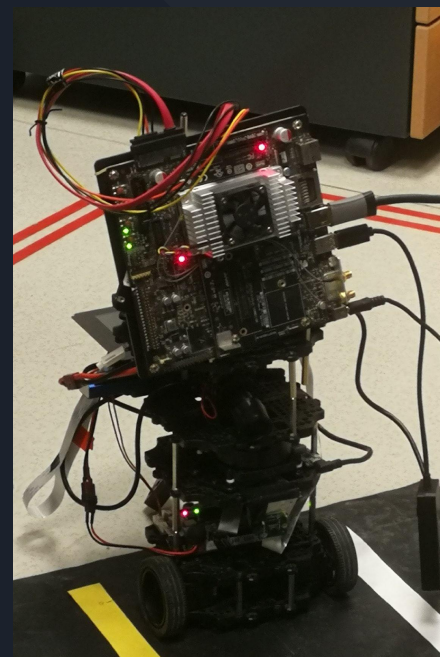


# Projet ANDROIDE

**Amélioration de la  
conduite d'un robot  
roulant intelligent et  
autonome**



## Equipe :

Jérémy Dufourmantelle - 21104331  
Ethan Abitbol - 3804139  
Elias Bendjaballah - 3700088  
Jules Cassan - 3810883

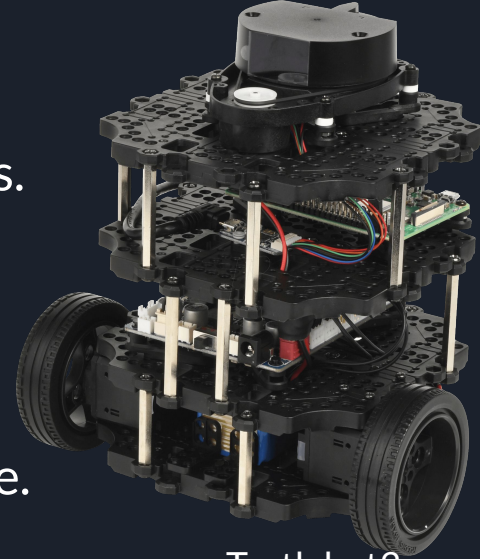
**24 Mai 2022**

## Encadrants :

Jean Michel Illié  
Farouk Meddah

# Introduction

- Repousser au maximum les interventions humaines.
- TurtleBot3
- Livrer des médicaments à des pharmacies réparties dans une ville dont on connaît la topologie.
- Améliorer la conduite -> Prise de virages
- Circuit + ensemble des virages = une suite de courbes de Bézier.



Turtlebot3

Comment améliorer la conduite d'un robot roulant intelligent et autonome par la méthode des courbes de Bézier ?

# Sommaire

- 1) Courbes de Bézier
- 2) Etude de la faisabilité d'une approche déterministe de la conduite autonome du véhicule
- 3) Conception d'un simulateur 2D
- 4) Environnement 3D sur GAZEBO

# Courbes de Bézier

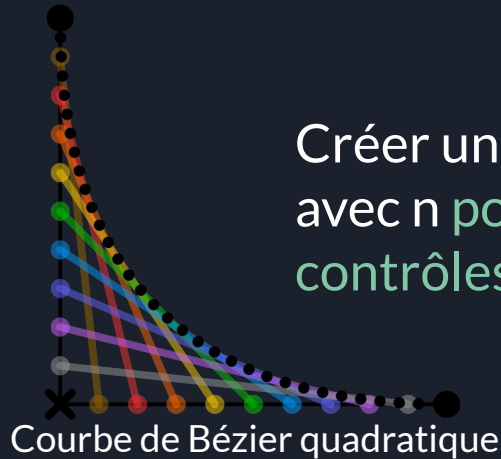
## Un peu d'histoire :

