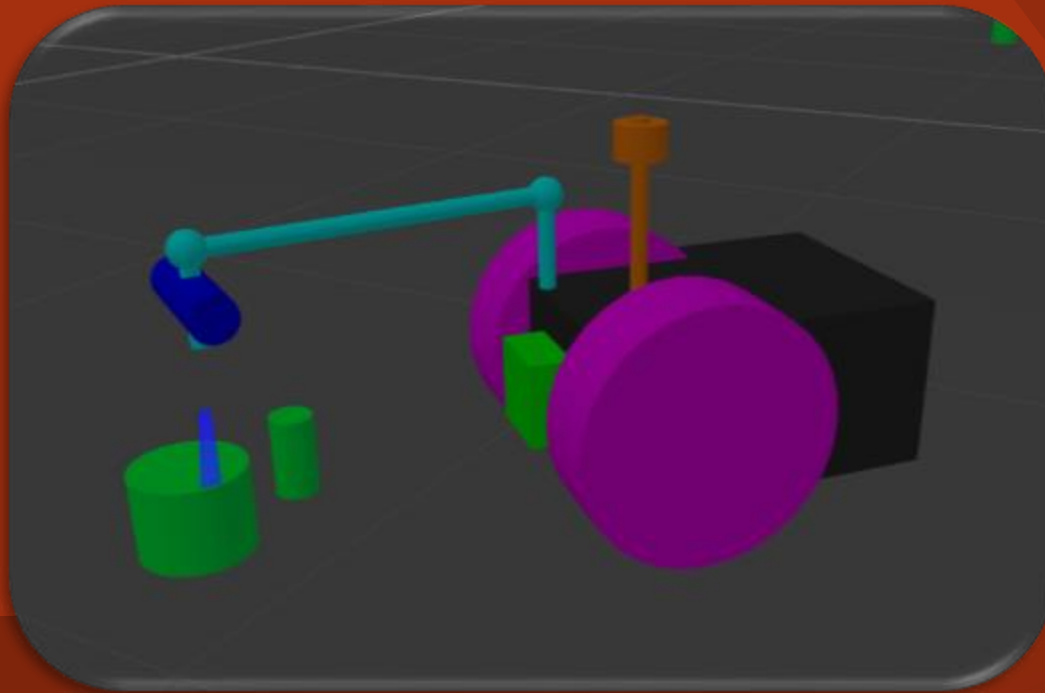


Desherbator

U.V. 5.8 - Robotique Ingénierie Système

Benoît Zerr
Joris Tillet
Rémi Rigal

© 2014 U.V. 5.8 - Robotique Ingénierie Système



Alexandre Houdeville

Élodie Noëlé

Fabrice Poirier

Romain Dussot

Maxime Do Rosario

Évann Clavier

Sommaire

- Présentation du projet
- Exigences initiales
- Solutions techniques
- Taiga
- Github
- Modifications du cahier des charges
- RETEX
- Défauts du produit final
- Conclusion

Présentation du projet

➤ Objectifs:

- Robot désherbeur capable d'évoluer dans un milieu **terrestre plan délimité par des murs**
- **Système de vision** pour percevoir l'environnement
- Destruction des herbes à l'aide d'un **LASER**
- Déplacement à l'aide d'une **plateforme roulante**

Présentation du projet

➤ Spécifications:

➤ Robot:

- Taille et vitesse du robot réalistes
- Libre choix des capteurs de navigation (GPS, LIDAR, encodeurs, sonars, etc)
- Prise en compte de l'évitement d'obstacles non requise
- Aspect énergétique non pris en compte

