LIVRABLE F : Prototype I et rétroaction du client

Groupe : F03 - FF32 - Les Chèvres

Caleb Cauchon

Noura Coulibaly

Cedric Gervais

Ica Ishimwe

Ethan Leroux

Dimitri Ouédraogo

GNG 1503 : Introduction au génie de la conception

Professeur : Emmanuel Boendeu

Résumé

Ce livrable a pour objectif de développer un premier prototype de base pour notre produit, de demander les rétroactions des autres étudiants dans le but d'améliorer le projet, et de faire un plan d'essai (prototypage) détaillé pour notre prochain prototype. Celle-ci servira comme une preuve de concept, de l'analyse critique des composants ainsi que la documentation des divers tests. Le prototype 1 est conçu pour répondre à une série d'objectifs et utilise que des matériaux peu coûteux de la maison afin de valider nos hypothèses et la structure. Il est important que nous soyons efficaces et ingénieux pour maximiser ce que l'on peut tester avec ces ressources limitées.

Table des matières

Résumé..... 2
Table des matières..... 2
1 Introduction..... 3
2- Rétroactroaction client rencontre 2..... 3
2.1 Suggestions du client..... 4
2.1.1 Logiciel..... 4
2.1.2 Ligne d'arrivée..... 4
2.1.3 Coque de couleur..... 4
2.2 comment les commentaires sont utilisés pour éclairer le choix du prototype 2..... 4
3 Développement du prototype..... 4
3.1 sous-système 1 : ligne d'arrivée..... 4
3.2 sous-système 2 : coque..... 5
3.3 sous-système 3: Code..... 6
3.3.1 résultat du test..... 7
3.3.2 Analyse des résultat..... 7
4 Rétroaction utilisateur..... 8
5 mise à jour des spécifications cible..... 8
6- Plan de prototypage du prototype 2..... 9
7- Conclusion..... 10

1- Introduction

La conception et le prototypage sont des étapes clés dans tout projet d'ingénierie, car ils permettent de valider les concepts, d'identifier les défis techniques et d'améliorer la solution avant sa mise en œuvre finale. Dans ce livrable, nous présentons le développement de notre premier prototype, basé sur les concepts préliminaires établis précédemment. Ce document détaille la rétroaction des clients, les tests effectués pour valider les hypothèses critiques, et les ajustements apportés à la conception pour optimiser la solution.

Nous commençons par décrire les retours des clients et leur impact sur les choix de conception. Ensuite, nous présentons le prototype, en mettant l'accent sur ses objectifs, les matériaux utilisés et les résultats des tests. Enfin, nous proposons un plan d'essai détaillé pour le deuxième prototype, visant à réduire les risques et à garantir la réussite du projet. Ce livrable marque une étape cruciale dans notre processus de conception, en nous permettant de concrétiser nos idées et de préparer les étapes suivantes avec une meilleure compréhension des défis et des opportunités.

2- Rétroactroaction client rencontre 2

2.1 Suggestions du client

2.1.1 Logiciel

- Pas de contrainte pour le langage de programmation.
- Simple qui pourrait être amélioré plus tard.
- Ajouter des commentaires pour chaque ligne de code si possible.

2.1.2 Ligne d'arrivée

- La ligne d'arrivée est suffisamment large (largeur de 1 m).
- Utilisation de capteurs de couleur : bonne idée.
- Meilleure alternative : supprimer le système de portes.

2.1.3 Coque de couleur

- Utiliser des couleurs sur les voitures et les portes pour faciliter l'identification.
- Utiliser une machine de **vacuum forming** au makerspace pour mouler du plastique sur la voiture.
- Coque interchangeable sur les voitures (ex. : plusieurs couleurs pour différentes écoles).
- Aimant envisageable, mais un système à clipser la coque serait plus efficace.

2.2 comment les commentaires sont utilisés pour éclairer le choix du prototype 2

Basés sur ces énoncés du client, nous avons déterminé que le capteur de couleur était assez éfficace. Car il voulait que le prototype soit conçu avec un logiciel flexible, sans contrainte de langage de programmation, tout en restant simple et évolutif. L'ajout de commentaires détaillés dans le code est recommandé afin de faciliter sa compréhension et son amélioration future.
Concernant la ligne d'arrivée, une largeur d'un mètre est suffisante, et l'utilisation de capteurs de couleur est privilégiée. Pour améliorer l'identification des voitures, celles-ci seront dotées de coques colorées, assorties aux portes si nécessaire. Ces coques seront fabriquées via une machine de **vacuum forming** au makerspace au lieu de l'impression 3D qui avait été proposée par notre équipe car cette technique est plus simple et plus précise. Aussi un système de fixation par clip est préférable aux aimants proposés par notre équipe pour garantir une meilleure tenue.