- Paul de Casteljaeu en 1959 pour Citroën
- Pierre Bézier en 1962 pour Renault

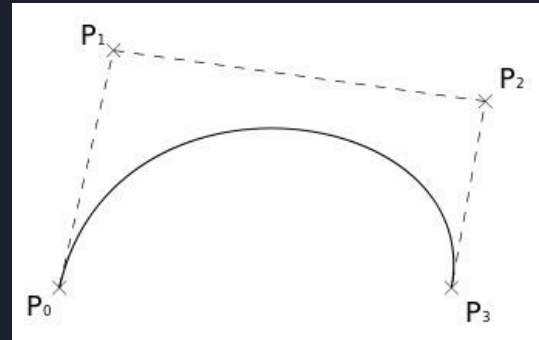
-> Courbes polynomiales paramétriques



Pierre Bézier

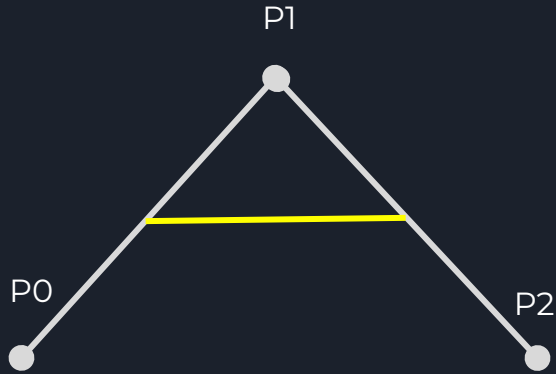


Créer une courbe  
avec  $n$  points de  
contrôles

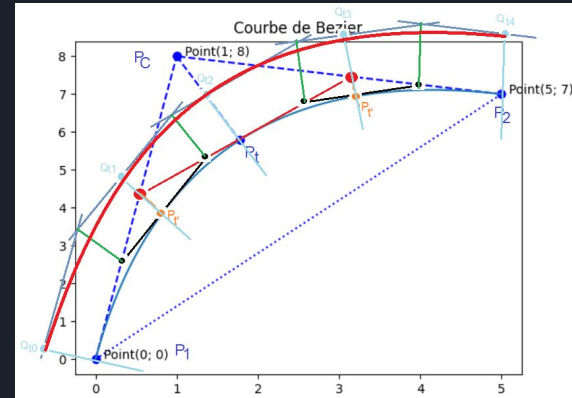
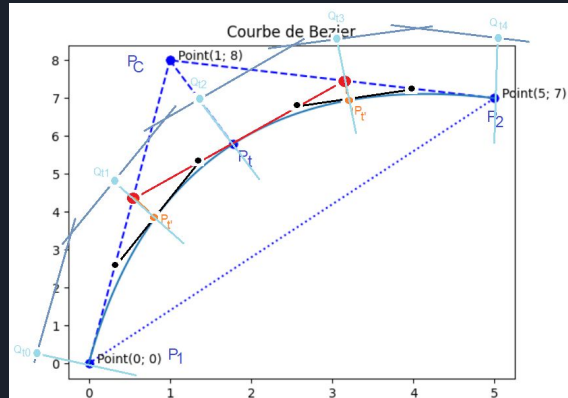
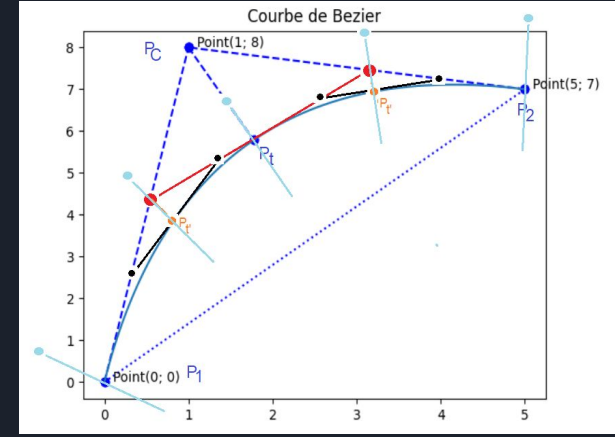
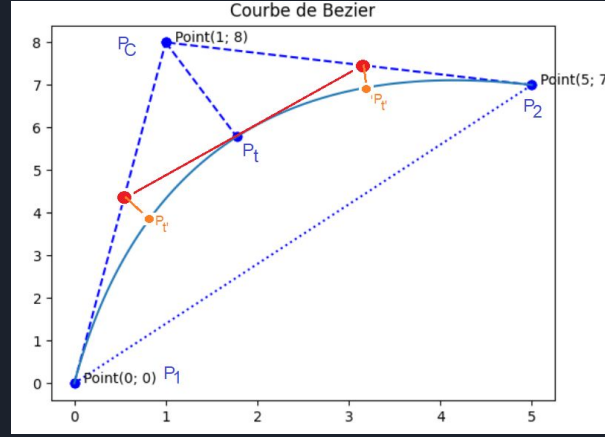
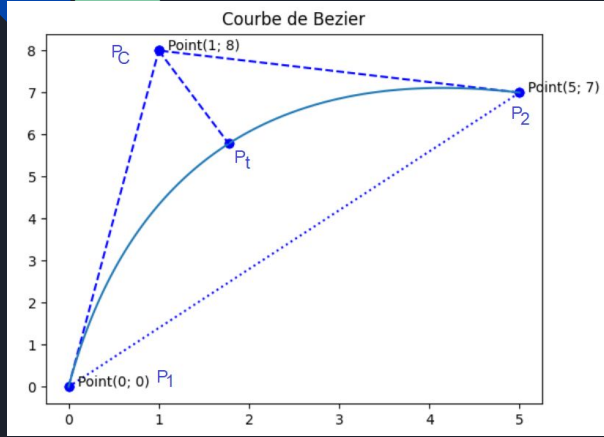


