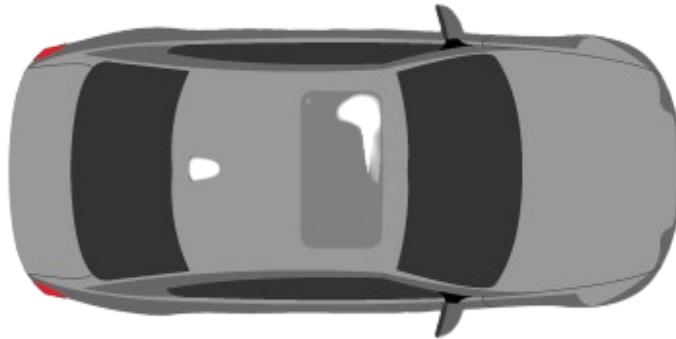


UE ARE DYNAMIC

Projet:

Modélisation d'une circulation routière



Auteurs:

Rym Benoumechiara
Sokhna Oumou Diouf

Enseignante : Garance Lucas

PLAN

1- Introduction.

2- Thématique et question principale.

3- Modèle, Choix Réalisés et Paramètres.

4- Résultats

5. Analyse

6. Conclusion

7. Perspectives d'Amélioration

8. Résumé en anglais

1- INTRODUCTION

Chaque jour, dans les grandes villes comme dans les zones périurbaines, des millions de véhicules circulent, s'arrêtent, attendent, redémarrent. Cette complexité quotidienne n'est pas le fruit du hasard, mais résulte d'une multitude de règles, d'interactions et de phénomènes parfois imprévisibles, comme les embouteillages. Alors que les défis liés à la mobilité urbaine ne cessent de croître, comprendre les dynamiques de circulation devient crucial pour penser la ville de demain.

Avec l'émergence de la ville intelligente (smart city), la modélisation du trafic routier prend une place centrale dans les projets de recherche et d'innovation. Il ne s'agit plus uniquement de tracer des routes ou de réguler le trafic avec des feux tricolores, mais de comprendre et simuler le comportement des véhicules dans un environnement contraint, avec ses obstacles, ses priorités, ses règles de circulation, et surtout ses imprévus.

Ce projet s'inscrit dans cette démarche en proposant une simulation 2D d'un réseau routier simplifié. Notre objectif est de représenter un système dynamique où des agents (les véhicules) évoluent dans un environnement structuré, interactif et contraint par diverses règles (sens uniques, signalisation, obstacles, interactions avec d'autres agents). L'environnement, la génération des entités et les logiques de comportement des véhicules ont été développés en utilisant la librairie Python Pygame, permettant une visualisation interactive de la simulation.

2- Thématique et question principale

2-1 Thématique du projet:

Ce projet s'inscrit dans le champ de la **modélisation de systèmes dynamiques appliqués à la circulation routière urbaine**. Il vise à explorer et simuler les comportements d'un réseau routier dans un espace quadrillé 2D, en intégrant des éléments fondamentaux de la réalité : des véhicules mobiles considérés comme des agents autonomes, des feux de signalisation à cycle variable, des directions de circulation prédéfinies, des obstacles physiques statiques et dynamiques (ajoutés par l'utilisateur), des piétons interagissant avec le trafic, et des points de destination spécifiques pour chaque véhicule.

2.2. Problématique: (Question Principale)

La question centrale qui guide notre travail est la suivante : Comment représenter une circulation routière urbaine dans un environnement simplifié, tout en prenant en compte les interactions complexes telles que les feux de circulation, les piétons, les obstacles et les collisions évitées ? De plus, quels impacts certains paramètres, comme la vitesse, peuvent-ils avoir sur la fluidité et la sécurité de cette circulation, en intégrant des véhicules agents dotés de capacités de décision autonomes ?

3- Modèle, Choix Réalisés et Paramètres

Cette section détaille la conception du modèle de simulation, les choix techniques effectués pour son implémentation, et les paramètres qui régissent son fonctionnement.

3.1. Approche Générale et Environnement:

Notre simulation repose sur une grille 2D de cellules carrées, où chaque cellule peut représenter soit une route, soit une zone non routière (ex. bâtiments, espaces verts), soit un obstacle temporaire. Cette grille est dimensionnée à partir de la résolution de la fenêtre, avec des cellules de taille fixe de 40 pixels, ce qui permet un rendu lisible et modulaire.

