

Sujets de projets tuteurés 2023–2024

1. Découper la recette de l'omelette (programmation python)

En traitement automatique des langues la segmentation est la tâche qui consiste à découper un texte en phrases et en mots. Cette tâche est relativement facile et résolue pour la langue écrite, mais pose de nombreuses difficultés pour le traitement de l'oral, où l'unité "phrase" est mal définie et où on peut voir apparaître ce qu'on appelle des dysfluences (marques d'hésitation, répétitions, mots tronqués) comme dans l'exemple : "et *euuh* il faut *il faut* apprendre à à s~ je dirais se se faire à ces variations".

Étapes :

1. État de l'art des techniques de segmentation de l'oral
2. Développement d'un outil de segmentation prenant en compte les spécificités de l'oral et application sur un corpus transcrit provenant de la base Cocoon (par exemple la recette de l'omelette <https://cocoon.huma-num.fr/exist/crdo/meta/cocoon-08671076-9f44-3c18-9771-37efe1965041> - cliquer sur "Afficher" sous le lecteur à droite pour voir la transcription de la recette de l'omelette.)

2. Capre : Conception d'un IDE pour Capacity

Capacity est un cadre formel de modélisation et d'évaluation du contrôle sur les données personnels dans un système d'information.

La modélisation du contrôle consiste en des *exigences* exprimées sur des *opérations* effectuées par des *agents* sur des *ressources* dans un *contexte*.

L'objectif de ce projet est de concevoir un IDE (environnement de développement) pour le langage de Capacity, pour faciliter la modélisation de système d'information.

Cela implique de pouvoir spécifier les différents objets (agents, ressources, opérations, contextes) puis d'exprimer des exigences sur ces objets suivant le modèle de Capacity.

À la fin, il faudra être capable de générer du code correspondant à la modélisation dans un format donné pour que le moteur de Capacity puisse s'en servir.

3. Predict the rating of a book

Using a publicly available dataset, that contains information on published books, the student will be asked to develop an NLP/machine learning model that uses the provided data and performs the following tasks.

Task#1: This part of the project focuses on the use of NLP methods to extract features from raw text. The student will have to scan the literature/online sources and propose features for this analysis.

What you will learn: Data preprocessing, feature extraction.

Task#2: Using the extracted features, the student has to build a model that predicts the genre of the book from the text. For this, she/he will have to:

- Build a feature matrix and start with a simple binary classification.
- Expand the classification to a multiclass case.
- After completing this step, the student must be able to report the classification metrics and explain them clearly in the text.

What you will learn: Classification

Task#3: At this stage, the dataset will be augmented with additional features provided by the supervisor. The student will then have to perform classification using the different feature matrices and compare the performance of a model based on the three different matrices (details below)

- Matrix#1: Text extracted features
- Matrix#2: Provided features
- Matrix#3: Combined

What you will learn: Model and feature interpretation.

4. Assistant de rédaction interactive pour créer des rapports en latex

Le projet consiste à développer une application interactive en Python qui assiste les étudiant-e-s dans la rédaction de rapports en LaTeX. L'application servirait de plateforme tout-en-un pour la rédaction, la génération de code LaTeX, et même la compilation en PDF si possible. Cesi est une introduction pratique à LaTeX, aux technologies d'apprentissage automatique et de développement d'interface utilisateur.

On utilisera de bibliothèques puissantes comme transformers pour l'intelligence artificielle et Tkinter pour l'interface graphique. L'application sera conçue pour

fournir des suggestions utiles sur la structure d'un document LaTeX, y compris comment ajouter des éléments comme des tableaux, des figures et des listes. L'un des points forts de cette application serait l'intégration d'un modèle GPT pour aider à la rédaction. Par exemple, l'assistant pourrait suggérer des phrases qui améliorent la qualité du texte. Pour rendre l'application encore plus interactive, on peut envisager l'intégration d'une fonction qui permet de compiler et de prévisualiser le code LaTeX directement dans l'application.

