printsec=frontcover>



Intelligence artificielle: une approche ludique

Tristan Cazenave, Ellipses, 2011

<https://www.amazon.ca/Intelligence-Artificielle-Une-Approche-Ludique/dp/2729864083>



AI for Game Developers

By David M. Bourg, Glenn Seemann · 2004

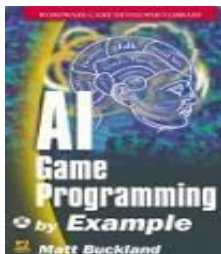
https://www.google.ca/books/edition/AI_for_Game_Developers/Sz-Sqvm-hSYC?hl=en&sa=X&ved=2ahUKEwi6-9qn977yAhVrEFkFHfoYDPYQiqUDMBh6BAgEEAw



Artificial Intelligence for Games

By Ian Millington, John Funge · 2018

https://www.google.ca/books/edition/Artificial_Intelligence_for_Games/V-KEDwAAQBAJ?hl=en



Programming Game AI by Example

https://www.google.ca/books/edition/Programming_Game_AI_by_Example/gDLpyWtFacYC?hl=en&gbpv=0

Neural networks and Deep learning (livre gratuit)

<http://neuralnetworksanddeeplearning.com/>

Plateformes & Communautés

<https://www.openml.org/>

<https://www.kaggle.com/>

<https://huggingface.co/>

<https://wandb.ai/site>

<https://roboflow.com/>

<https://www.analyticsvidhya.com/>

Intelligences artificielles populaires

<https://chatgpt.com/> par OpenAI

<https://claude.ai/> par Anthropic

<https://gemini.google.com/> par Google

<https://grok.com/> et <https://x.com/i/grok> par x.ai (anciennement Twitter)

<https://www.meta.ai/> par Meta (Facebook)

<https://www.deepseek.com/> (AI chinoise interdite au Cégep par décret du gouvernement)

APIs

OpenAI - <https://api.openai.com/v1/>

Anthropic (Claude) - <https://api.anthropic.com/v1/>

Google Gemini - <https://generativelanguage.googleapis.com/v1/>

Hugging Face - <https://api-inference.huggingface.co/>

Cohere - <https://api.cohere.ai/v1/>

Stability AI - <https://api.stability.ai/v1/>

Replicate - <https://api.replicate.com/v1/>

Mistral AI - <https://api.mistral.ai/v1/>

ElevenLabs - <https://api.elevenlabs.io/v1/>

AssemblyAI - <https://api.assemblyai.com/v2/>

Notebooks

<https://jupyter.org/try-jupyter/lab/>

<https://colab.google/> & <https://colab.research.google.com/>

<https://observablehq.com/>

<https://deepnote.com/workspace/>

Nouvelles des grands joueurs

<https://openai.com/> <https://openai.com/blog/>

<https://github.com/openai/> & <https://www.facebook.com/groups/OpenAIGroup>

<https://ai.googleblog.com/>

<https://www.anthropic.com/news>

<https://ai.meta.com/blog/>

Blogs

<https://deepmind.com/blog>

<https://bair.berkeley.edu/blog/>

https://blog.feedspot.com/ai_blogs/

Magazines & E-zines

<https://becominghuman.ai/>

<https://www.technologyreview.com/magazines/the-artificial-intelligence-issue/>

<https://www.hitachi.com/rd/sc/aiblog/index.html> (Industrial AI)

<https://towardsdatascience.com/>

Papers with code : <https://huggingface.co/papers/trending> & <https://github.com/paperswithcode>

<https://www.kdnuggets.com/>

Forums et médias sociaux

Reddit :

r/MachineLearning – discussions de recherche et actu.

r/LearnMachineLearning – parfait pour étudiants.

r/DataScience – plus généraliste, utile pour le côté métier.

Discords & Slack publics :

Hugging Face Discord (très actif, idéal pour poser des questions).

Alcrowd Discord (fort esprit communautaire, très proche de Kaggle mais plus interactif).

Fast.ai forums & Discord (super pour apprendre ensemble).