Les routes sont automatiquement générées selon une logique régulière : Les lignes 1, 4, 7, 10, etc. (tous les 3 rangs) et les colonnes correspondantes sont définies comme routes. Les autres cellules restent "non-routières" et accueillent des bâtiments ou des arbres selon une configuration prédéfinie.

- **Bâtiments, écoles et arbres** sont placés sur des cellules non routières à des positions précises. Les maisons sont représentées graphiquement par un rectangle (mur) surmonté d'un triangle (toit), et les arbres par un tronc et un feuillage (rose ou vert).
- **Les passages piétons(zones prioritaires)**, représentés graphiquement sur la grille de simulation, sont généralement situés à proximité des feux de circulation ou aux intersections de routes. Ils peuvent également être placés de manière aléatoire sur les routes, à condition que la cellule comporte des routes contiguës horizontalement ou verticalement, tout en évitant leur superposition avec des obstacles ou des feux, afin de préserver la cohérence de l'environnement. Lorsqu'un véhicule arrive à un passage piéton, il doit impérativement s'arrêter si un piéton traverse, respectant ainsi la priorité accordée aux piétons. Cette règle introduit une complexité supplémentaire dans la gestion du trafic en intégrant une priorité aux entités piétonnes dans le cadre de la simulation.

3.2- Interactivité utilisateur:

Un aspect fondamental du projet réside dans l'interaction offerte à l'utilisateur, qui peut ajouter ou retirer des obstacles directement sur la grille de simulation par un simple clic de souris. Cette fonctionnalité permet de représenter des événements extérieurs tels que des travaux ou des accidents, tout en testant la robustesse et l'adaptabilité du système.

Les obstacles, qui ne peuvent être ajoutés que sur les routes, sont positionnés aléatoirement sur la grille et sont des éléments infranchissables. Lorsqu'un obstacle entre en conflit avec la trajectoire initiale d'un véhicule, celui-ci est contraint de re-calculer son itinéraire. En perturbant les trajets et en créant des détours, ces éléments contribuent à la complexification des schémas de circulation, jouant ainsi un rôle clé dans la dynamique de la simulation.

3.3- Feux de signalisation:

Les feux tricolores sont positionnés automatiquement à des emplacements stratégiques, principalement aux intersections ou carrefours importants, en respectant une densité maximale par ligne et colonne. Leur placement repose sur une heuristique simple, nécessitant au moins trois routes connectées autour, et est influencé par la structure globale du réseau routier plutôt que totalement aléatoire.

Chaque feu de circulation suit un cycle dynamique composé de trois phases : vert (15 secondes), orange (3 secondes) et rouge (10 secondes). Pour éviter que tous les feux changent simultanément, un décalage aléatoire est appliqué au démarrage. Lorsqu'un feu passe au rouge, les véhicules arrivant à son niveau sont contraints de s'arrêter et de patienter jusqu'à ce que le feu devienne vert pour poursuivre leur trajet. Ce mécanisme joue un rôle crucial dans la régulation et la fluidité du trafic urbain simulé.

3.4- Intégration des piétons:

Pour davantage de réalisme, des **piétons** apparaissent aléatoirement sur les passages piétons valides et traversent en fonction de leur orientation (horizontal/vertical), sauf si une voiture est arrêtée sur la case. Ce mécanisme contribue à modéliser les interactions véhicule/piéton.

3.5- Directions de circulation:

Afin d'éviter les comportements aléatoires irréalistes, chaque ligne et chaque colonne de la grille impose une **direction de circulation** : les lignes sont orientées "droite" ou "gauche" en alternance, et les colonnes "haut" ou "bas".

Ce choix contraint les déplacements des voitures en leur imposant des trajets plus crédibles et des détours réalistes.

3.6- Génération et gestions des voitures:

Chaque voiture est un **agent autonome** doté :

- D'une **position de départ** et une **destination finale**, toutes deux choisies **aléatoirement** parmi les cellules praticables (routes non bloquées).
- D'un **numéro unique** croissant pour les identifier, visible à l'écran.
- D'une **image personnalisée** avec un changement de couleur des voitures lorsqu'elles pénètrent dans le réseau routier (si le fichier **car.png** est présent), sinon un simple cercle coloré est affiché.