Concernant le codage (Python) :

- Interface Utilisateur (UI) : utilisation de Tkinter ou d'une autre bibliothèque d'interface graphique pour créer une interface utilisateur simple mais fonctionnelle.
- Intégration du modèle de langage : utilisation de la bibliothèque transformers pour intégrer un modèle GPT (comme DistilGPT) qui peut aider à la rédaction de texte.
- Fonctions de génération de code LaTeX : fonctions qui génèrent des bouts de code LaTeX en fonction des besoins des utilisateurs. Par exemple, une fonction qui génère un bloc de code pour insérer une image en LaTeX.
- Compilateur LaTeX intégré (optionnel) : intégration d'un système pour compiler le code LaTeX directement depuis l'application et afficher le résultat PDF. Cela peut être fait en utilisant des appels système pour exécuter une commande pdflatex.
- Analyse textuelle : utilisation de bibliothèques comme spaCy ou NLTK pour ajouter des fonctionnalités d'analyse de texte plus avancées si nécessaire.

Le codage sera en Python en travaillant l'intégration de divers composants pour créer une application complète et utile.

5. Implémentation d'un jeu de Hanabi

Hanabi est un jeu de société coopératif dont vous trouverez les règles ici: [https://fr.wikipedia.org/wiki/Hanabi_\(jeu\)](https://fr.wikipedia.org/wiki/Hanabi_(jeu)).

- Implémenter un jeu de Hanabi avec une interface graphique.
 - Développer une IA capable de jouer selon différentes stratégies : certitude, omniscience, stratégie d'information incomplète, méthodes arborescentes
 - Ajouter une fonctionnalité réseau permettant le jeu multijoueur.
-

6. Intégration de données à partir de plusieurs sources

Lorsqu'on travaille sur différents projets à la fois, ou sur un gros projet à multiples facettes, on accumule très vite un nombre élevé de données de natures diverse telles que des mails, des documents ou encore des tâches. Même si on prend la peine d'organiser ces données en arborescence de fichiers, en dossiers de mails, etc, lorsqu'on doit accéder à différentes informations il est nécessaire de switcher d'app en app (ou de logiciel en logiciel) et de naviguer jusqu'à l'endroit où sont stockées les informations.

L'objectif du projet est alors de faire un tour d'horizon de l'existant en termes de gestion de données, puis de développer une interface qui permette d'accéder aisément à des données de différentes natures relatives à un même projet.

7. Clustering d'objets dans des images hyperspectrales

- état de l'art clustering
 - implémentation différentes méthodes
 - comparaison résultats (validation experte possible)
-

8. Cocote : Développement d'une application web d'élaboration collective de textes

L'objectif de ce projet est le développement d'une application web permettant l'élaboration collective de textes. L'idée est qu'étant donnée une première version d'un texte, l'application opère automatiquement à un découpage et une numérotation des sections, paragraphes, listes, etc. pour permettre ensuite aux utilisatrices de proposer des amendements au texte avec une interface conviviale permettant de faire des suggestions d'ajouts, de suppressions, et de modifications.

L'application doit aussi proposer une interface permettant aux utilisatrices d'exprimer leur préférence sur chaque amendements via un vote (différents systèmes de vote doivent être implémenter suivant le type d'amendement : un simple pour/contre, assentiment ou notation en cas de choix multiple).

Enfin, l'application doit être capable de produire automatiquement la versions finales consolidées des textes, prenant en compte les résultats des votes.

9. Interpreteur pour dessiner

Le but : Créer un interpreteur qui interprète un code contenant (entre autre) des instructions de dessin afin de produire une image.

Fonctions : Interpretation d'instructions de simple tracé ou de remplissage qui se combinent à des instructions de transformations géométriques (peut-être dans l'espace 3D).

Possibilité d'incorporer des commandes de programmation ou/et la possibilité de créer et utiliser des macros-instructions éventuellement incluses dans des fichiers annexes.

Illustration : On peut s'inspirer du logiciel tex (latex) qui lui interprète du code pour mettre en page du texte.