Courbe de Bézier cubique

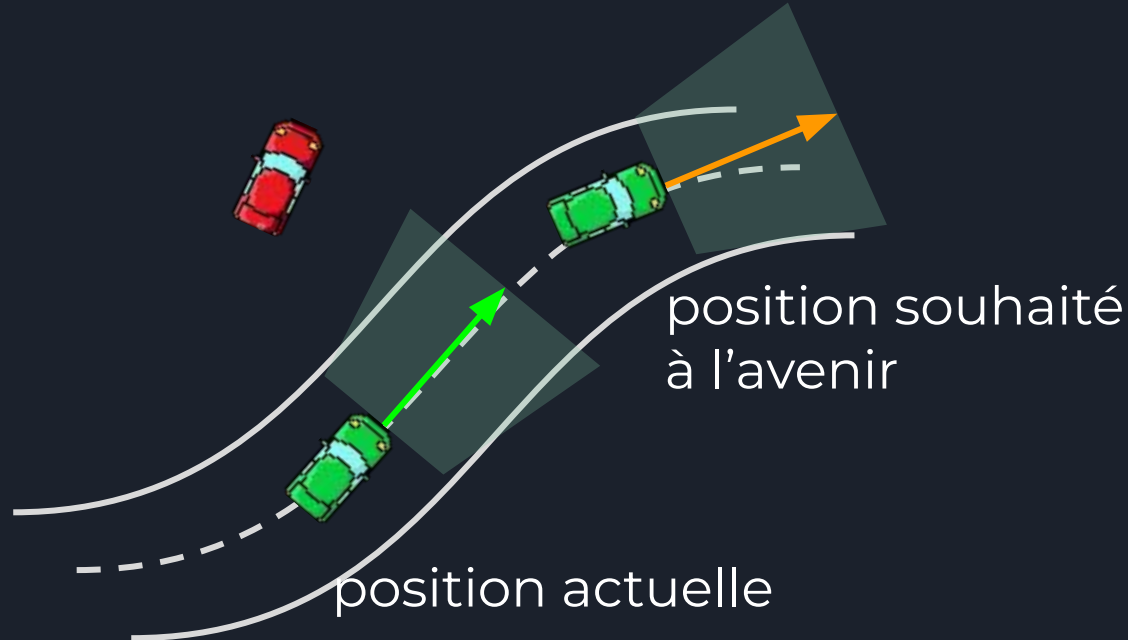
- Méthode de barycentrage successif



# ● Méthode par subdivision sélective



# Étude de faisabilité d'une approche déterministe de la conduite autonome du véhicule

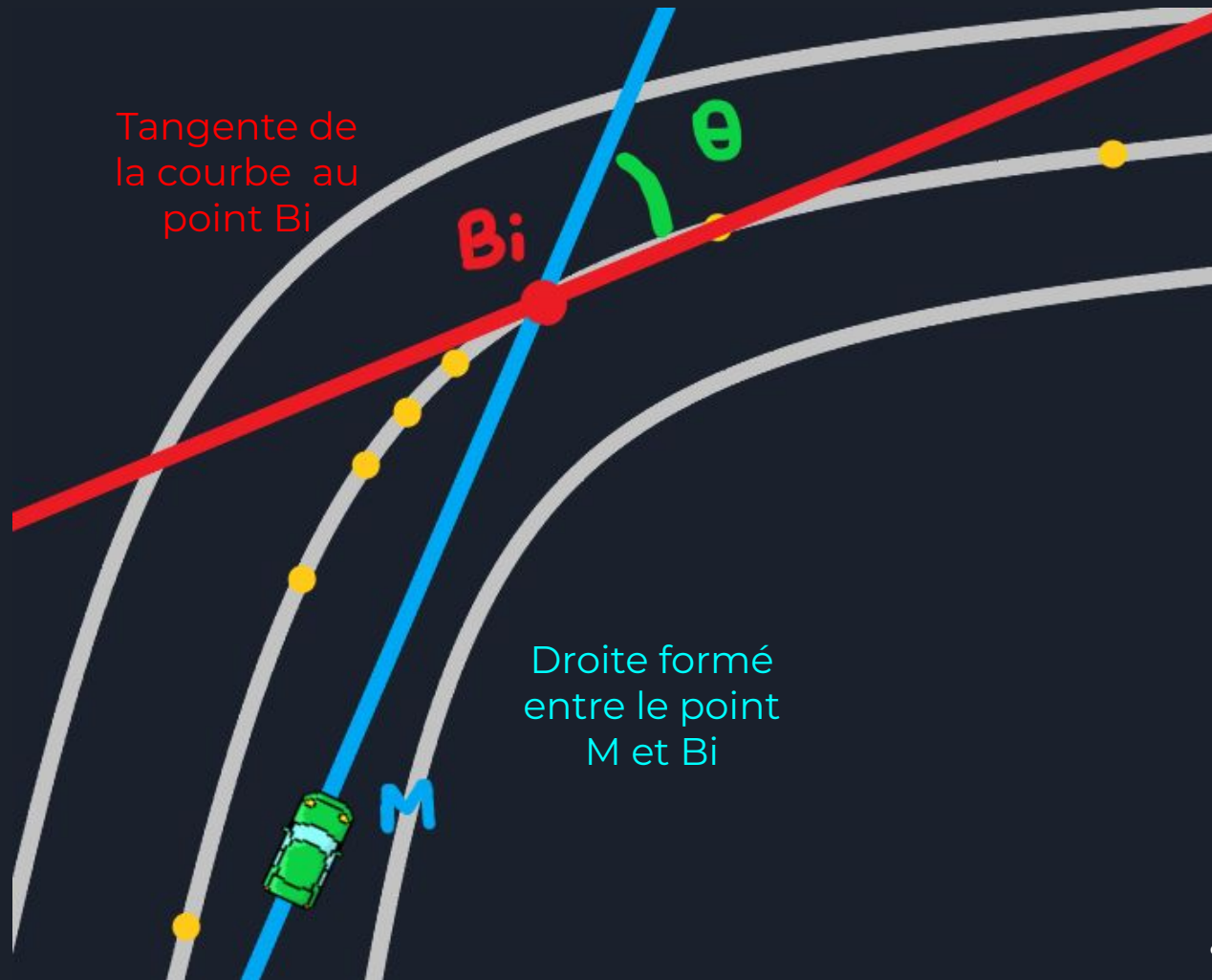


# Représentation de la route

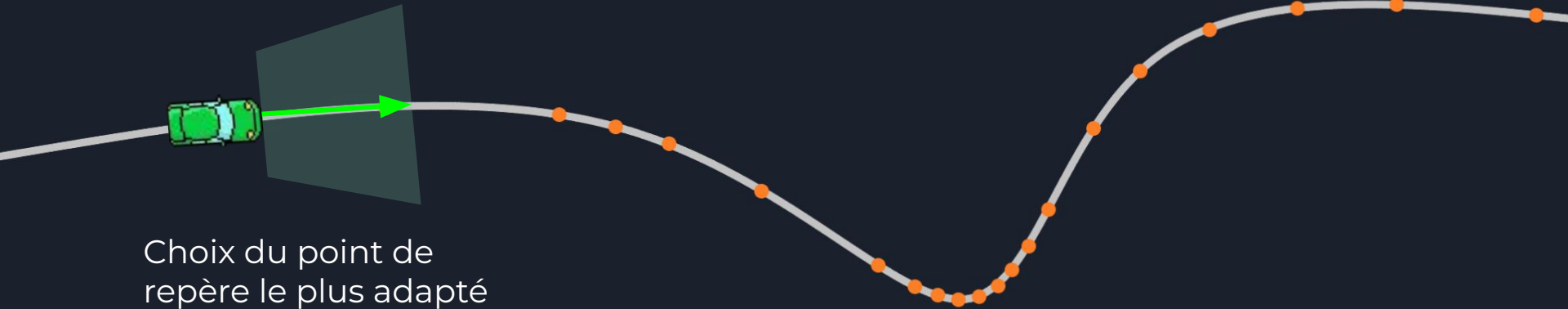




Recherche  
de l'angle  