Les destinations sont **fixes**, en couleur jaune et accompagnée du numéro de la voiture. et les voitures utilisent l'**algorithme A*** pour calculer le **chemin le plus court**, en respectant les directions de circulation imposées par chaque route. Ce pathfinding est déclenché dès l'apparition de la voiture, et recalculé en cas d'obstacle ou de blocage.

Les voitures **respectent les feux** en s'arrêtant si le feu sur leur trajectoire est rouge, et **attendent qu'il passe au vert**. Elles sont également capables de détecter les voitures devant elles pour éviter les collisions et les blocages.

3.7- Algorithme de cheminement:

Pour que chaque voiture atteigne sa destination de manière optimale, nous avons intégré l'algorithme **A*** (A-Star). Il s'agit d'un algorithme de pathfinding basé sur une **heuristique** qui permet de trouver le chemin le plus court tout en prenant en compte :

- Les **obstacles physiques**
- Les **voitures déjà présentes sur le chemin**
- Les **directions autorisées des routes**
- Les **feux rouges** (bloquants)

Cette approche assure que les voitures ne se déplacent que vers des cases logiques, sécurisées et respectant la dynamique d'une ville réelle.

3.8- Fonctions et paramètres

Pour modéliser un système de circulation réaliste, plusieurs fonctions essentielles ont été implémentées. Ces fonctions gèrent le déplacement des véhicules, la gestion des obstacles, la signalisation routière et l'adaptation dynamique du trafic.

Catégorie	Fonction	Description	Paramètres
Gestion des déplacements	<code>est_case_escapable</code>	Vérifie si une case est une route praticable et connectée à d'autres routes, pour valider les destinations et éviter les impasses.	<code>position, taille_x, taille_y, grille</code>
Gestion des déplacements	<code>trouver_chemin</code>	Utilise l'algorithme A* pour calculer le chemin optimal entre deux points tout en respectant les sens uniques et évitant les obstacles.	<code>grille, départ, arrivée, directions_lignes, directions_colonnes</code>
Gestion des déplacements	<code>trouver_nouvelle_destination_valide</code>	Réattribue dynamiquement une destination alternative si le trajet initial est bloqué, simulant les changements d'itinéraire en temps réel.	<code>voir code</code>
Gestion des obstacles	<code>ajouter_obstacle_manuel / auto</code>	Ajoute ou retire des obstacles sur les routes : manuels via clic souris, automatiques placés aléatoirement.	Retourne : bool (succès/échec)
Gestion des obstacles	<code>forcer_recalcul_si_affecte</code>	Déclenche un recalcul des itinéraires pour les voitures impactées par un nouvel obstacle.	<code>Voir code</code>
Signalisation et éléments routiers	<code>initialiser_passages_pietons_sur_routes</code>	Place aléatoirement des passages piétons sur les routes, en évitant les feux et obstacles.	<code>n_passages, taille_x, taille_y, feux, grille</code>
Signalisation et éléments routiers	<code>initialiser_feux_repartis_sur_routes</code>	Positionne les feux tricolores aux intersections stratégiques avec un cycle dynamique (vert → orange → rouge) et un décalage aléatoire entre les feux.	<code>Voir code</code>
Signalisation et éléments routiers	<code>mettre_a_jour_feux</code>	Fait évoluer l'état des feux tricolores en fonction de leur timing prédefini.	<code>voir code</code>
Noyau de simulation	<code>mettre_a_jour_voitures</code>	Fonction principale gérant en temps réel les arrivées, recalculs d'itinéraires, nouvelles destinations et mouvements des véhicules.	<code>DELAI_MIN_MOUVEMENT, SEUIL_BLOCAGE, MAX_RECALCUL_ECHECS, autres paramètres : voitures, grille, etc.</code>
Affichage et visualisation	Ordre des éléments : Fonds, décor, interactifs, dynamiques	Dessine tous les éléments de la simulation (routes, obstacles, piétons, voitures), en évitant les superpositions.	<code>TAILLE_CELLULE, VITESSE_VOITURE, NOMBRE_VOITURES_CIBLE</code>

4

fig 1: quelques fonctions résumés

3.8.1- Des constantes

Afin de garantir la cohérence, la lisibilité et la flexibilité de notre code, nous avons utilisé de nombreuses constantes. Celles-ci définissent l'environnement graphique, les comportements dynamiques, les seuils de décision, ainsi que les éléments visuels et décoratifs. Leur centralisation dans le fichier de configuration permet une **modification simple et rapide des paramètres globaux** de la simulation.