10. Debunking, fact-checking, détection de fake news

- état de l'art multidisciplinaire sur :
 - la post-vérité,
 - le fact-checking,
 - la modélisation de fake news,
 - la détection de fake news
 - comparaison des benchmark disponibles
 - étude de la généralité des algos publiés et de la variété des données
 - proposition d'extensions
-

11. Effets peinture

Il s'agit de prendre une image et de lui donner un rendu peinture.

- État de l'art sur les effets peinture.
 - État de l'art sur la forme des coups de pinceau.
 - Génération automatique de coups de pinceau sur l'image.
 - Remplacement de l'image par une série de coups de pinceau.
-

12. Book Recommendation System using Graph Neural Networks

In this project, the student will develop a book recommendation system using Graph Neural Networks (GNNs) to provide book recommendations based on user interactions with books. The project involves the following key steps:

Task#1: Familiarize with the dataset (<https://www.kaggle.com/code/dushyantnagar7806/book-recommender-system-project>) This dataset should include data such as book attributes (e.g., title, genre) and user-book interactions (e.g., ratings).

Task#2: Data Preprocessing. In this part of the project, the student will have to perform data cleaning and preprocessing to handle missing values, and outliers, and format the data for GNN input.

What you will learn: Data preprocessing.

Task#3: Graph Neural Network (GNN) Design. The student should design and implement a Graph Neural Network architecture for learning user and book embeddings from the interaction graph.

What you will learn: Understanding of recommendation systems and Graph Neural Networks.

Task#4: GNNs

What you will learn: Evaluation and optimization of machine learning models.

Task#5: User Interface (Optional) The student can create a user-friendly interface (e.g., web application or command-line tool) to interact with the recommendation system, where users can input their preferences and receive book recommendations.

13. Portal 2D ou 3D collaboratif en Godot

Portal est un jeu de réflexion dont le but est de traverser une série de salles en résolvant des énigmes via la création de portails : [https://fr.wikipedia.org/wiki/Portal_\(jeu_vid%C3%A9o\)](https://fr.wikipedia.org/wiki/Portal_(jeu_vid%C3%A9o))

- Créer un jeu de Portal en 2D (ou 3D) sur Godot 4 avec plusieurs énigmes.
 - Incorporer un mode multijoueur en réseau permettant de résoudre des énigmes en collaboration.
-

14. SpimScalp : Intégration de SeseLab dans Spim

SeseLab est une plateforme d'enseignement des attaques physiques écrite en Python. Spim est un simulateur de processeur MIPS écrit en C/C++, qui sert également pour l'enseignement.

L'objectif de ce projet est d'intégrer les fonctionnalités pertinentes de SeseLab (la sonde de consommation de courant, la possibilité de faire des injections de fautes, la bibliothèque bignum) dans le projet SPIM pour permettre la réalisation d'atelier d'enseignement des attaques physiques sur une "vraie" plateforme.

Si le temps le permet de développement d'un TP orienté contre-mesure serait intéressant également.

15. Dessin de graphe

But : Écrire un programme qui à partir de la description d'un graphe (liste des arêtes et des sommets) produit une représentation dessinée du graphe (où les sommets sont reliés par des traits représentant les arêtes) en essayant de disposer les sommets et les arêtes de façon que le résultat soit "clair/jolie". Par exemple, une stratégie serait d'essayer de placer dans un même voisinage les sommets reliés par une arête.

Fonctions : On peut imaginer le programme sous la forme d'un interpréteur, on pourrait par exemple forcer la place de certains sommet. On peut aussi imaginer une interaction graphique où on déplace les sommets avec un curseur et une modification du dessin en temps réel.

16. Migration et extension de la plateforme START avec intégration de modules Python

En combinant la programmation, la gestion de version et la gestion de projet, ce projet offre une expérience complète tout en travaillant sur des technologies actuelles et pertinentes en analyse textuelle.

La plateforme START est actuellement conçue en Java et est utilisée comme un outil pédagogique pour l'ingénierie des langues. L'objectif de ce projet est de créer un pont entre cette plateforme Java existante et des modules Python. Ceci permettra d'intégrer les modules Python dans une interface utilisateur Tkinter, ce qui rendra l'expérience plus interactive.