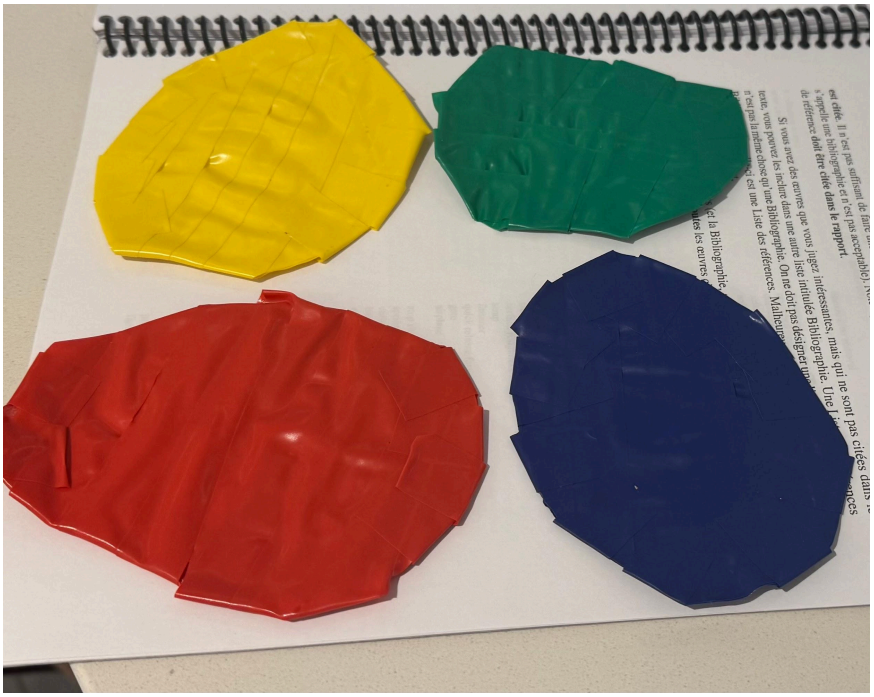
3 Développement du prototype

3.1 sous-système 1 : ligne d'arrivée

Ligne d'arrivée:

Troue pour le capteur:

3.2 sous-système 2 : coque

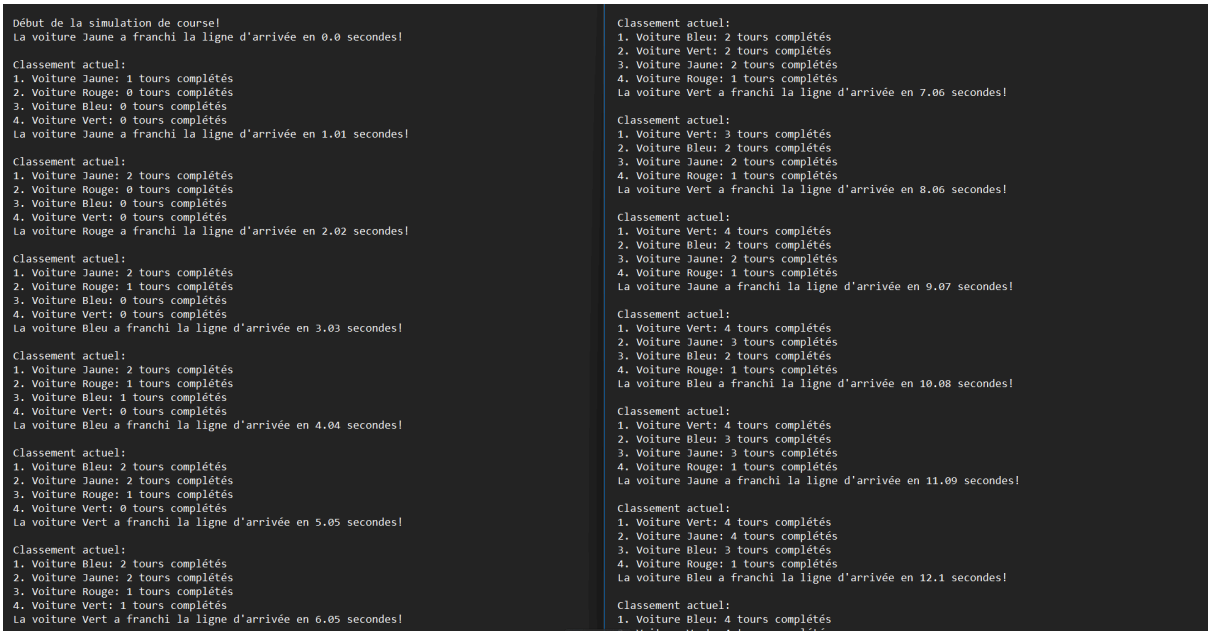


3.3 sous-système 3: Code

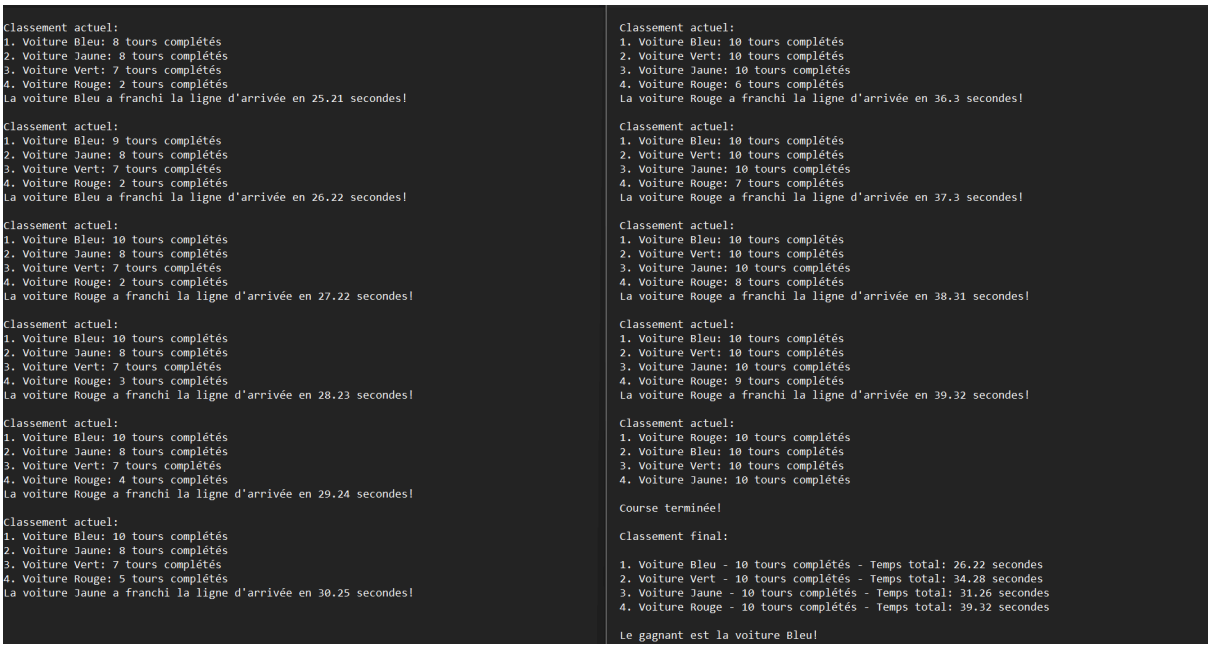
Proto1 > ...

```python
import time
import random
from collections import defaultdict

# Parameters
NOMBRE_VOITURES = 4
TOURS_A_FAIRE = 10

tours_completes = defaultdict(int)  # n laps completé
voitures = ["Rouge", "Bleu", "Vert", "Jaune"]  # couleurs des voitures
temps_par_voiture = {voiture: [] for voiture in voitures}  # Stores les temps des voitures

def simulation_capteur():
    # Simulation des voitures qui passe
    return random.choice([v for v in voitures if tours_completes[v] < TOURS_A_FAIRE])

def afficher_classement():
    # Va nous donner le classement des voitures
    classement = sorted(tours_completes.items(), key=lambda x: x[1], reverse=True)
    print("\nClassement actuel:")
    for i, (voiture, tours) in enumerate(classement, 1):
        print(f"{i}. Voiture {voiture}: {tours} tours complétés")

def course():
    # Simule une course
    print("Début de la simulation de course!\n")
    temps_debut = time.time()
    voitures_en_course = set(voitures)

    while (voitures_en_course):
        voiture_detectee = simulation_capteur()
        tours_completes[voiture_detectee] += 1
        lap_time = round(time.time() - temps_debut, 2)
        temps_par_voiture[voiture_detectee].append(lap_time)
        print(f"La voiture {voiture_detectee} a franchi la ligne d'arrivée en {lap_time} secondes!")
        afficher_classement()

        if tours_completes[voiture_detectee] >= TOURS_A_FAIRE:
            voitures_en_course.remove(voiture_detectee)

        time.sleep(1)  # Simule une pause entre le passage des voitures

    print("\nCourse terminée!")
    classement_final = sorted(tours_completes.items(), key=lambda x: x[1], reverse=True)
    print("\nClassement final:")
    for i, (voiture, tours) in enumerate(classement_final, 1):
        print(f"{i}. Voiture {voiture} - {tours} tours complétés - Temps total: {temps_par_voiture[voiture][-1]} secondes")

    print(f"\nLe gagnant est la voiture {classement_final[0][0]}!")

if __name__ == "__main__":
    course()
```