Constantes	Valeurs	Descriptions
LARGEUR	1200	Largeur de la fenêtre de simulation (en pixels)
HAUTEUR	600	Hauteur de la fenêtre de simulation (en pixels)
TAILLE_CELLULE	40	Taille d'une cellule carrée dans la grille (en pixels)
TAILLE_X_GRILLE	30	Nombre de colonnes dans la grille
TAILLE_Y_GRILLE	15	Nombre de lignes dans la grille
BLANC	(255, 255, 255)	Couleur du fond
NOIR	(0, 0, 0)	Couleur des bordures et obstacles manuels
GRIS_CLAIR	(200, 200, 200)	Couleur des zones non routières
GRIS_ROUTE	(55, 55, 55)	Couleur du fond des routes
VERT	VERT	Couleur pour l'état des feux de signalisation (vert)
ORANGE	ORANGE	Couleur pour l'état des feux de signalisation (orange)
ROUGE	ROUGE	Couleur pour l'état des feux de signalisation (rouge)
COULEUR_PASSAGE	(220, 220, 220)	Couleur des passages piétons
JAUNE_PARKING	(255, 190, 0)	Couleur des destinations
COULEUR_PIETON	(0, 0, 0)	Couleur des piétons
ROUTE	" "	Symbolique représentant une cellule praticable
NON_ROUTIER	"N"	Symbolique représentant une cellule non routière
OBSTACLE_MANUEL	"X"	Symbolique représentant un obstacle manuel
OBSTACLE_AUTO_SYM	"A"	Symbolique représentant un obstacle automatique
SYMBOLES_NON_PRATICABLES	SYMBOLES_NON_PRATICABLES	Symbolique pour tous les types de cellules non accessibles
VITESSE_VOITURE	0.75	Vitesse des voitures (déplacements par seconde)
DELAI_MIN_MOUVEMENT	1.0 / VITESSE_VOITURE	Temps minimal entre deux mouvements

fig 2: constantes

4- Résultats

Cette section présente les résultats concrets observés lors de l'exécution de notre modèle de simulation. L'objectif est de démontrer le fonctionnement effectif des mécanismes décrits précédemment (Section 3) et de visualiser les dynamiques de trafic émergentes.

4.1- Visualisation de l'Environnement Initial Statique

Pour établir la fondation sur laquelle repose l'ensemble de la simulation, nous présentons d'abord l'état initial et statique de l'environnement. La **Figure 3** illustre la grille générée automatiquement au lancement du programme, avant l'introduction de tout élément dynamique (véhicules, feux, piétons, etc.)