Objectifs Techniques

- Conception d'une architecture hybride: concevoir une architecture qui permettra d'intégrer efficacement des modules Python dans la plateforme Java existante.
- Test des modules externes: tester et intégrer des modules Python existants. Ces modules sont développés par d'autres étudiants et sont destinés à des analyses textuelles spécifiques.
- Création d'une interface utilisateur avec Tkinter: pour rendre l'expérience utilisateur plus interactive, une interface utilisateur en Python en utilisant la bibliothèque Tkinter sera créée.

Outils

Java pour la plateforme existante et Python pour le développement de nouveaux modules et de l'interface utilisateur. Utilisation Git pour maintenir une version propre et suivie du code. Pour suivre les tâches, les étudiants utiliseront Wekan, qui est une alternative open-source à Trello. Ils pourront créer des cartes pour chaque tâche et suivre leur progression de manière visuelle.

17. Contrôle d'un perso platformer à l'aide d'une caméra type Kinect

étude et développement d'une gestuelle permettant de contrôler un personnage dans un platformer ou d'un shoot them up 2D sideview

18. VK4D : état des lieux et étude de faisabilité

Il s'agit d'étudier l'architecture logicielle de la bibliothèque Vulkan (petite soeur d'OpenGL) afin de pouvoir concevoir puis réaliser un prototype de sur-couche logicielle, voire une suite d'outils éventuellement couplés avec un langage de description, qui permettrait d'aborder facilement l'utilisation de Vulkan par des néophytes. Le projet sera :

- Opensource
 - Multiplateformes
-

19. Algorithmes évolutionnaires pour le décodage d'erreurs

Dans un canal de transmission, des interférences peuvent intervenir et corrompre les messages à transmettre. Pour répondre à ce problème, on encode les messages en leur ajoutant une certaine redondance. Cette redondance permet alors de réparer certaines erreurs grâce à un algorithme de décodage.

Pour certaines familles d'encodages (notamment, ceux dits "aléatoires"), les meilleurs algorithmes de décodage sont assez lents. Le but de ce projet est d'étudier *expérimentalement* des méthodes basées sur les algorithmes "évolutionnaires" (famille d'algorithmes d'optimisation dont le principe s'inspire de la théorie de l'évolution) afin de décoder ces erreurs.

Remarques :

- sujet de découverte (notions a priori non-vues en classe),
 - pas de prérequis, mais il faudra une certaine curiosité scientifique.
-

20. Makros : Macros clavier universelles dans Kate

Kate est l'éditeur de texte avancé du projet KDE. C'est un logiciel développé en C++ avec les frameworks KDE et la bibliothèque Qt.

Récemment lui a été ajoutée une fonctionnalité de macros clavier, c'est-à-dire, la possibilité d'enregistrer des séquences de frappe répétées pour automatiser la manipulation de texte. Cependant une limitation subsiste : la manipulation au clavier de certains composants graphiques (widgets) échoue lors du rejeu des macros qui en font usage. C'est notamment le cas de composants centraux pour la manipulation clavier efficace, comme la palette de commandes et la palette de sélection rapide de fichier.

L'objectif de ce projet tuteuré est de comprendre le fonctionnement de ces composants graphiques spéciaux et ce qui les différencie des autres afin de corriger le bug. (Ce projet tuteuré peut être couplé avec le cours de développement de logiciel libre.)

21. GL4D-comp : wrapper d'OpenGL classique (antique) incluant GLU et GLUT

Participer au développement de la bibliothèque GL4D (<https://gl4d.api8.fr>) en proposant des modifications et des ajouts aboutissant à un wrapper complet d'OpenGL « classique » vers OpenGL « moderne ». Il s'agit de pouvoir recompiler des codes sources ciblant OpenGL 2 ou 1, GLU et GLUT pour les faire tourner sur une base OpenGL 3.3+ gérée par le biais de GL4D. On se donnera comme premier objectif de pouvoir recompiler et faire tourner le maximum d'exemples « classiques » (ou Legacy) proposés par l'ancien site NeHe (<http://nehe.gamedev.net/>) dont l'archive est encore disponible sur <https://github.com/qamedev-net/nehe-opengl>