de  
correction  
de la  
trajectoire



# Algorithme Général Déterministe par méthode de Bézier

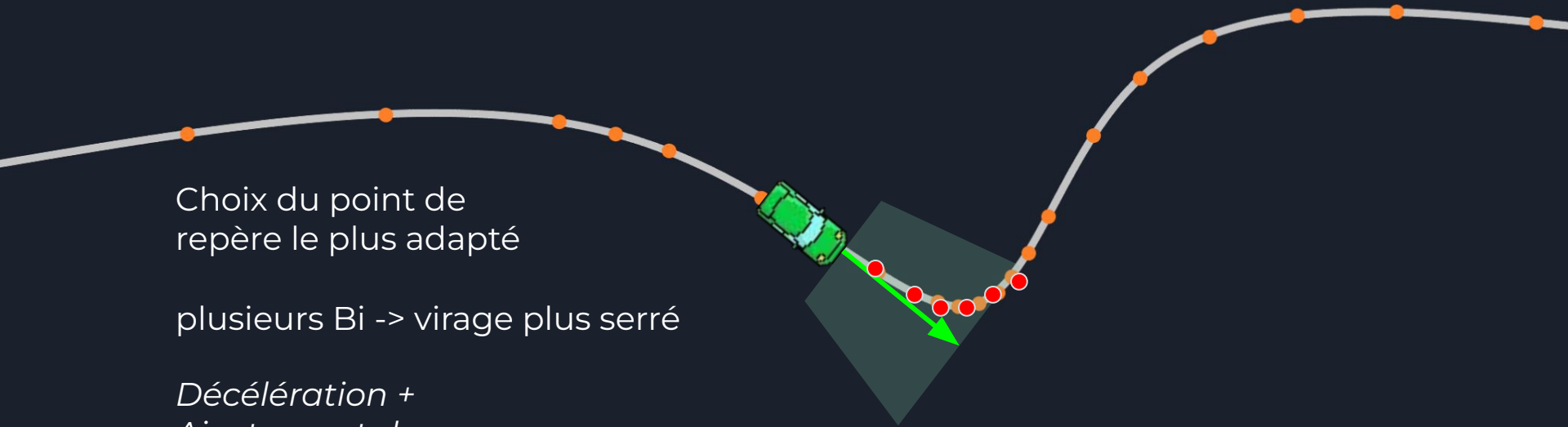


Choix du point de  
repère le plus adapté

pas de virage dans  
l'immédiat

*accélération*

# Algorithme Général Déterministe par méthode de Bézier



Choix du point de  
repère le plus adapté

plusieurs Bi -> virage plus serré