3.3.1 résultat du test



```
Début de la simulation de course!
La voiture Jaune a franchi la ligne d'arrivée en 0.0 secondes!

Classement actuel:
1. Voiture Jaune: 1 tours complétés
2. Voiture Rouge: 0 tours complétés
3. Voiture Bleu: 0 tours complétés
4. Voiture Vert: 0 tours complétés
La voiture Jaune a franchi la ligne d'arrivée en 1.01 secondes!

Classement actuel:
1. Voiture Jaune: 2 tours complétés
2. Voiture Rouge: 0 tours complétés
3. Voiture Bleu: 0 tours complétés
4. Voiture Vert: 0 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 2.02 secondes!

Classement actuel:
1. Voiture Jaune: 2 tours complétés
2. Voiture Rouge: 1 tours complétés
3. Voiture Bleu: 0 tours complétés
4. Voiture Vert: 0 tours complétés
La voiture Bleu a franchi la ligne d'arrivée en 3.03 secondes!

Classement actuel:
1. Voiture Jaune: 2 tours complétés
2. Voiture Rouge: 1 tours complétés
3. Voiture Bleu: 1 tours complétés
4. Voiture Vert: 0 tours complétés
La voiture Bleu a franchi la ligne d'arrivée en 4.04 secondes!

Classement actuel:
1. Voiture Bleu: 2 tours complétés
2. Voiture Jaune: 2 tours complétés
3. Voiture Rouge: 1 tours complétés
4. Voiture Vert: 0 tours complétés
La voiture Vert a franchi la ligne d'arrivée en 5.05 secondes!

Classement actuel:
1. Voiture Bleu: 2 tours complétés
2. Voiture Jaune: 2 tours complétés
3. Voiture Rouge: 1 tours complétés
4. Voiture Vert: 1 tours complétés
La voiture Vert a franchi la ligne d'arrivée en 6.05 secondes!

Classement actuel:
1. Voiture Bleu: 2 tours complétés
2. Voiture Vert: 2 tours complétés
3. Voiture Jaune: 2 tours complétés
4. Voiture Rouge: 1 tours complétés
La voiture Vert a franchi la ligne d'arrivée en 7.06 secondes!

Classement actuel:
1. Voiture Vert: 3 tours complétés
2. Voiture Bleu: 2 tours complétés
3. Voiture Jaune: 2 tours complétés
4. Voiture Rouge: 1 tours complétés
La voiture Vert a franchi la ligne d'arrivée en 8.06 secondes!

Classement actuel:
1. Voiture Vert: 4 tours complétés
2. Voiture Bleu: 2 tours complétés
3. Voiture Jaune: 2 tours complétés
4. Voiture Rouge: 1 tours complétés
La voiture Jaune a franchi la ligne d'arrivée en 9.07 secondes!

Classement actuel:
1. Voiture Vert: 4 tours complétés
2. Voiture Jaune: 3 tours complétés
3. Voiture Bleu: 2 tours complétés
4. Voiture Rouge: 1 tours complétés
La voiture Bleu a franchi la ligne d'arrivée en 10.08 secondes!

Classement actuel:
1. Voiture Vert: 4 tours complétés
2. Voiture Bleu: 3 tours complétés
3. Voiture Jaune: 3 tours complétés
4. Voiture Rouge: 1 tours complétés
La voiture Jaune a franchi la ligne d'arrivée en 11.09 secondes!

Classement actuel:
1. Voiture Vert: 4 tours complétés
2. Voiture Jaune: 4 tours complétés
3. Voiture Bleu: 3 tours complétés
4. Voiture Rouge: 1 tours complétés
La voiture Bleu a franchi la ligne d'arrivée en 12.1 secondes!

Classement actuel:
1. Voiture Bleu: 4 tours complétés
```



```
Classement actuel:
1. Voiture Bleu: 8 tours complétés
2. Voiture Jaune: 8 tours complétés
3. Voiture Vert: 7 tours complétés
4. Voiture Rouge: 2 tours complétés
La voiture Bleu a franchi la ligne d'arrivée en 25.21 secondes!

Classement actuel:
1. Voiture Bleu: 9 tours complétés
2. Voiture Jaune: 8 tours complétés
3. Voiture Vert: 7 tours complétés
4. Voiture Rouge: 2 tours complétés
La voiture Bleu a franchi la ligne d'arrivée en 26.22 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Jaune: 8 tours complétés
3. Voiture Vert: 7 tours complétés
4. Voiture Rouge: 2 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 27.22 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Jaune: 8 tours complétés
3. Voiture Vert: 7 tours complétés
4. Voiture Rouge: 3 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 28.23 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Jaune: 8 tours complétés
3. Voiture Vert: 7 tours complétés
4. Voiture Rouge: 4 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 29.24 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Jaune: 8 tours complétés
3. Voiture Vert: 7 tours complétés
4. Voiture Rouge: 5 tours complétés
La voiture Jaune a franchi la ligne d'arrivée en 30.25 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Vert: 10 tours complétés
3. Voiture Jaune: 10 tours complétés
4. Voiture Rouge: 6 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 36.3 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Vert: 10 tours complétés
3. Voiture Jaune: 10 tours complétés
4. Voiture Rouge: 7 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 37.3 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Vert: 10 tours complétés
3. Voiture Jaune: 10 tours complétés
4. Voiture Rouge: 8 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 38.31 secondes!

Classement actuel:
1. Voiture Bleu: 10 tours complétés
2. Voiture Vert: 10 tours complétés
3. Voiture Jaune: 10 tours complétés
4. Voiture Rouge: 9 tours complétés
La voiture Rouge a franchi la ligne d'arrivée en 39.32 secondes!

Classement actuel:
1. Voiture Rouge: 10 tours complétés
2. Voiture Bleu: 10 tours complétés
3. Voiture Vert: 10 tours complétés
4. Voiture Jaune: 10 tours complétés

Course terminée!

Classement final:

1. Voiture Bleu - 10 tours complétés - Temps total: 26.22 secondes
2. Voiture Vert - 10 tours complétés - Temps total: 34.28 secondes
3. Voiture Jaune - 10 tours complétés - Temps total: 31.26 secondes
4. Voiture Rouge - 10 tours complétés - Temps total: 39.32 secondes

Le gagnant est la voiture Bleu!
```