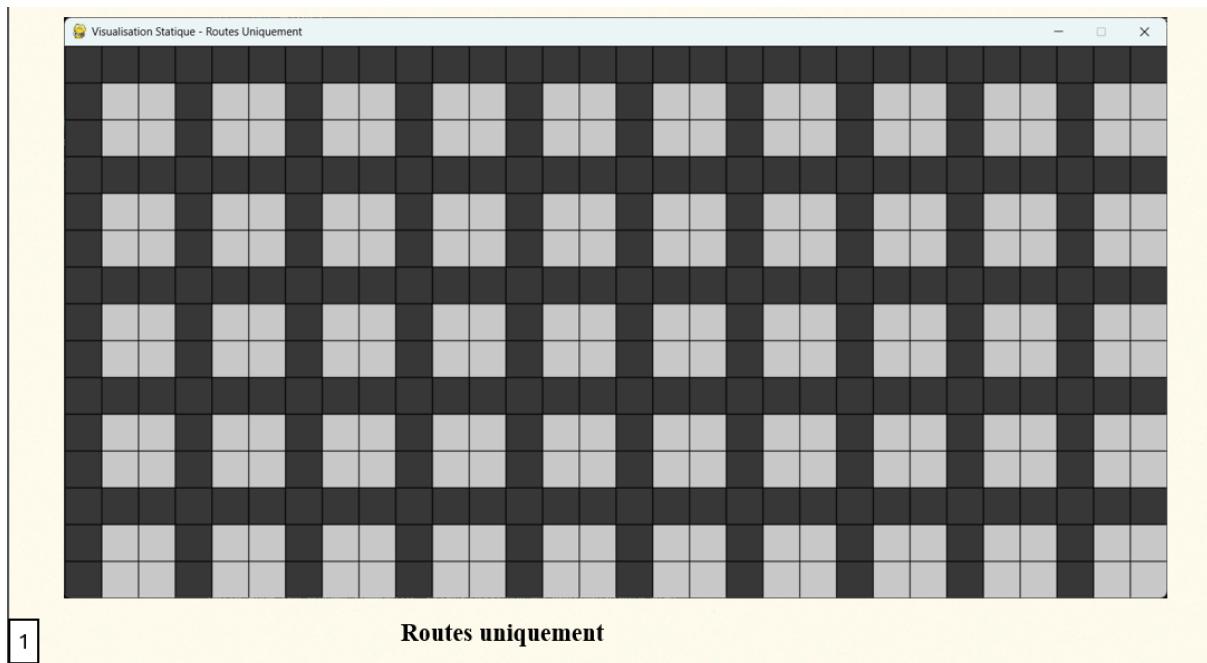


Fig 3: routes uniquement

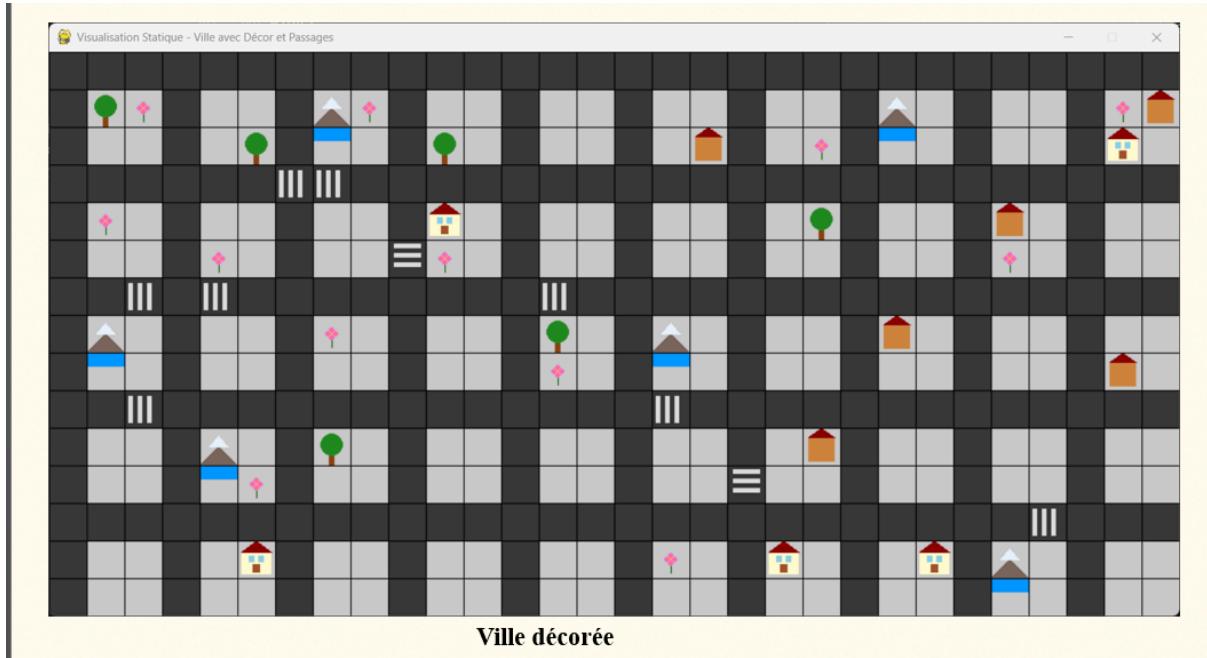


fig 4: ville avec routes et décoration

En complément de la structure routière de base (illustrée en Figure 3), l'environnement statique de simulation est enrichi par des éléments de décor et l'ajout des passages piétons initiaux. La Figure 4 montre le résultat de cette phase d'initialisation.

Enrichissant davantage la représentation de l'environnement statique, la **Figure 5** intègre désormais la visualisation des directions de circulation prédéfinies pour chaque ligne et colonne de la grille routière. Ces directions constituent une règle fondamentale contrignant les déplacements des véhicules.