*Décélération +  
Ajustement de  
la direction*

# Conception d'un simulateur 2D

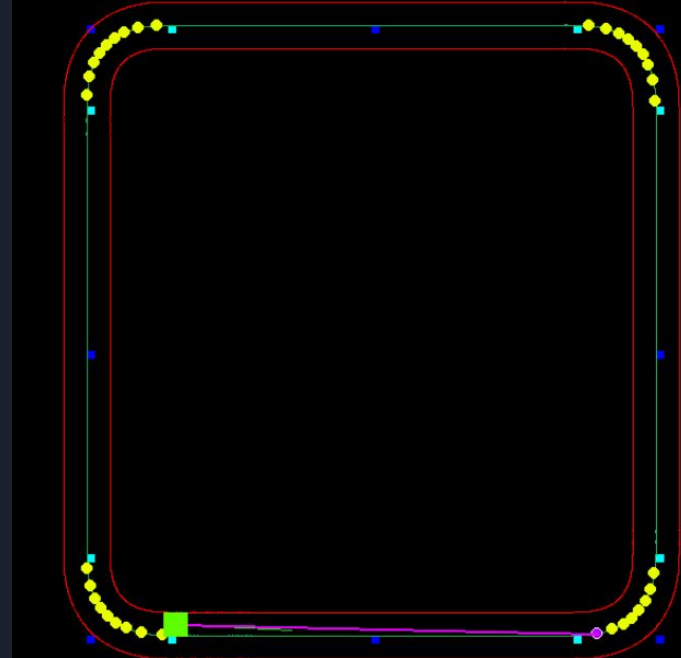
Pourquoi ?

fonctionnalités

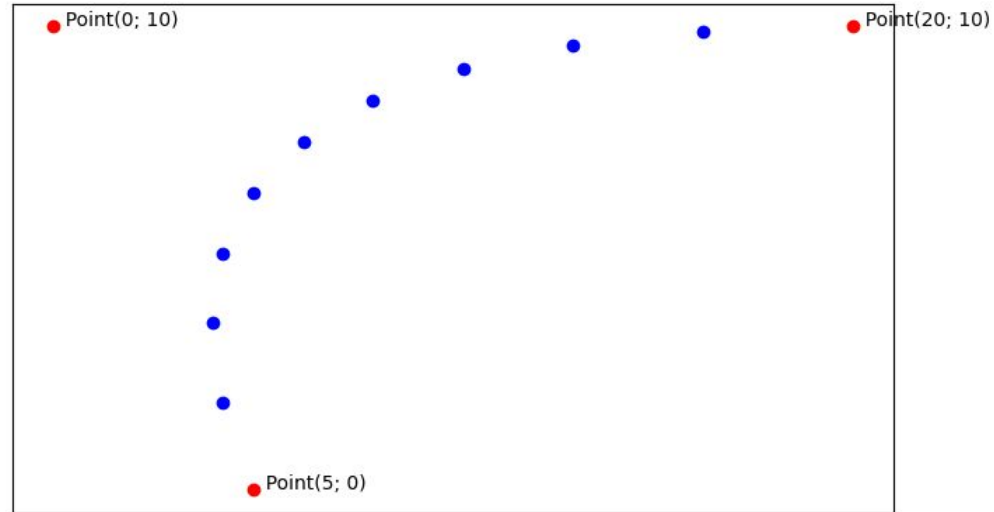
Architecture et choix  
d'implémentation

Connection avec l'état de l'art

Taille circuit : 1014.36 px  
Position robot : (136.487, 638.205)  
Position curseur : (451, 466)