3.3.2 Analyse des résultat

L'objectif de cette simulation est d'offrir une première validation fonctionnelle avant l'intégration du capteur réel. Une fois le matériel disponible, il faudra comparer les résultats obtenus ici avec ceux du système final pour ajuster les éventuels écarts de détection et améliorer la précision globale.

Voici une analyse des principaux éléments utilisés et de leur impact sur la validation du système :

1. Variables et Structures de Données
    a. Liste/dictionnaire pour stocker les voitures et leurs informations

        i. Permet de suivre individuellement chaque voiture en stockant son nombre de tours complétés, son temps total et son état dans la course.
        ii. Essentiel pour l'affichage des résultats et le classement final.
    b. Compteur de tours pour chaque voiture
        i. Simule la détection des voitures franchissant la ligne d'arrivée.
        ii. Utile pour vérifier que le système arrête bien de suivre une voiture lorsqu'elle termine la course.
    c. Temps simulé pour les tours
        i. Remplace temporairement les données réelles des capteurs en générant des valeurs aléatoires ou en utilisant une fonction de temps.
        ii. Permet de tester le classement sans dépendre du capteur physique.
2. Système de Détection (Simulation de la Captation du Signal)
    a. Utilisation d'un événement déclencheur pour simuler un capteur
        i. Simule le passage d'une voiture en envoyant un signal fictif à chaque tour.
        ii. Permet de tester si l'application enregistre correctement les tours et le temps.
    b. Conditions pour déterminer quand une voiture a fini la course
        i. Vérifie que la simulation fonctionne avec des scénarios variés (exemple : voitures finissant à des temps différents).
3. Affichage et Interface
    a. Affichage en temps réel de la position des voitures
        i. Permet d'évaluer si le système met à jour correctement le classement au fur et à mesure que les tours sont complétés.
        ii. Facilite la visualisation du déroulement de la course.
    b. Affichage du classement final
        i. Vérifie que le programme attribue correctement la première place à la voiture ayant terminé en premier.

