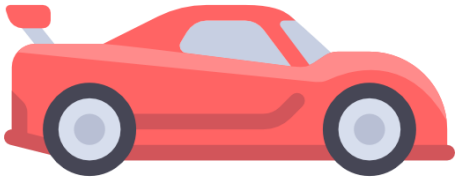
Finite State Machines

Rémy Kaloustian

SI4 – G3



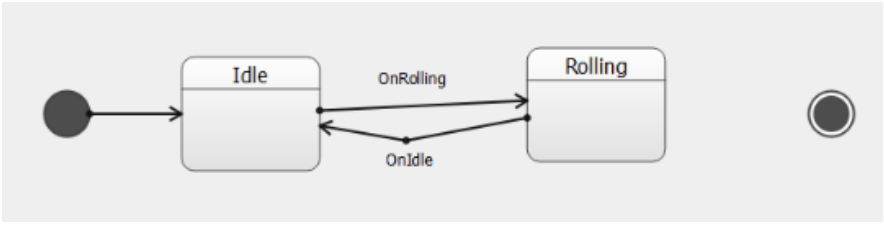
1..Description du domaine possible

Le domaine d'application serait possiblement l'embarqué. Pour plus de précisions, nous dirons une voiture télécommandée pour enfants.

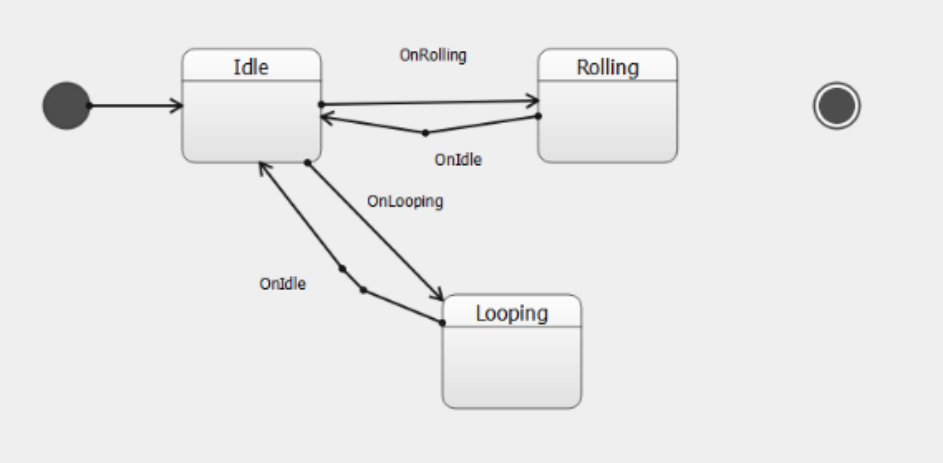
J'ai choisi ce domaine tout d'abord car il me permettait de voir comment ce que je vois à Polytech pourrait servir dans une application embarquée, mais aussi car il ne présente pas de contraintes au niveau des améliorations que je pourrai apporter à mon générateur de code (états parallèles, etc…).

Je me suis basé sur ce cas là pour créer mon générateur de code. De ce fait, la machine à état utilisée correspond bien au domaine visé, car elle a été établie précisément pour ce cas-là.

Nous avons tout d'abord une machine à états simple :



Puis une machine à états plus avancée :



NB : Il y a un delay à l'entrée dans looping.

2..Contraintes

Les fonctions suivant un changement d'état ne sont exécutées qu'une fois dans l'état (c'est une machine de Moore).

La fonction exécutée dans l'état doit s'appeler InNomDeLEtat.

Le delay est ajouté sur le onentry->send de l'état correspondant.

Compilation : g++ -std=c++11

3..Amélioration

Mon but était de proposer une machine à état concrète qui puisse être utilisable dans un vrai contexte, sans avoir à supposer certains paramètres. Cette amélioration devait avoir du sens pour moi. J'ai pensé alors à la possibilité de faire faire un looping à la voiture télécommandée. Toutefois, il se peut qu'on ait besoin d'utiliser des capteurs pour vérifier qu'aucun élément ne va bloquer le looping. Il serait donc pratique d'attendre le temps que les capteurs détectent l'environnement, avant de rentrer dans le comportement de l'état looping. D'où l'utilisation d'un delay avant d'exécuter le « vrai » code de Looping.