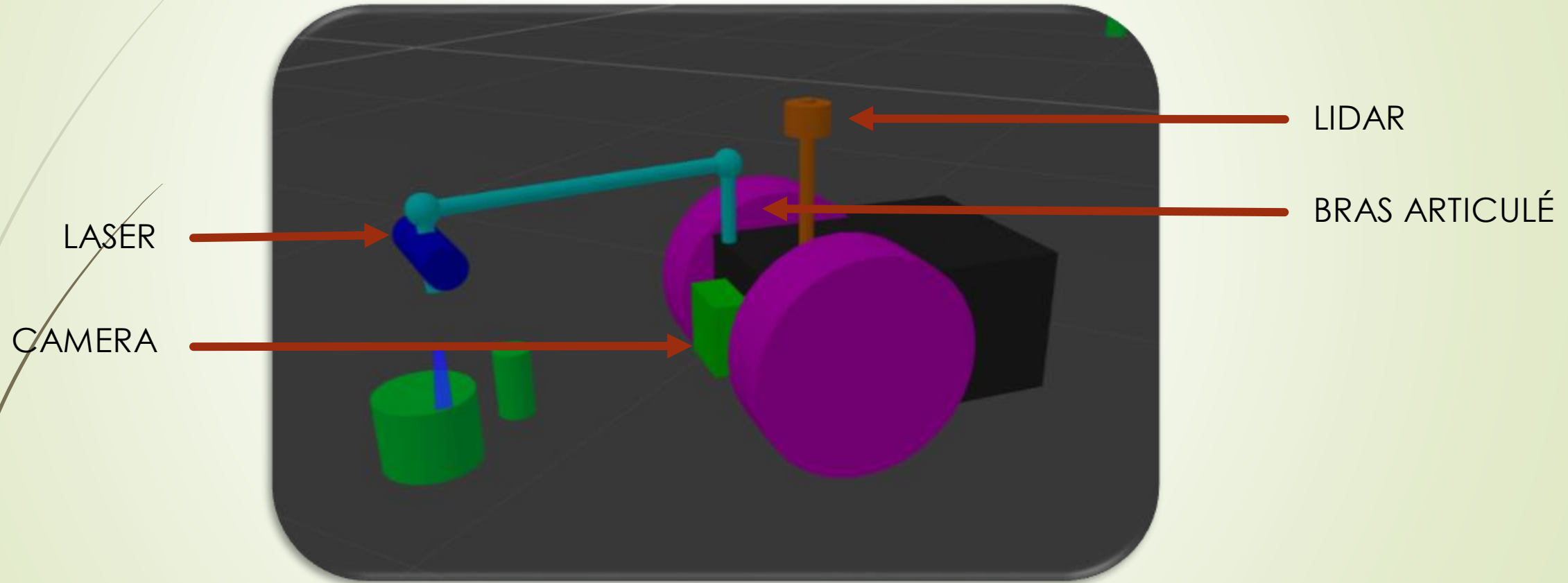
➤ Terrain:

- Représentation simplifiée du terrain (fluidité et implémentation rapide du simulateur)
- Temps d'efficacité du laser accéléré pour la simulation

Exigences initiales

Exigences	Sous-exigences	Critères	Tags
Modélisation physique d'un robot désherbeur ("Desherborator")	Plate-forme roulante réaliste	Vitesse linéaire: [0-10] km/h	A1
	Détection de mauvaises herbes	Système de vision	A2
	Destruction de mauvaises herbes	LASER désherbeur agit en 3 minutes* (* temps réel et non simulé)	A3
	Taille restreinte	Cube de 0.5 m de côté	A4
	Localisation et Mapping	Capteurs intéroceptifs et extéroceptifs (bruités)	A5
Modélisation d'un environnement	Cour plane horizontale	Sol de (10m x 10m) gris	B1
	Murets	Mur de 0.5m de haut	B2
	Mauvaises herbes	Positions aléatoires de cylindres verts (h*d : 10x[2-15] cm)	B3

Solutions Techniques - Hardware

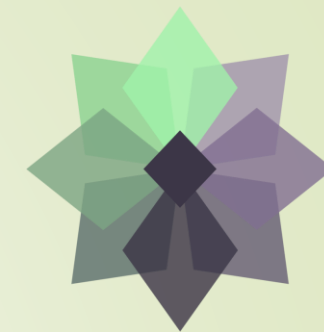


Solutions Techniques - Software

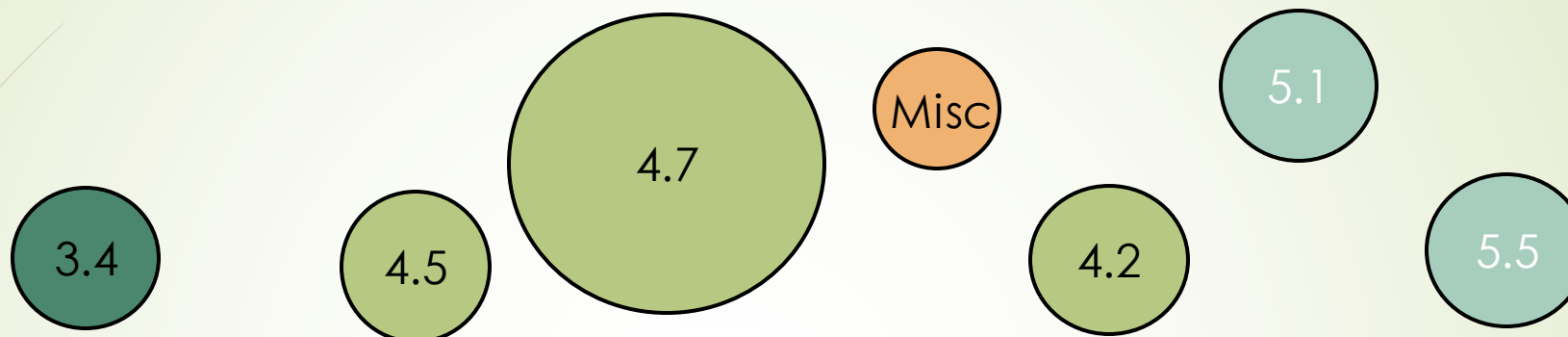
Tags	Exigences Techniques	Solutions apportées	Mots-clefs
A1	Vitesse linéaire: [0-10] km/h	Régulateur et Seuil	<i>PID</i>
A2	Système de vision	Caméra et traitement d'image	<i>HSV, Isolation, Barycentre</i>
A3	LASER désherbeur agit en 3 minutes* (* temps réel et non simulé)	Bras articulé (2 ddl)	<i>GAZEBO Architecture Xacro / URDF</i>
A4	Cube de 0.5 m de côté	Architecture globale	<i>GAZEBO Architecture Xacro / URDF</i>
A5	Capteurs intéroceptifs et extéroceptifs (bruités)	GPS, LIDAR & Caméra	<i>SLAM, Régulation</i>
B1	Sol de (10m x 10m) gris	Architecture sur Gazebo	<i>Xacro /URDF, Distribution Aléatoire, Xacro Création Dynamique</i>
B2	Mur de 0.5m de haut		
B3	Positions aléatoires de cylindres verts (h*d : 10x[2-15] cm)		