Livres gratuits

- Deep Learning (Goodfellow, Bengio, Courville)
👉 <https://www.deeplearningbook.org/>
- Artificial Intelligence: A Modern Approach (Russell & Norvig) – ressources
👉 <http://aima.cs.berkeley.edu/>
- Foundations of Machine Learning (Mohri, Rostamizadeh, Talwalkar)
👉 <https://cs.nyu.edu/~mohri/mlbook/>
- The Elements of Statistical Learning (Hastie, Tibshirani, Friedman)
👉 <https://hastie.su.domains/ElemStatLearn/>
- Dive into Deep Learning (D2L.ai) – livre interactif
👉 <https://d2l.ai/>
- Machine Learning Yearning (Andrew Ng)
👉 <https://www.deeplearning.ai/machine-learning-yearning/>
- Probabilistic Machine Learning (Kevin Murphy) – drafts
👉 <https://probml.github.io/pml-book/>
- Fastbook (fast.ai)
👉 <https://github.com/fastai/fastbook>
- Interpretable Machine Learning (Christoph Molnar)
👉 <https://christophm.github.io/interpretable-ml-book/>
- Full Stack Deep Learning – cours & notes
👉 <https://fullstackdeeplearning.com/>
- Machine learning Mastery
👉 <https://machinelearningmastery.com/>
- Fast AI
👉 <https://www.fast.ai/>
- Neural networks and deep learning
👉 <http://neuralnetworksanddeeplearning.com/>
- Artificial Intelligence: A Modern Approach, 4th US ed.
👉 <https://aima.cs.berkeley.edu/>

Datasets

Recherche

<https://datasetsearch.research.google.com/>

<https://archive.ics.uci.edu/>

Datasets communautaires

<https://www.kaggle.com/datasets>

<https://paperswithcode.com/datasets>

<https://huggingface.co/datasets>

Datasets démos de librairies

<https://www.tensorflow.org/datasets/catalog/overview>

<https://pytorch.org/vision/stable/datasets.html>

Datasets gouvernementaux/officiels

<https://www.data.gov/>

<https://open.canada.ca/fr/donnees-ouvertes>

<https://www.donneesquebec.ca/>

<https://data.europa.eu/en>

Exemples de code

<https://colab.google/notebooks/>

<https://huggingface.co/spaces>

<https://replicate.com/explore>

<https://gradio.app/demos/>

<https://streamlit.io/gallery>

<https://paperswithcode.com/>

<https://experiments.withgoogle.com/>

<https://platform.openai.com/examples>

<https://colab.research.google.com/notebooks/>

<https://tfhub.dev/>

<https://pytorch.org/tutorials/>

<https://keras.io/examples/>

<https://fastapi.tiangolo.com/tutorial/>

Modèles open-sources gratuits

<https://huggingface.co/models>

<https://tfhub.dev/>

<https://github.com/onnx/models>

<https://modelzoo.co/>

<https://keras.io/api/applications/>

<https://github.com/tensorflow/models>

<https://github.com/pytorch/vision/tree/main/torchvision/models>

<https://opencv.org/opencv-zoo/>

https://github.com/openvinotoolkit/open_model_zoo

<https://github.com/microsoft/onnxruntime/tree/main/onnxruntime/python/tools/transformers/models>