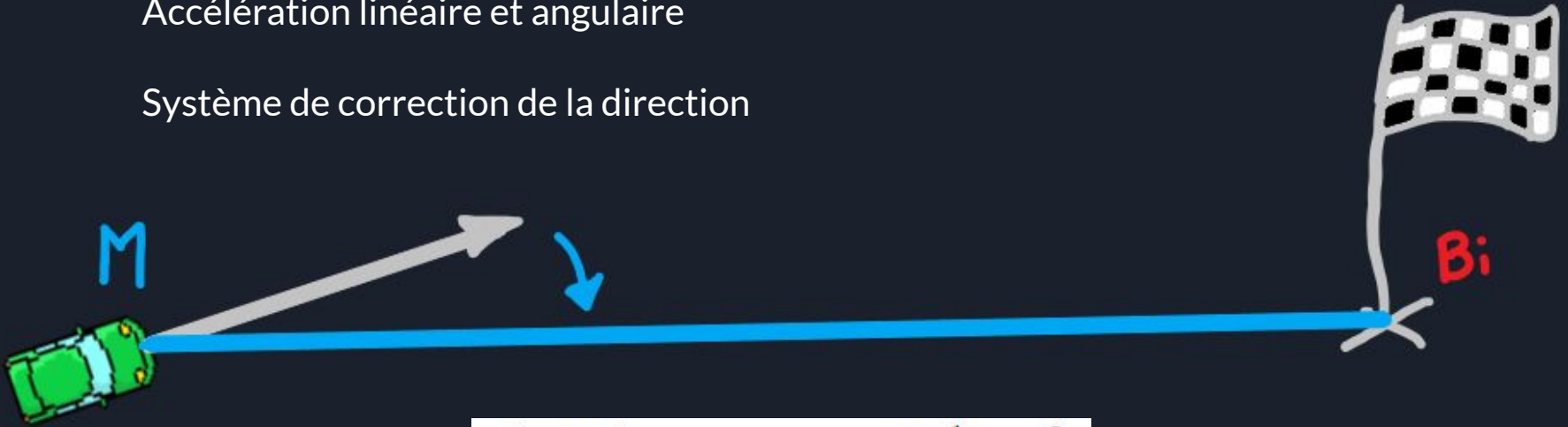
# Construction de la route



# Vitesse et rotation du véhicule

Accélération linéaire et angulaire

Système de correction de la direction



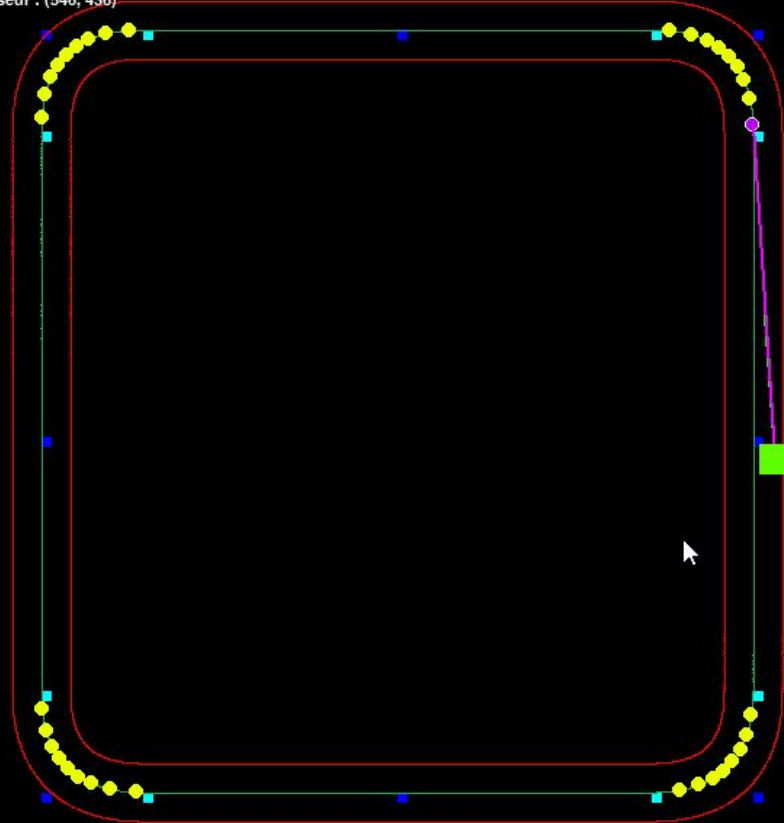
$$x(t+1) = x_t + v_{x_t} * t + \frac{1}{2}a * t^2$$

$$\theta(t+1) = \theta_t + \omega_{\theta_t} * t + \frac{1}{2}\alpha * t^2$$

# Démonstration du simulateur

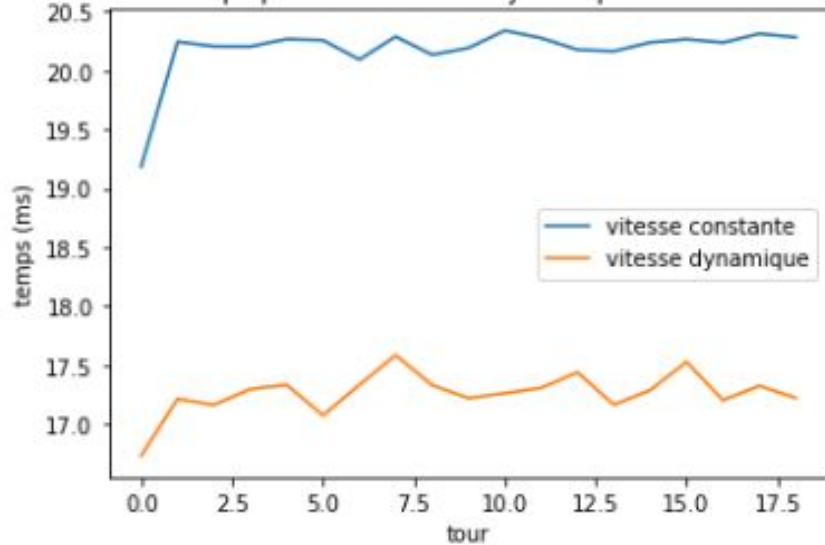


Taille circuit : 1914.36 px  
Position robot : (599.838, 371.985)  
Position curseur : (546, 436)



