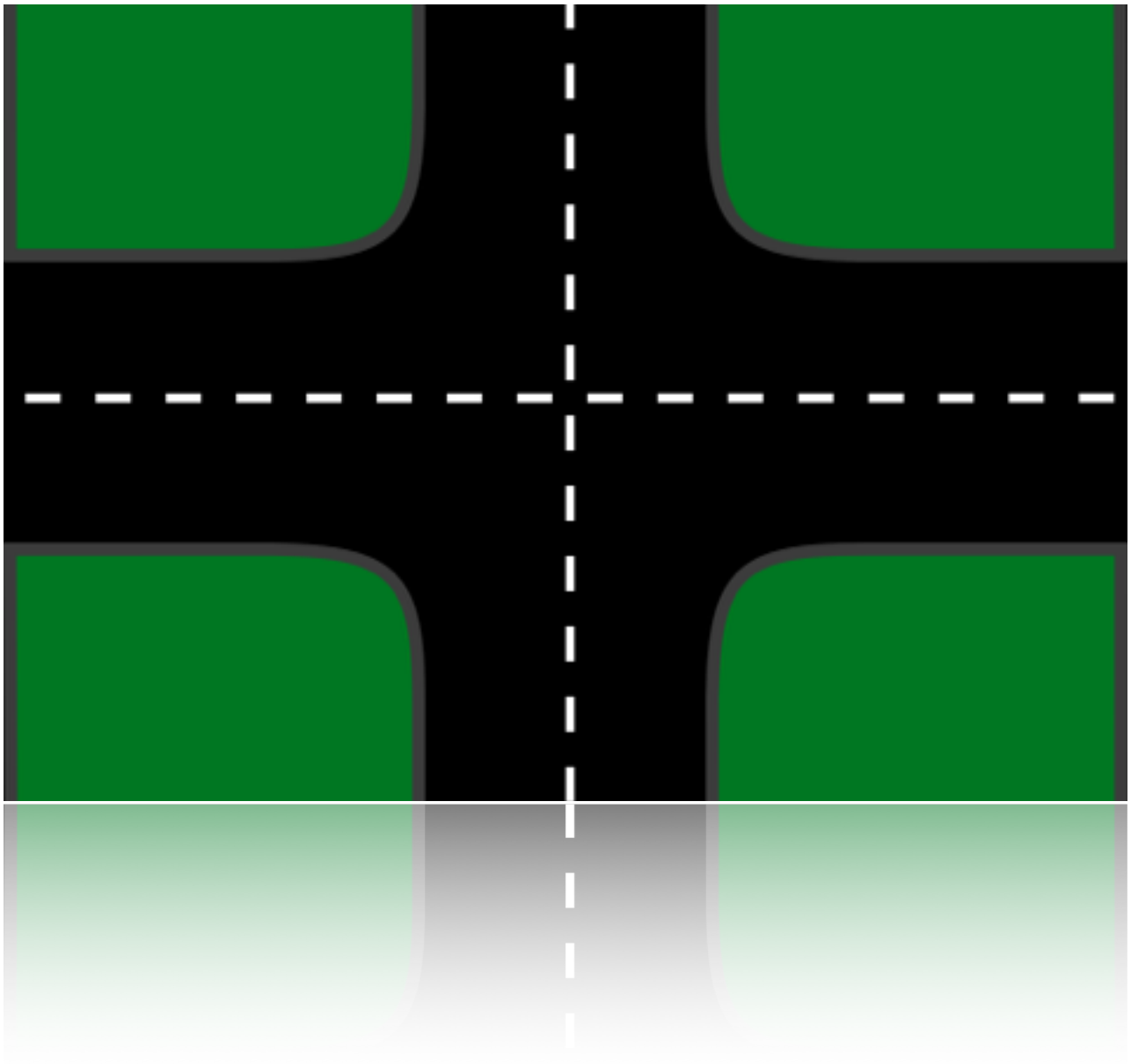

Document descriptif

Crossroad

FAUCHER Alexandre & REZGUI Gada - 27/03/2015



Document descriptif	1
FAUCHER Alexandre & REZGUI Gada - 27/03/2015	1
1. Variables partagées & sémaphores	3
a) Variables partagées	3
b) Sémaphores et mutex	4
Mutex	4
Sémaphores	4
2. Structures de données	4
3. Algorithmes	4