<https://github.com/salesforce/LAVIS>

https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md

<https://github.com/ultralytics/yolov5>

<https://github.com/WongKinYiu/yolov7>

<https://github.com/meituan/YOLOv6>

https://github.com/open-mmlab/mmdetection/docs/en/model_zoo.md

<https://github.com/PaddlePaddle/PaddleHub>

<https://github.com/google-research/bert>

<https://github.com/allenai/allennlp-models>

<https://pytorch.org/hub/>

Grands modèles rendus publics

<https://openai.com/index/introducing-gpt-oss/>

<https://huggingface.co/openai/gpt-oss-120b>

<https://github.com/openai/whisper>

<https://github.com/openai/CLIP>

<https://github.com/openai/point-e>

<https://github.com/openai/gpt-2>

Alternatives importantes

<https://huggingface.co/microsoft/DialoGPT-large>

<https://huggingface.co/EleutherAI/gpt-j-6b>

<https://huggingface.co/EleutherAI/gpt-neox-20b>

<https://huggingface.co/togethercomputer/RedPajama-INCITE-7B-Chat>

<https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1>

<https://huggingface.co/meta-llama/Llama-2-7b-chat-hf>

<https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B>

<https://huggingface.co/NousResearch/Nous-Hermes-2-Mixtral-8x7B-DPO>

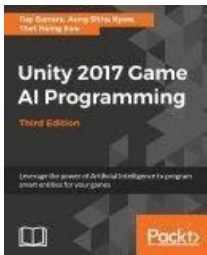
Programmation de l'IA

Engins pour les jeux et l'IA

<https://learn.unity.com/course/artificial-intelligence-for-beginners>

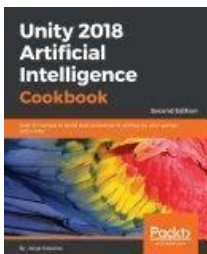
<https://unity.com/products/machine-learning-agents>

https://www.google.ca/books/edition/Unity_from_Proficiency_to_Mastery_Artifi/0gYBEAAAQBAJ?hl=en&gbpv=0



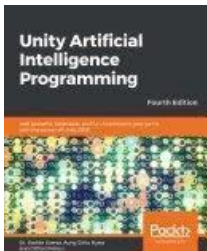
Unity 2017 Game AI Programming - Third Edition

https://www.google.ca/books/edition/Unity_2017_Game_AI_Programming_Third_Edi/qtRJDwAAQBAJ?hl=en&gbpv=0



Unity 2018 Artificial Intelligence Cookbook

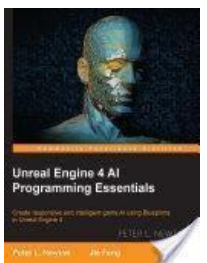
https://www.google.ca/books/edition/Unity_2018_Artificial_Intelligence_Cookb/chBrDwAAQBAJ?hl=en&gbpv=0



Unity Artificial Intelligence Programming

Add Powerful, Believable, and Fun AI Entities in Your Game with the Power of Unity 2018!, 4th Edition

https://www.google.ca/books/edition/Unity_Artificial_Intelligence_Programmin/ji59DwAAQBAJ?hl=en&gbpv=0



Unreal Engine 4 AI Programming Essentials

Peter L. Newton, Jie Feng

Packt Publishing Ltd

<https://books.google.ca/books?id=OGjiCwAAQBAJ&printsec=frontcover>

<https://www.packtpub.com/game-development/unreal-engine-4-ai-programming>

Python pour les jeux et l'IA

Python Machine Learning

https://www.w3schools.com/python/python_ml_getting_started.asp

Python for artificial intelligence

<https://wiki.python.org/moin/PythonForArtificialIntelligence>

Machine learning for Dummies

<https://books.google.ca/books?id=-ASEAAQBAJ&printsec=frontcover>

Python Machine Learning

https://www.google.ca/books/edition/Python_Machine_Learning/GOVOCwAAQBAJ?hl=en&gbpv=0

Artificial Intelligence and Machine Learning Fundamentals

https://www.google.ca/books/edition/Artificial_Intelligence_and_Machine_Lear/IW5_DwAAQBAJ?hl=en&gbpv=0

Introduction to Machine Learning with Python

https://www.google.ca/books/edition/Introduction_to_Machine_Learning_with_Py/vbQIDQAAQBAJ?hl=en&gbpv=0

Artificial Intelligence with Python

https://www.google.ca/books/edition/Artificial_Intelligence_with_Python/O1AoDwAAQBAJ?hl=en&gbpv=0

Deep Learning with Python

https://www.google.ca/books/edition/Deep_Learning_with_Python/wzozEAAQBAJ?hl=en&gbpv=0

Python Data Science Handbook

https://www.google.ca/books/edition/Python_Data_Science_Handbook/6omNDQAAQBAJ?hl=en&gbpv=0

Nature de l'AI & Définitions

The quest for artificial intelligence : <http://ai.stanford.edu/~nilsson/QAI/qai-webpage.html>

Lex Fridman : <https://www.youtube.com/channel/UCSHZKyawb77ixDdsGog4iWA/featured>

Arxiv Insight : <https://www.youtube.com/channel/UCNikB2leJ-6AmZv7bQ1oBYg>

https://en.wikipedia.org/wiki/Machine_learning

https://fr.wikipedia.org/wiki/Apprentissage_automatique

<https://www.intel.ca/content/www/ca/en/artificial-intelligence/blog.html>