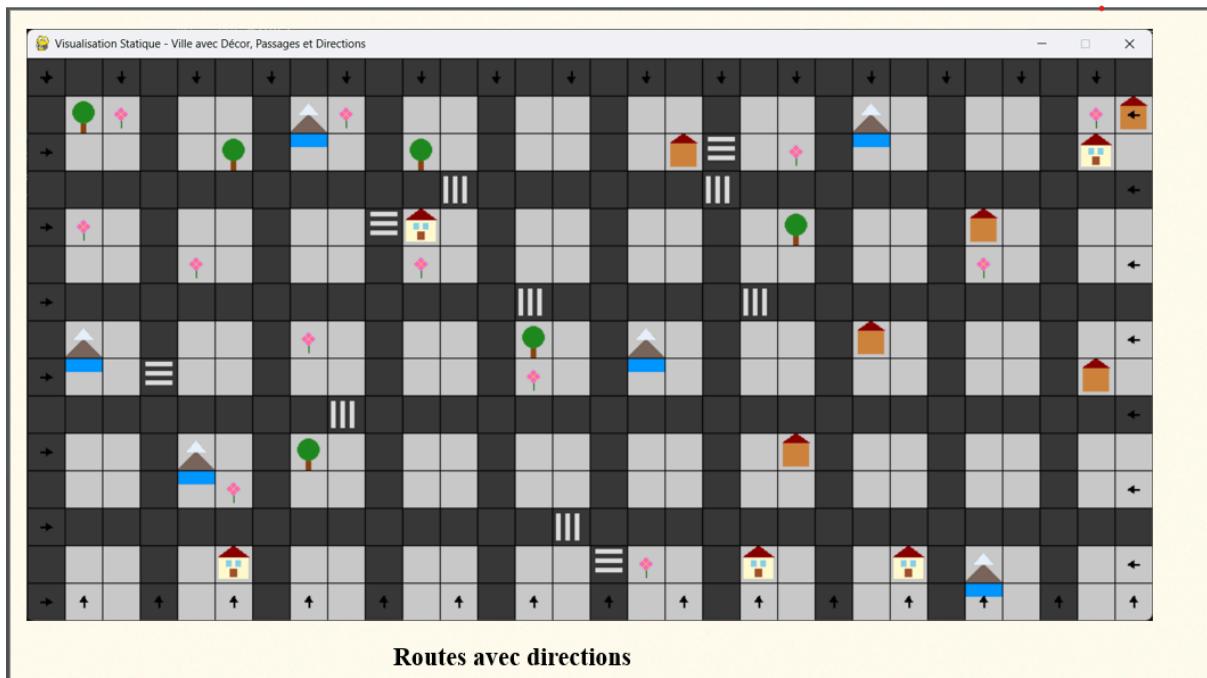


fig 5 : directions dessinées

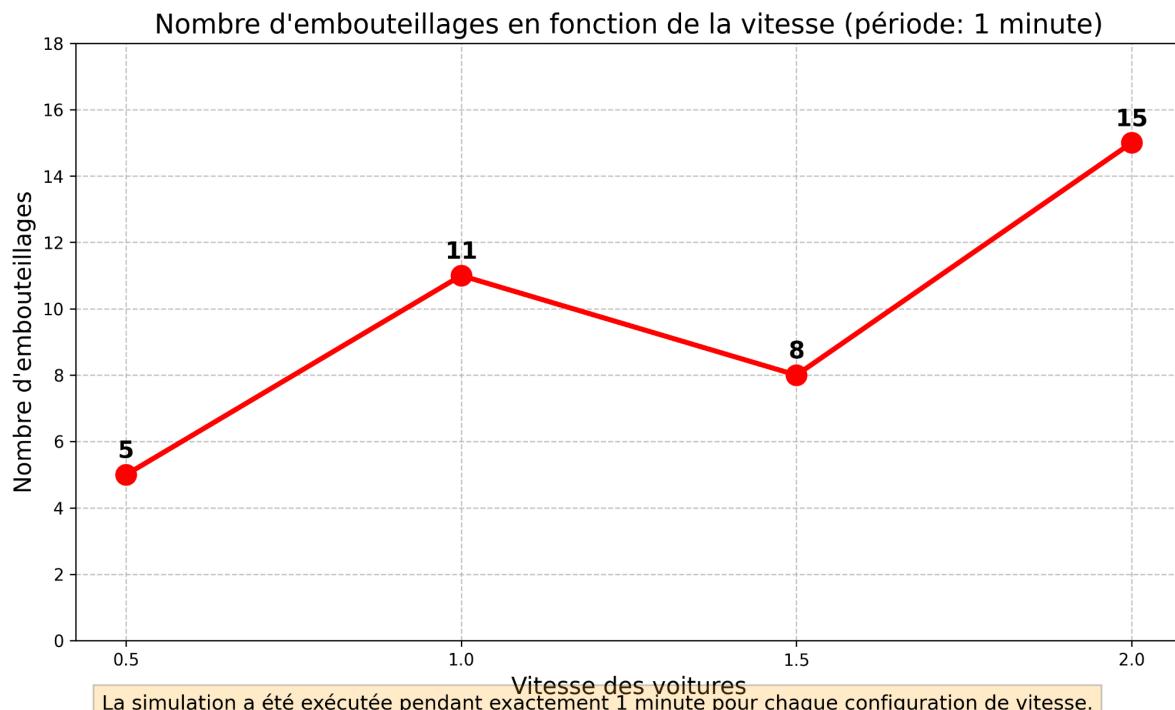
4.2. Dynamique de Circulation en Régime Stabilisé