# Résultats et limites

Mesure de temps par tour : vitesse dynamique vs vitesse constante



Deux approches :

- Vitesse Dynamique
- Vitesse Constante

Amélioration :

- Méthode d'apprentissage pour les paramètres



# Environnement 3D sur Gazebo



- Mise en commun des éléments et observations effectuées sur l'environnement 2D
- Réflexion sur le portage en 3D + focus sur les éléments importants
- Documentation et apprentissage de l'API Gazebo
- Extension du modèle -> Env de simulation Custom

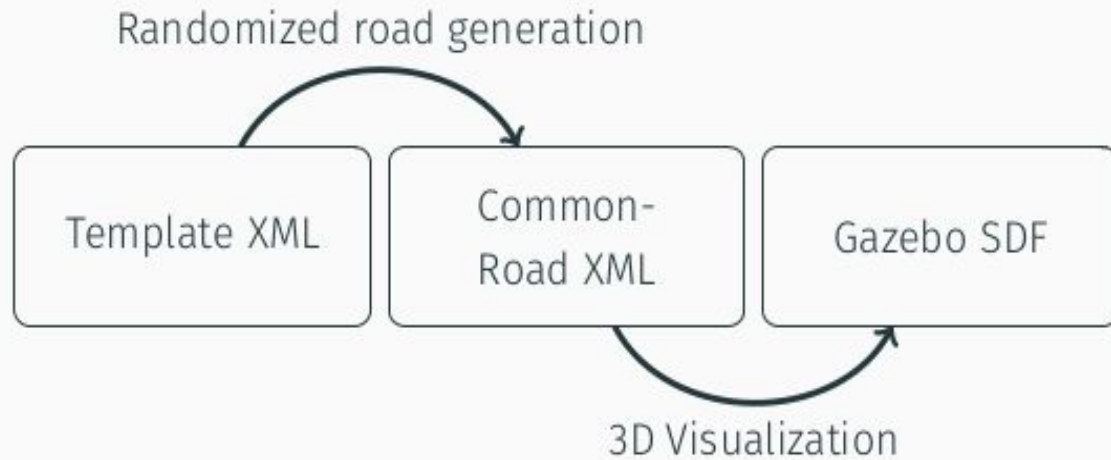
- Environnement de simulation mono/multi-robots
- Open Source
- Moteur 3D basé sur le moteur OGRE
- Rendu des caractéristiques physiques par divers moteurs physiques :
  - ODE (Open Dynamics Engine)
  - Bullet Physics
  - Simbody (multibody physics API)
  - Dart (Dynamic Animation and Robotics toolkit)
- Forte synergie avec ROS
- Possibilité d'augmenter la simulation par ajout de Plugin C++



# Génération randomisée, semi-randomisée et stricte

- Base XML
- Format très similaire au Simulation Description Format (.sdf)
- Etude de la spécification : <http://sdformat.org/spec> + [https://classic.gazebo.org/tutorials?tut=build\\_model](https://classic.gazebo.org/tutorials?tut=build_model)
- Identification des éléments clés + concaténation de plusieurs sdf ou génération d'un sdf principal

