

Simulation Numérique : Taxis

Vincent

October 4, 2013

Reflexion Classes + Variables

Ville : Rayon R

Centrale Taxis : N Taxis

Centre de la ville ?

Taxis : 2 Usagers

Vitesse, Position

Usagers : Position Initiale

Destination

Temps d'attente

Variables Aléatoires (de base) :

X_1 : ΔT Apparition entre d'un nouveau clients

X_2 : ΔT Disparition Client \Leftrightarrow Temps d'Insatisfaction

X_3 : Position Initiale Client

X_4 : Destination du Client

Cycle Jour/nuit : induit deux gaussiennes de probabilités pour l'apparition des clients (heures de pointes)

Ville divisée en quartier ? (Habitations, Commerces/Amusements, Travail/Bureau)

Taille de la ville (R) ? Vitesse des taxis (vt) ?

R en kilometre, vt en km.h^{-1}

R environ 5km, vt environ 50km.h^{-1} , 1 seconde dans la réalité environ 5 minutes (12s/h)

Client choisit destination :

Destination coordonnées polaires : θ loi uniforme et r loi normale

Destination en fonction de l'heure \Rightarrow après l'heure de pointe du soir, plus grande

probabilité d'aller "s'amuser"

Taxis ralentissent en prenant des usagers ?

Ajout Carburant/Station Essence

Pour 1 client, 1 seul taxis bouge

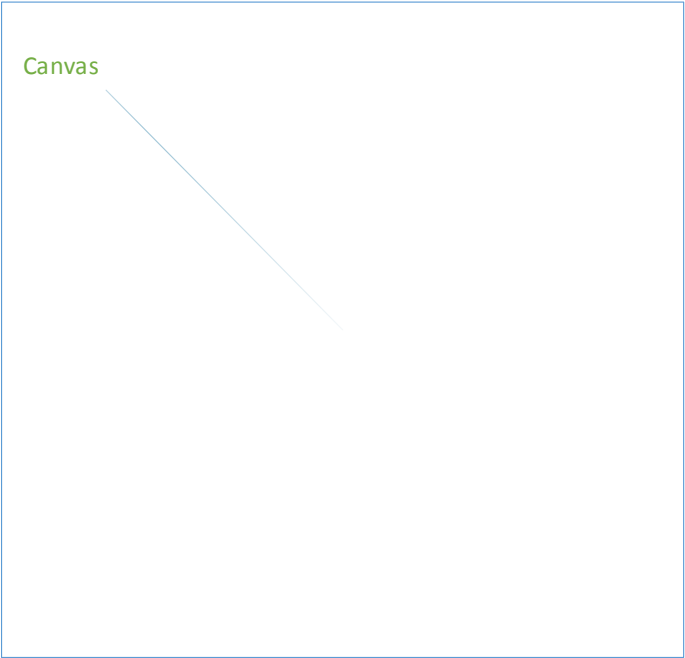
Pages suivantes :

Design Appli

Fonctionnement Appli

Fonctionnement "Prise en charge d'un client par un Taxis"

Simulation Taxis



Temps IApp :