22. skinOfGL4D : prise en charge bas niveau du skinning et du rigging

Ce projet a pour cadre la participation au développement d'une extension pour GL4D (<https://gl4d.api8.fr>). A l'aide d'une API de chargement de modèles 3D articulés (par exemple libAssimp ou USD), il est attendu de développer un ensemble de nouvelles fonctionnalités permettant de mettre à jour les surfaces liées au modèle afin de gérer l'articulation par bones ainsi que le rigging. Les principales étapes sont :

- Proposer un état de l'art sur l'animation de personnages articulés ;
 - Faire un choix — argumenté — concernant le modèle de fichier et l'API de chargement du modèle ;
 - Concevoir et réaliser, en C/GLSL et au sein de l'API GL4D, les nouvelles fonctionnalités décrites plus haut ;
 - Montrer, par le biais d'un exemple utilisant ces nouvelles fonctionnalités, l'étendu de ce qui est possible de faire avec. Nous donnons à titre d'exemple ce viewer web <https://playcanvas.com/viewer> basé sur TreeJS et capable de manipuler les paramètres d'un modèle 3D au format glTF.
-

23. Du cactus à l'hippopotame. Dictionnaire étymologique inversé

Connaître l'étymologie d'un mot c'est connaître son origine. Par exemple le mot "sandwich" vient du nom du 4e comte de Sandwich (une ville américaine), qui l'a popularisé. Dans certains cas, les mots dérivent d'une même origine grecque ou latine, comme par exemple la "météo(ologie)" et les "météorites".

Le wiktionnaire propose une section Etymologie pour un grand nombre d'entrées (voir par exemple : <https://fr.wiktionary.org/wiki/quoicoubeh>), mais il n'est pas possible de trouver tous les mots qui dérivent d'une même origine.

- À partir d'un dump du wiktionnaire, extraire les liens entre entrées lexicales (mots du wiktionnaire) et leur origine (à partir des liens présents dans l'étymologie par exemple).
 - Proposer une interface qui permet à partir d'un mot de trouver tous les mots qui partagent la même étymologie (par exemple, à partir du mot "cactophile" - qui aime les cactus -, retrouver "potamophile" - qui vit en eau douce -... et à partir de "potamophile", retrouver "hippopotame" - cheval d'eau douce).
-

24. Replicate the results of the article "Deep Learning for Symbolic Dynamics"

The aim of this project is to understand and replicate the results of the article entitled "Deep Learning for Symbolic Dynamics" written by G. Lample and F. Charton. This article demonstrates that neural networks can be surprisingly good at solving elaborated mathematical tasks.

Task#1: Generating the dataset that will be used in the task.

Task#2: Cleaning the dataset.

Task#3: Experiments. A seq2seq model will be trained to predict solutions to given mathematical problems.

25. IA pour Antichess par recherche arborescente

Antichess est un jeu jouable sur un échiquier standard avec un principe simple : pour gagner, il faut se faire capturer toutes ses pièces. Les règles sont disponibles ici: [https://fr.wikipedia.org/wiki/Qui_perd_gagne_\(%C3%A9checs\)](https://fr.wikipedia.org/wiki/Qui_perd_gagne_(%C3%A9checs))

- Implémenter un jeu de Antichess sur des échiquiers de taille variable
 - Développer des IA utilisant différentes méthodes de recherche arborescente : minimax, élagage alpha-beta, Monte Carlo Tree Search
 - Comparer les résultats obtenus en faisant s'affronter les différentes IA.
-

26. Pavages 2D en Racket

Il s'agit d'étudier des pavages périodiques et non périodiques dans le plan et de les dessiner avec le package racket/draw.