# Générateur :



# Primitives



Straight roads



Circular arcs



Static obstacles



Traffic signs



Intersections



Cubic Bézier curves



Quadratic Bézier curves



Parking lots



Parking obstacles



Zebra crossings



Blocked areas

# Définition du pattern :

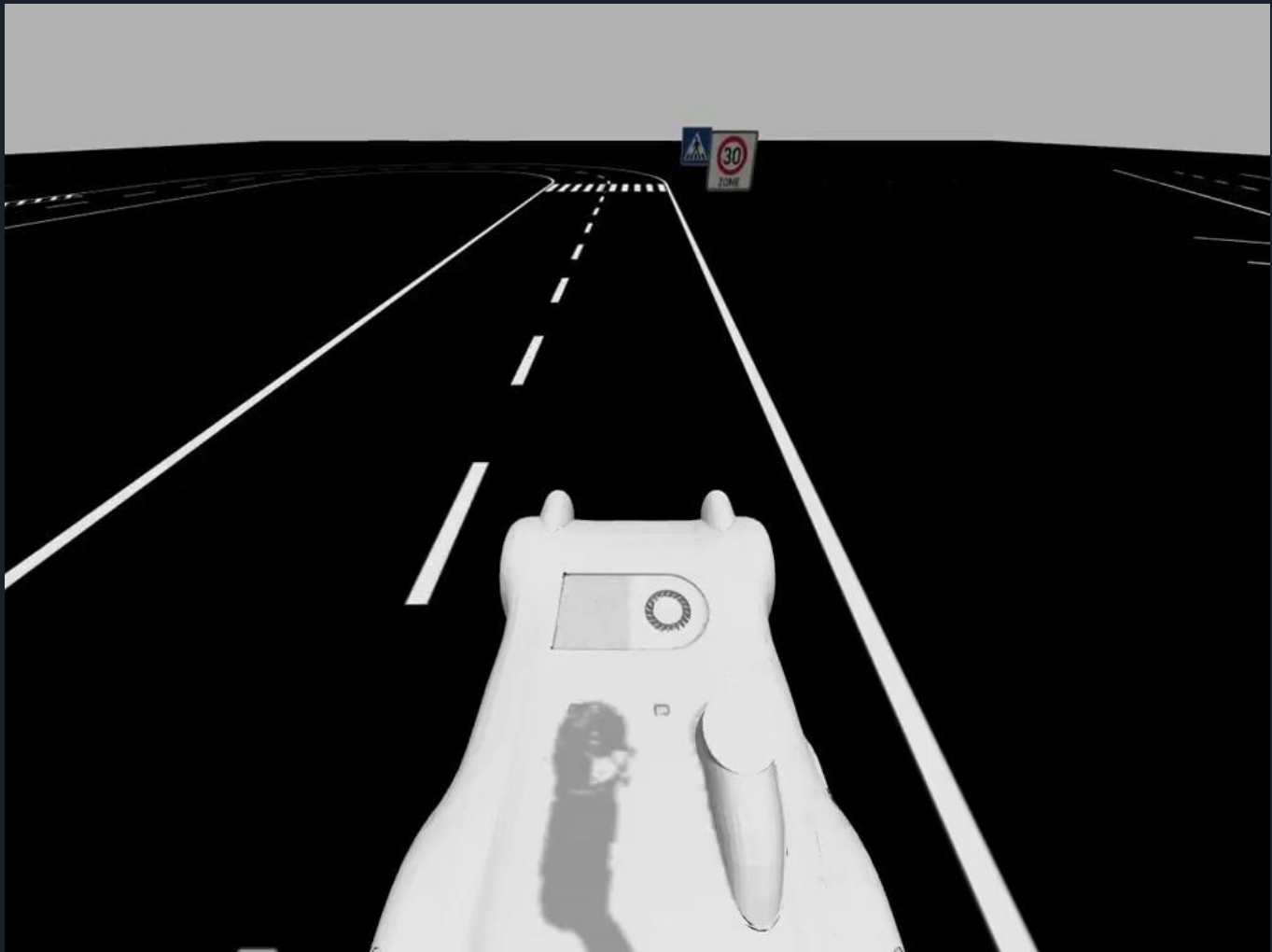
- `<sequence> </sequence>`
- `<optional> </optional>`
- `<select> (<case></case>)* </select>`
- `<repeat> </repeat>`
- `<shuffle> </shuffle>`

```
1 <sequence>
2   <trafficSign type="stvo-350-10" />
3   <zebraCrossing length="0.4" />
4 </sequence>
```

```
1 <repeat min="1" max="3">
2   <select>
3     <case w="0.9">
4       <trafficSign type="stvo-350-10" />
5       <line length="0.4" />
6       <zebraCrossing length="0.4" />
7     </case>
8     <case w="1">
9       <trafficSign type="stvo-208" />
10      <line length="0.5" />
11      <blockedArea width="0.15" length="2" />
12    </case>
13  </select>
```



```
1 <shuffle>
2   <sequence>
3     <parkingObstacle width="0.28" length="0.2" />
4     <parkingLot length="0.55" />
5     <parkingObstacle width="0.28" length="0.2" />
6   </sequence>
7   <sequence>
8     <parkingObstacle width="0.28" length="0.2" />
9     <parkingLot length="0.63" />
10    <parkingObstacle width="0.28" length="0.2" />
11  </sequence>
12  <sequence>
13    <parkingObstacle width="0.28" length="0.2" />
14    <parkingLot length="0.70" />
15    <parkingObstacle width="0.28" length="0.2" />
16  </sequence>
17    <parkingObstacle width="0.28" length="0.2" />
18    <parkingObstacle width="0.28" length="0.4" />
19  <sequence>
20    <parkingObstacle width="0.15" length="0.2" />
21    <repeat min="3" max="5">
22      <optional p="0.8">
23        <select>
24          <case w="1"><parkingLot length="0.10" /></case>
25          <case w="1"><parkingLot length="0.20" /></case>
```







# Conclusion

- Projet très enrichissant :
  - Personnellement (gestion, recherche..)
  - Notion courbe de Bézier,
  - Conception d'une approche déterministe,
  - Implémentation de notre propre simulateur,
  - maîtriser un simulateur 3D,
- Améliorations possibles:
  - Obtenir les valeurs critiques par apprentissage,
  - Profil de conduite plus avancé,
- Projet pas totalement complet:
  - Implémentation sur Robot réel.

*Merci pour votre attention*

*Avez-vous des questions?*