4 Rétroaction utilisateur

- **Rétro utilisateur 1 :** "Votre projet est bien dans l'ensemble, mais il serait intéressant d'améliorer la précision des capteurs pour éviter les erreurs de détection. Peut-être en testant différents types de capteurs ou en optimisant leur position ?"

- **Rétro utilisateur 2 :** "L'idée d'utiliser des coques interchangeables est très pertinente, surtout pour différencier les voitures. Cependant, il faudrait s'assurer que le système d'attache soit assez robuste pour éviter que les coques ne se détachent pendant la course."

- **Rétro utilisateur 3 :** "Le concept basé sur des capteurs de couleur est prometteur, mais avez-vous envisagé les variations d'éclairage qui pourraient affecter la détection ?"

5 mise à jour des spécifications cible

- Le code doit être capable d'analyser les images qui sont prises par le capteur .
- La ligne d'arrivée doit tenir en équilibre.
- Les coques sont capables de rester en place sur chaque voiture.
- La ligne d'arrivée doit être capable de supporter la caméra et avoir un emplacement raisonnable.

6- Plan de prototypage du prototype 2

Concept de Conception :	Capteur de couleur d'une course de voiture		
Numéro de test	**Problème critique probable** Quelles assomptions testez-vous ?	**Objectif du test (Pourquoi)** Communications, mesure de la performance, gestion des risques, apprentissage/ compréhension	**Description du test (quoi)** Qu'allez-vous tester précisément ? Quelle est votre hypothèse ?
1	Le code est capable d'analyser les images prises par le capteur	Vérifie la communication entre le capteur et l'interface utilisateur, et assurer la fiabilité de détection	Test la reconnaissance des couleurs et la fiabilité du code à reconnaître les différentes couleurs
2	La ligne d'arrivée est solide et stable	Assurer la fiabilité de la détection des voitures lorsqu'elles franchissent la ligne	Tester la stabilité de la ligne en passant des voitures à différentes vitesses. Hypothèse : La ligne reste en place et ne cause pas d'erreurs dans la détection.

Numéro de test	**Méthode d'analyse (comment et quand)** Plus précisément, comment allez-vous tester, en incluant des éléments tels que la durée, la séquence de test, l'équipement, les critères de réussite/échec, etc. Comment les résultats seront-ils collectés ?	**Déterminer les éléments mesurables** Que testez-vous avec votre concept (attributs mesurables cibles) ?
1	Méthode de test: - Faire passer un objet d'une certaine couleur à différente vitesse, puis itérer l'action pour chaque différent couleur Criteres de reussite: - On veut un minimum de 80% de précision dans l'identification. Collecte des résultats: - On va enregistrer chaque détection correcte/incorrecte et comparer avec l'attendu Durée: 2-4 heures	Plus précisément, on veut s'assurer que le logiciel détecte chaque voiture lorsqu'elle passe sous le capteur. Et en soit, la fiabilité et la précision de la détection des couleurs en condition de course.
2	Il sera difficile de tester des cas où des voitures frappent la ligne d'arrivée, mais on va s'assurer que les dimensions permettent de tenir la piste en place, et à vue d'œil elle est stable et solide.	On veut s'assurer que la ligne d'arrivée reste en place. En soi, on veut tester la stabilité du système.

7- Conclusion

En conclusion, ce premier prototype constitue une étape essentielle dans le développement de notre projet, nous permettant de concrétiser nos idées tout en intégrant les retours des clients. L'analyse approfondie des sous-systèmes, incluant la ligne d'arrivée, la coque, l'interface utilisateur et le capteur de couleur a guidé nos choix techniques et optimisé notre approche. La planification rigoureuse des ressources et l'évaluation des risques nous ont offert une vision claire des défis à relever et des solutions envisageables. Enfin, le plan d'essai détaillé garantira la validation et

l’amélioration du système, nous préparant efficacement pour les prochaines phases de développement.