Après l'initialisation, les voitures sont générées et commencent leur parcours. On observe généralement les comportements suivants :

- **Navigation:** Les voitures se déplacent le long des chemins calculés, une cellule par pas, respectant les sens de circulation indiqués. Leur vitesse effective est limitée par DELAI_MIN_MOUVEMENT et la nécessité d'attendre que la prochaine cellule soit libre.
- **Respect des Règles Statiques:** Les voitures évitent les zones non routières et les obstacles permanents définis dans la grille, conformément au chemin calculé par A*.

5. Analyse :

5.1. Graphe :



5.2. Analyse du graphe :

Le graphique ci-dessous illustre l'évolution du nombre d'embouteillages en fonction de la vitesse des véhicules, mesurée sur une période fixe d'une minute de simulation. On observe une tendance non linéaire en forme de "W" : le nombre d'embouteillages augmente d'abord entre les vitesses 0.5 et 1.0, diminue à 1.5, puis croît à nouveau à 2.0. Cette dynamique s'explique par plusieurs mécanismes liés à l'interaction des véhicules au sein du réseau.

- À vitesse très faible (0.5), les voitures avancent lentement, occupent longtemps les intersections, mais leur progression ralentie réduit le nombre de nouveaux véhicules entrant dans la zone de simulation. Cela limite les interactions et donc les embouteillages, malgré le temps de passage prolongé à chaque point de croisement.
- En passant à vitesse moyenne (1.0), les véhicules se déplacent plus rapidement, ce qui augmente la densité de circulation : davantage de voitures peuvent entrer et circuler dans le réseau sur la même période. Cependant, cette densité plus élevée génère plus de croisements simultanés aux intersections, augmentant la probabilité de conflits et donc d'embouteillages.
- À vitesse modérément élevée (1.5), une amélioration apparaît. La vitesse suffisante permet aux voitures de traverser les intersections rapidement et efficacement, réduisant la durée d'occupation des zones critiques. Le flux devient plus fluide, et bien que la densité reste relativement élevée, la capacité d'évacuation du réseau compense, ce qui diminue le nombre d'embouteillages.
- Enfin, à vitesse maximale (2.0), la situation se détériore brutalement. Les véhicules arrivent aux intersections à une vitesse trop élevée, sans assez de temps pour ajuster leur comportement face aux autres. Cela génère des situations chaotiques, des blocages soudains et une forte hausse du nombre d'embouteillages. La rapidité excessive réduit la possibilité de coordination locale entre véhicules, ce qui dégrade sévèrement la fluidité du trafic.

En conclusion, cette analyse montre que ni la vitesse minimale ni la vitesse maximale ne sont idéales. Une vitesse modérée offre un compromis optimal entre fluidité et gestion des interactions.

6. Conclusion

Ce projet a permis de concevoir et d'implémenter une simulation de trafic routier 2D offrant une représentation interactive et dynamique d'un écosystème urbain simplifié. En intégrant une grille spatiale, des règles de circulation définies (sens uniques), des éléments de régulation (feux) et d'interaction (piétons, obstacles), ainsi que des agents véhicules dotés de capacités de navigation avancées (A*) et d'adaptation

(détection de blocage, recalcul d'itinéraire), nous avons réussi à modéliser les interactions complexes inhérentes à la circulation urbaine.