- https://fr.wikipedia.org/wiki/Groupe_de_papier_peint
 - https://en.wikipedia.org/wiki/Penrose_tiling
 - <https://docs.racket-lang.org/draw/index.html>
-

27. L-Systèmes

Ce projet a pour cadre la participation au développement d'une extension pour GL4D (<https://gl4d.api8.fr>). Il s'agit d'un module englobant de nouvelles fonctions permettant la modélisation à la volée, et au sein de GL4D, de maillages générés par L-Systèmes (arbres ...), selon des spécifications qu'on retrouve dans l'ouvrage « algorithmic beauty of plants ». Ci-après quelques étapes importantes du projet :

- Acquérir des connaissances et faire un état de l'art sur les L-Systèmes ;
 - Proposer et concevoir l'intégration du module à GL4D (en extra, un éditeur en ligne pourra être proposé) ;
 - Développer le parser et le générateur de L-Systèmes ;
 - Développer l'interprète permettant de passer au modèle 3D : idéalement un maillage unifié -> voir les isosurfaces ;
 - Proposer au moins un rendu adapté à ces modèles.
-

28. Gros corpus brut ou petit corpus annoté ? Combiner apprentissage non supervisé et apprentissage supervisé pour le traitement automatique de langues peu dotées (programmation python)

Le traitement automatique des langues (ou TAL, en anglais NLP : Natural Language Processing) s'attache à développer des "outils capables d'interpréter et de synthétiser du texte pour diverses applications" (définition de la CNIL). Ces outils s'appuient sur des mécanismes d'apprentissage divers alimentés par des données linguistiques. L'apprentissage non supervisé repose sur de grandes quantités de données peu coûteuses, dites brutes (par exemple des textes), alors que l'apprentissage supervisé requiert une plus faible quantité de données plus coûteuses à produire (par exemple des textes enrichis d'informations linguistiques). Lorsque peu de données sont accessibles, trouver le bon équilibre entre ces deux techniques est fondamental.

Étapes :

1. choisir une tâche linguistique pour une langue particulière (voir par exemple les langues et les datasets présents ici : <https://universaldependencies.org/>)
 2. faire un état de l'art des ressources existantes pour cette langue et des techniques mises en oeuvre pour résoudre la tâche choisie
 3. Entraîner divers modèles sur la base des ressources disponibles
 4. Faire une étude comparative en faisant varier 1/ la taille des données 2/ les techniques d'apprentissage pour comparer l'efficacité des techniques.
-

29. Recherche de collisions

But : Un travail de recherche pour implanter (et trouver) de manière efficace des collisions d'une fonction (typiquement une fonction de hachage).

29. Compression d'images : méthode empirique de réduction du nombre de couleurs

État de l'art sur les méthodes de compression avec perte.

- Construction de la CLUT.
 - Trouver les agrégats de couleurs, les formes générales dans le nuage de points.
 - Choisir pour chaque agrégat la couleur par laquelle il sera représenté.
 - Conserver l'image et sa CLUT.
-

30. Évaluation avancée et amélioration de Scrapy pour le web scraping adaptatif - Apprentissage de l'architecture DOM et scrapers adaptatifs

Le scraping du web est un domaine essentiel de l'ingénierie des données. Alors que différents outils sont disponibles pour cette tâche, le projet propose une exploration en profondeur de Scrapy, l'un des outils les plus utilisés, pour comprendre ses forces et faiblesses par rapport à d'autres scrapers comme BeautifulSoup ou Selenium. Une étude comparative de ces outils sur des critères comme la vitesse, la flexibilité, la facilité d'utilisation et l'efficacité dans l'extraction de données sans bruit serait très instructive.