1. Variables partagées & sémaphores

a) Variables partagées

Nous avons 8 variables partagées, qui sont déclarées dans une structure nommée `Shared`, se trouvant dans le fichier `carrefour.h` :

- `numberOfCarsInCrossroads` : nombre de voitures se trouvant dans le carrefour, cette variable partagée permet aux feux de savoir quand ils peuvent passer au vert. En effet, les feux ne peuvent passer au vert que s'il n'y a aucune voiture dans le carrefour.
- `end` : est une variable qui permet de détruire (lorsque l'utilisateur le demande) tout les processus, ainsi que de détacher la variable partagée `shared` qui est reliée à tout les processus.
- `numberOfAllCarCreated` : nombre total de voitures créées, permet de mettre des indices aux voitures créées. Dans le mode automatique cette variable permet de savoir si nous avons atteint ou non le nombre de voitures maximum que l'utilisateur a choisit.
- `firstRoadLights` : permet de savoir la couleur du feu¹ dans la route principal. Cette variable est utilisée dans la fonction qui change la couleur du feu, chaque feu a besoin de savoir la couleur de l'autre feu pour savoir s'il peut passer au vert ou non.
- `secondRoadLights` : suit le même principe que `firstRoadLights`, sauf que la variable permet de contrôler la gestion du feu se trouvant dans la route secondaire.
- `nbCarWaitingFirstRoadLights` : cette variable partagée et associée à un sémaphore qui permet de contrôler la gestion des voitures lorsque le feu est rouge (vous aurez plus de détail sur ce sujet par la suite). Elle compte le nombre de voitures qui attendent que le feu passe au rouge dans la route principal.
- `nbCarWaitingSecondRoadLights` : suit le même principe que `nbCarWaitingFirstRoadLights`, sauf que la variable compte le nombre de voitures attendant que le feu passe au rouge dans la route secondaire.
- `timeToWaitRoadLights` : indique le temps d'attente pour le passage au rouge du feu sur la seconde voie ($2 * \text{timeToWaitRoadLights}$, pour la voie principale) . Dans le mode automatique, cette variable est modifiée après que les deux processus permettant la gestion des feux soient créés. Nous aurions pu faire une variable globale et créer les processus permettant la gestion des feux avant de modifier la variable, me diriez vous

¹ Les couleurs du feu sont 2 constantes :

❖ `RED` = 0

❖ `GREEN` = 1

sans doute, mais nous avons choisis de faire de cette manière pour avoir une plus grande flexibilité. En effet, si l'utilisateur décide de changer le temps d'attente aux feux en plein milieu du programme (dans l'interface graphique par exemple), cela est tout à fait possible.

b) Sémaphores et mutex

Nous avons 4 mutex et deux sémaphores.

Mutex

Les variables partagées qui sont modifiées par les processus « cars » sont protégées, car même si les voitures ne sont pas toutes générées en même temps, les fonctions $P()$ et $usleep()$ peuvent altérer leurs comportements et faire en sorte que 2 processus utilisent les mêmes variables partagées à un même instant.

- `crossroadMutex` : est un mutex qui protège la variable partagée `numberOfCarsInCrossroads`, qui est incrémenter lorsque les voitures traversent le carrefour et décrémenter lorsqu'elles quittent le carrefour.
- `nbCarWaitingFirstRoadLightsMutex` : protège `nbCarWaitingFirstRoadLights`.
- `nbCarWaitingSecondRoadLightsMutex` : protège `nbCarWaitingSecondRoadLights`.
- `numberOfAllCarCreatedMutex` : protège `numberOfAllCarCreated`, qui est utilisée lors de la création des voitures.

Les autres variables partagées ne sont pas protégées car elles sont uniquement modifiées par un seul processus. Par exemples, les variables `end` et `timeToWaitRoadLights` ne sont modifiées que dans le processus père. Les autres processus les manipulent mais ne modifient pas leurs valeurs.

Sémaphores

Les 2 sémaphores permettent de réguler la circulation du carrefour.

2. Structures de données

3. Algorithmes