

Rapport projet

I. Implémentation

Nous avons implémenté quatre sémaphores `semLightHorizontal`, `semLightVertical`, `semRoadHorizontal` et `semRoadVertical`. Nous les avons initialisés respectivement à 1, 0, 1 et 1.

Pour l'algorithme, nous avons suivi l'algorithme suivant :

Pour le feu :

Feu	Voiture
<pre>feu = true répéter : attendre() si feu = true alors : P(semLightVertical) v(semLightHorizontal) feu = false sinon : P(semLightHorizontal) v(semLightVertical) feu = false</pre>	<pre>avancer_jusqu_au_feu() attendre_au_feu() Si type = vertical alors : P(semRoadVertical) P(semLightVertical) traverser_Intersection() V(semLightVertical) V(semRoadVertical) Sinon P(semRoadHorizontal) P(semLightHorizontal) traverser_Intersection() V(semLightHorizontal) V(semRoadHorizontal) finir_de_traverser()</pre>

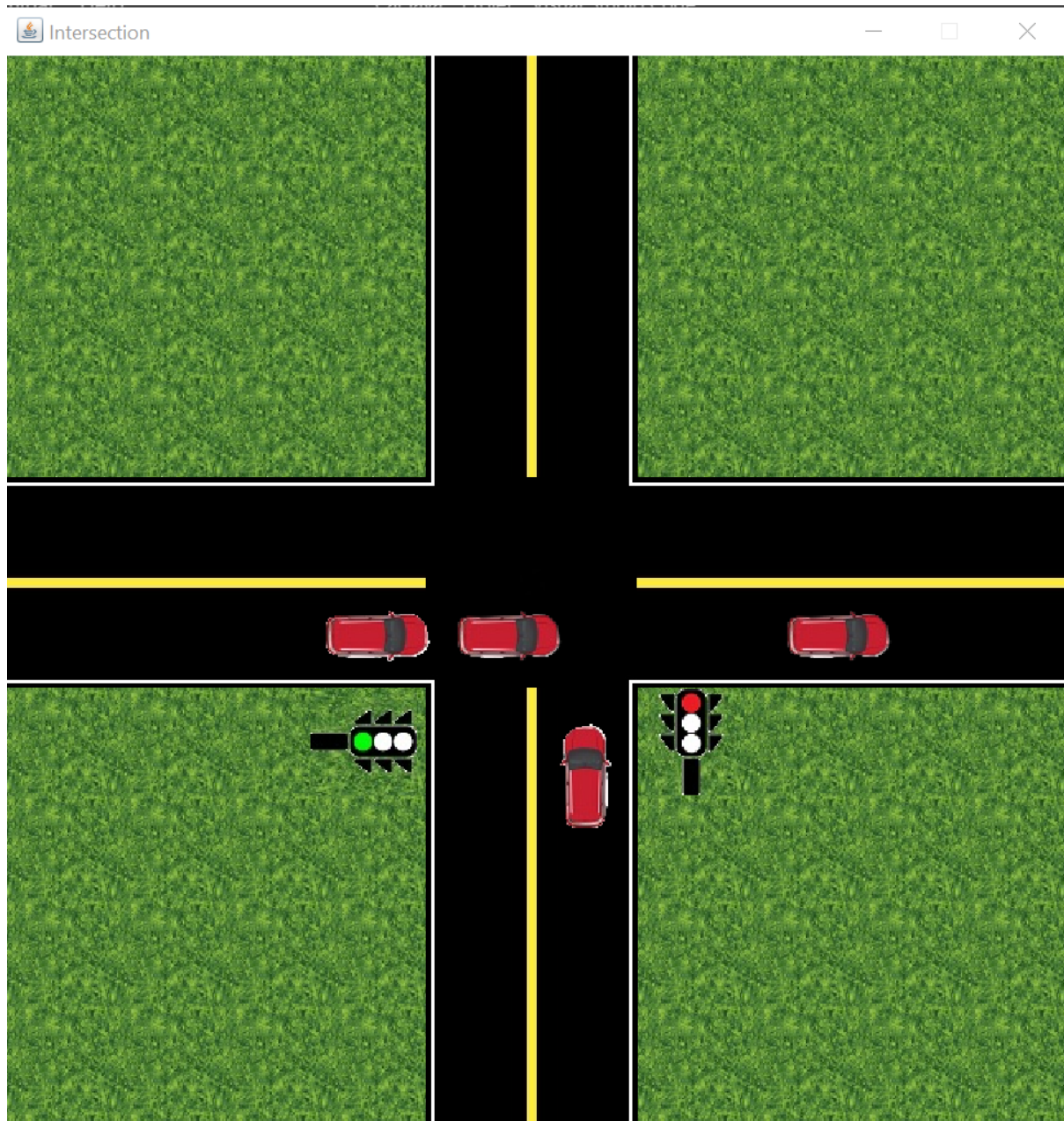
Dans la classe `Car`, le `type` (vertical ou horizontal) nous permet d'assigner directement sans refaire de classe spécifique à l'orientation.

II. Interface graphique

Nous avons une fenêtre qui va venir afficher un panneau. Dans le constructeur du panneau, nous avons défini la liste des voitures ainsi que le feu qui va venir piloter cette intersection. Dans la fonction *paintComponent*, nous avons défini les aspects graphiques fixes tels que le fond et les routes.

Pour les éléments graphiques mobiles (`Car` et `Light`), nous les appelons dans *paintComponent* cependant nous allons les rafraîchir directement dans leurs classes avec des fonctions graphiques.

III. Résultats



Extrait de l'interface pour avec passage de voiture.

Nous pouvons voir sur l'image ci-dessous les voitures qui passent dans l'intersection. Nous avons décidé que dès que la voiture sort cela va venir libérer les sémaophores. Par conséquent, nous avons bien une unique voiture qui passe par l'intersection. Nous n'avons pas pu implémenter l'interface avec des voitures qui attendent à la suite.