Objectifs techniques

- Analyse comparative: faire une évaluation comparative de Scrapy par rapport à BeautifulSoup et Selenium. Les critères d'évaluation incluront la vitesse, la flexibilité et l'efficacité dans l'extraction de données.
- Stratégies d'extraction 'sans bruit': développement de techniques pour une extraction propre et précise de données textuelles, par exemple depuis des forums.
- Analyse de la structure du DOM de différents sites web : une alternative open-source aux DevTools de Chrome, pourrait-être par exemple les Firefox Developer Tools qui offrent des fonctionnalités similaires aux DevTools de Chrome. Ou alors les sélecteurs XPath ou CSS en Python où les bibliothèques comme BeautifulSoup en Python peuvent être utilisées pour examiner la structure d'une page web, bien que ce ne soit pas aussi interactif que les DevTools.
- Développement de scrapers adaptatifs: en utilisant des techniques d'apprentissage supervisé, par renforcement ou non supervisé, on va créer des scrapers adaptatifs avec Scrapy. Ces scrapers seront capables de s'ajuster à différentes architectures de sites web grâce à un apprentissage fine-tuned sur des structures de DOM spécifiques. Cela permettra une extraction de données plus robuste et adaptable.
- Évaluation de l'adaptabilité: mesurer l'efficacité et la précision des scrapers adaptatifs par rapport à des scrapers standard.

Une solide compréhension de l'architecture DOM est nécessaire, de même que des compétences en XPath ou CSS pour l'extraction de données et une familiarité avec des méthodes d'apprentissage machine pour le développement de scrapers adaptatifs.

31. PyGL4D : wrapper Python pour GL4D

Tout est dans le titre, il s'agit de participer au développement de la bibliothèque GL4D (<https://gl4d.api8.fr>) en proposant un wrapper (et l'ensemble de la procédure — simple — d'installation) permettant d'exploiter pleinement et efficacement les fonctionnalités de GL4D avec le langage Python.

32. Compression d'images : réduction du nombre de couleurs HSV

Vous travaillerez en mode HSV sur la base d'une CLUT dont il s'agit de réduire le nombre de couleurs.

- État de l'art sur les méthodes classiques de compression avec perte et sur les modes de couleurs.
 - Construction de la CLUT, puis d'une CLUT en HSV.
 - Classement des couleurs (transformation de la CLUT).
 - Réduction du nombre de couleurs.
-

33. VR on Linux - Godot & GL4D

Projet assez ouvert où il s'agira d'étudier et de proposer la meilleure solution « bas-niveau » (C/OpenGL) permettant d'utiliser en multiplateformes (donc surtout sous Linux où les solutions sont réduites) plusieurs casques VR dont les Oculus et surtout le HTC Vive. Nous visons à priori un couplage Godot/GL4D pour ce projet, néanmoins, après étude des alternatives possibles, sur la base d'argumentation, une autre solution pourra être envisagée pour la partie implémentation du projet.

34. Puppet-Incarnation : apprentissage supervisé basé simulation et exploitation sur caméra RGBD

C'est un projet de « vision », il s'agit de simuler en 3D temps-réel une partie articulée du corps — plus ou moins détaillée et plus ou moins large — afin d'en faire une base d'apprentissage pour un réseau neuronal (à priori). L'idée est de pouvoir exploiter cet apprentissage afin d'articuler en temps-réel (comme devant un miroir) les parties simulées par le biais d'une caméra RGBD filmant le l'utilisateur. Dans ce projet, le visage est la partie la plus complexe à traiter, il sera donc judicieux de commencer par une autre partie du corps telle que la main.

35. Pavages de polyominos en JavaScript

Il s'agit d'étudier les polyominos et de proposer des solutions de pavage avec une ou plusieurs pièces fixées, en 2D avec l'API Canvas, en 3D avec la librairie three.js.

- <https://en.wikipedia.org/wiki/Polyomino>
 - https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API
 - <https://threejs.org/>
-

36. Mid++ : sampler automatique et synthétiseur pour fichiers midi

Projet très ouvert dont l'objectif « large » est de proposer un outil permettant d'améliorer la restitution audio des fichiers midi. Il s'agira d'avoir donc, après analyse de la piste midi, une partie générant des échantillons d'instruments couplée à une partie qui exploite au mieux les échantillons générés afin de synthétiser le son final. Une ou plusieurs possibilités de post-traitement sur la sortie pourront être proposées. L'utilisation directe de banques de sons est proscrite, on pourra par contre limiter les genres musicaux gérés mais en gardant un spectre assez large (exemple : heavy-metal et musique classique). Pour ce projet, l'analyse détaillée de l'existant (dans la recherche, dans les solutions OpenSources ou commerciales) est incontournable avant de débiter la phase de conception et de développement.