Variable

<val>

Accélérer

Nombre Taxis

<val>

Variable

Nombre Clients

<val>

Variable

En Cours

<val>

Variable

En Taxis

<val>

Variable

Traités

<val>

Variable

Perdus

<val>

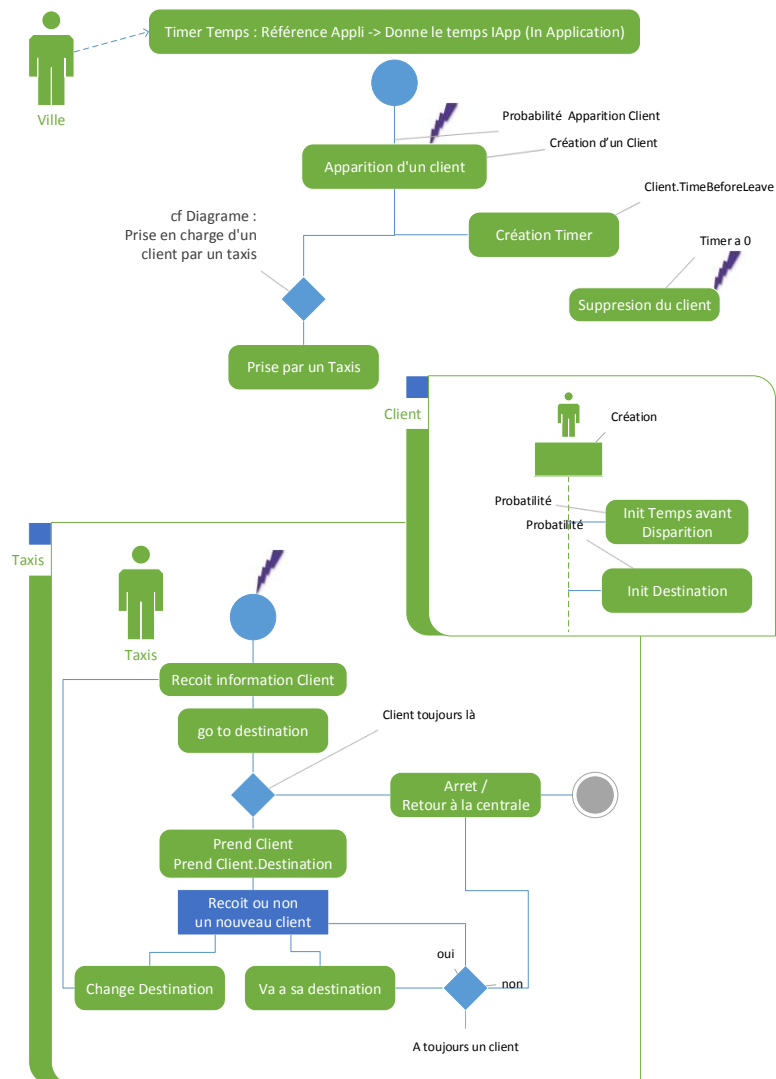
Variable

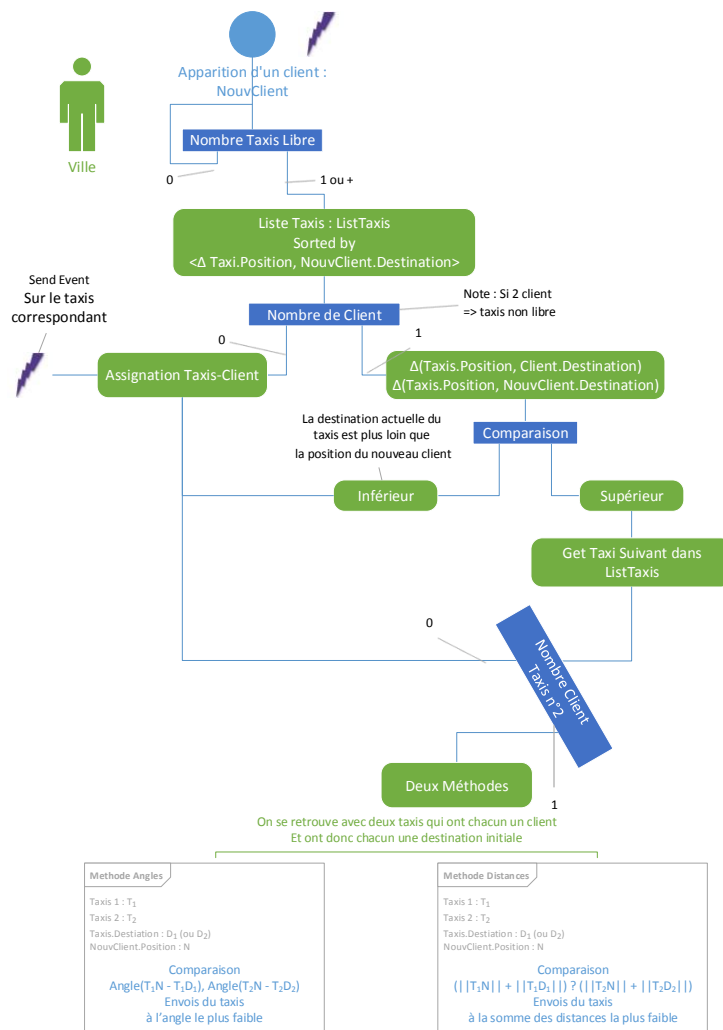
Taille Ville

<val>

Variable

Variable





<<Interface>>
IAleaGen

+bool ClientAppears()
+Point PosnitNewClient()
+Point DestinationNewClient()
+TimeSpan ClientDisappears()

Ville

+ManageNewClient
-List<Taxi> m_Taxis
-double m_TailleVille
-Timer m_CurrentTime
-double m_RatioTime
-List<Client> m_ClientInCity

Est-ce qu'un client est pris en charge

Ratio : temps lapp par seconde réelle

Client

-Point m_Destination { get; }
-TimeSpan m_LifeTime { get; }

Taxis

-Client m_Client1 { get; set; }
-Client m_Client2 { get; set; }
-Point m_Destination { get; }
-Point m_Position { get; }
-double m_Speed { get; }