La simulation démontre qualitativement comment les choix d'algorithme de pathfinding, les règles de priorité (feux, piétons) et les mécanismes de gestion des perturbations (obstacles, blocages) influencent la dynamique globale du trafic. Les agents sont capables d'atteindre leurs destinations, d'éviter les collisions entre eux, de réagir à l'environnement dynamique et de s'adapter dans une certaine mesure aux modifications de la grille ou aux impasses temporaires.

Bien que simplifiée, cette modélisation atteint les objectifs fixés et constitue une base solide pour explorer des scénarios de trafic, visualiser l'impact de certains éléments (position des feux, présence d'obstacles), et sert de plateforme d'expérimentation pour des règles de comportement d'agents plus sophistiquées.

7. Perspectives d'Amélioration

Ce projet ouvre plusieurs pistes pour des travaux futurs :

- **Validation Quantitative:** Collecter des métriques (temps de trajet moyens, délais aux intersections, nombre de recalculs, etc.) pour analyser et valider plus précisément l'impact des paramètres et comparer l'efficacité de différentes configurations de réseau ou règles de trafic.
- **Réalisme Routier:** Modéliser des chaussées avec plusieurs voies, autorisant les dépassements ou les changements de voie selon des règles précises.
- **Comportement Agent Avancé:** Intégrer des modèles de comportement de conducteur plus réalistes (vitesse variable, respect des distances de sécurité plus complexes, réaction aux véhicules d'urgence, covoiturage).
- **Régulation Smarter:** Implémenter des feux de circulation dynamiques dont les cycles s'adaptent en fonction du trafic détecté en temps réel, plutôt que des cycles fixes.
- **Piétons Complexes:** Permettre aux piétons d'avoir leurs propres destinations et logiques de traversée, potentiellement avec attente du moment opportun pour traverser le trafic automobile.
- **Obstacles Modulables:** Ajouter différents types d'obstacles avec des effets variés (ralentissement, rétrécissement de voie, blocages partiels ou complets selon le type de véhicule).
- **Interface Utilisateur Riche:** Permettre de sélectionner des voitures pour inspecter leur état (chemin, destination), ajouter des points de passage forcé sur un chemin, ou définir manuellement les destinations.
- **Large Échelle:** Adapter l'algorithme de pathfinding (ou en utiliser un plus avancé, comme A* bidirectionnel ou D* Lite) pour gérer des cartes beaucoup plus grandes et des environnements plus denses efficacement.

8.Résumé en anglais

This project presents a 2D simulation model of urban road traffic, developed using Python and the Pygame library. The core objective is to explore and visualize the complex dynamics of vehicle movement within a simplified, interactive urban environment.

The simulation is built upon a 2D grid where cells represent either navigable roads or non-road areas populated with static decorative elements (such as buildings, trees, schools, mountains, and flowers). The road network itself is procedurally generated following a regular pattern (every 3rd row and column).

Autonomous vehicle agents are the primary dynamic elements. Each vehicle is assigned random start and destination points within the road network. They utilize the A* pathfinding algorithm to calculate the shortest valid path, adhering to several crucial constraints:

- **Fixed One-Way Directions:** Each road segment (row or column) has a predefined, alternating direction (e.g., left/right or up/down).
- **Traffic Lights:** Strategically placed at intersections, these lights follow dynamic green-orange-red cycles with randomized offsets to prevent perfect synchronization. Vehicles obey the light signals.
- **Obstacles:** The simulation includes impassable obstacles. Users can interactively add or remove "manual" obstacles (representing events like road work) via mouse clicks, forcing affected vehicles to recalculate their routes. Automatic obstacles can also appear.
- **Pedestrians:** Pedestrian crossings are placed on valid road segments. Pedestrians randomly appear and cross, and vehicles correctly yield priority to them.
- **Collision Avoidance:** Vehicles detect cars directly ahead and wait to avoid collisions.

The simulation features logic for handling blockages: if a vehicle's path is obstructed or becomes invalid, it attempts to recalculate. If repeatedly unsuccessful, it employs logic to find a new valid destination, simulating real-world rerouting.

Key parameters like vehicle speed are configurable via constants, allowing observation of their impact on traffic flow, congestion, and safety.

Overall, the project aims to realistically represent complex traffic interactions (vehicle-vehicle, vehicle-infrastructure, vehicle-pedestrian) in a constrained grid environment, providing an interactive platform to study dynamic traffic systems and agent behavior.

Rapport intégralement rédigé par Rym Benoumechiara et Sokhna Oumou Diouf