37. Vérification et application des conventions de programmation

Lorsqu'un projet est développé en équipe, des conventions de programmation sont normalement imposées aux développeurs. Il est donc important de pouvoir vérifier régulièrement si ces conventions sont bel et bien respectées.

L'objectif de ce projet est d'étudier dans un premier temps les différents types de conventions et d'identifier celles qui peuvent être vérifiées automatiquement, et comment, puis de développer un outil de vérification, voire même de correction automatique ou semi-automatique.

38. Échecs à l'aveugle avec reconnaissance en Python

Il s'agit d'étudier les résultats d'algorithmes de recherche arborescente pour jouer aux échecs à l'aveugle avec reconnaissance, en commençant par minimax, alphabeta, mcts, alphazero et en poursuivant avec leurs variantes.

- <https://rbc.jhuapl.edu/>
 - <https://github.com/suragnair/alpha-zero-general>
-

39. Simulation en GPU

Il s'agit de simuler un phénomène naturel (biologique, physique voire sociologique) par le biais de modèles procéduraux ou itératifs. Ce projet comprendra aussi bien la simulation que la visualisation du résultat obtenu et aura pour objectif d'être temps réel. L'usage du GPU est ainsi fortement recommandé (Cuda ou OpenGL/GLSL). Les principales étapes de ce projet sont :

- Proposer un état de l'art en fonction de la problématique choisie ;
 - Concevoir et réaliser le(s) module(s) de simulation ainsi que la partie visualisation ;
 - Réaliser des mesures de performance sur les différents composants et si possible comparer ces résultats avec l'existant.
-

40. Introduction à la biométrie par l'authentification floue

La biométrie est une branche de la cryptologie qui utilise les caractéristiques morphologiques des individus comme valeurs secrètes. C'est par exemple le cas de l'authentification par empreinte digitale.

Dans ce projet, le but est simuler une authentification biométrique à partir d'images simples. Plusieurs aspects pourront être étudiés : la reconnaissance d'éléments caractéristiques de l'image, la sécurisation des données personnelles, la malléabilité du système, etc.

Pour conduire ce travail, on pourra par exemple se baser sur des schémas de "coffre flou" (fuzzy vault).

Remarques :

- sujet assez transverse
 - la partie "sécurité" requiert une petite appétence pour les mathématiques
-

41. Transfert d'apprentissage pour StarCraft2 en Python

Il s'agit d'étudier des algorithmes de transfert d'apprentissage pour résoudre des mini-missions du jeu StarCraft2.

- <https://www.deepmind.com/open-source/pysc2>

- <https://github.com/google-deepmind/pysc2>
 - <https://github.com/thuml/Transfer-Learning-Library>
-

42. Rendu expressif : stylisation fonctionnelle ou artistique

Projet tuteuré à spectre large autour du rendu expressif (donc à spécifier de manière détaillée après discussion avec le tuteur). Le rendu expressif est un sous-domaine de la synthèse d'images s'intéressant aux techniques infographiques permettant de générer des images « artistiques » ou aillant une utilité fonctionnelle (ex. plan d'architecte). Les principales étapes de ce projet sont :

- Proposer un état de l'art du sous-domaine en mettant l'accent sur le « style » choisi pour le projet ;
 - Concevoir et réaliser le module de traitement : abstraction(s), étapes de stylisation, ... si possible par le biais d'un pipeline temps réel (en partie ou à 100% en GPU) ;
 - Comparer les résultats produits avec l'existant et se donner un moyen de les évaluer qualitativement.
-

43. Nuées d'étourneaux

On voit souvent des nuées d'étourneaux dans le ciel. Leur mouvement semble concerté tant il est cohérent.

- Modélisation sommaire d'un étourneau.
 - État de l'art des systèmes masse-ressort.
 - Génération de mouvements de nuées d'étourneaux.
 - Interaction aléatoire avec un/des rapace.
-