7

Taiga : Infographie

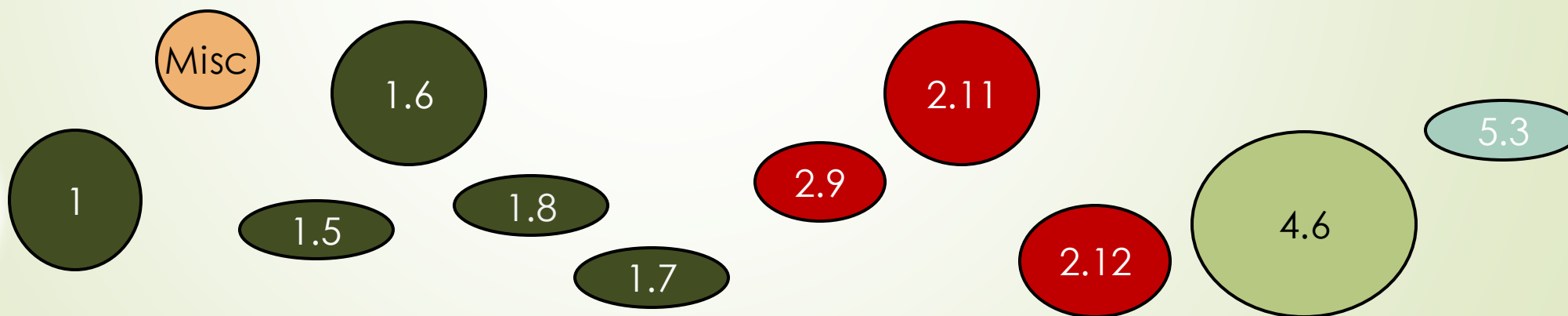


Maxime



Timeline

Fabrice

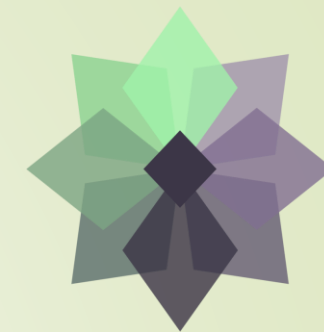


Timeline

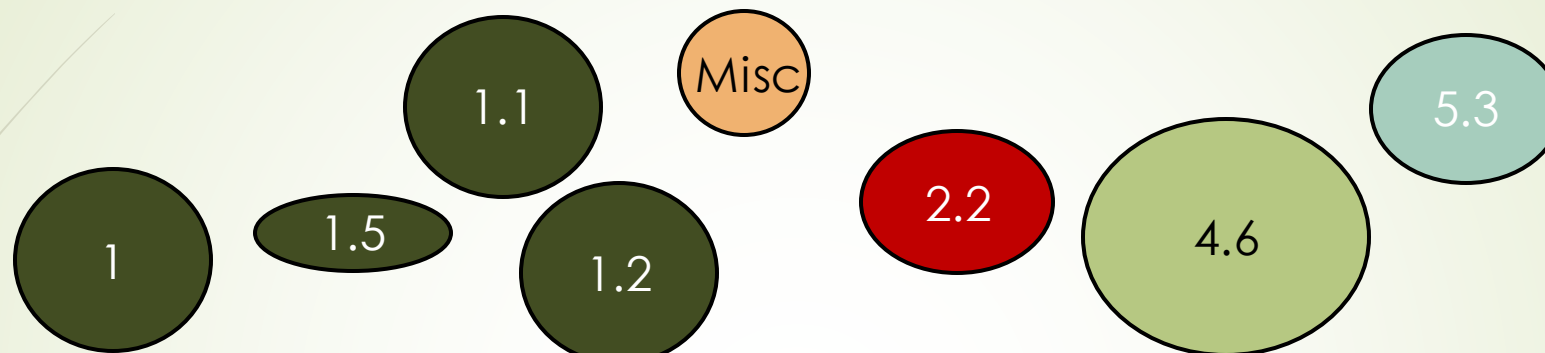
■ Miscellaneous
 ■ Task 1.X
 ■ Task 2.X
 ■ Task 3.X
 ■ Task 4.X
 ■ Task 5.X

8

Taiga : Infographie

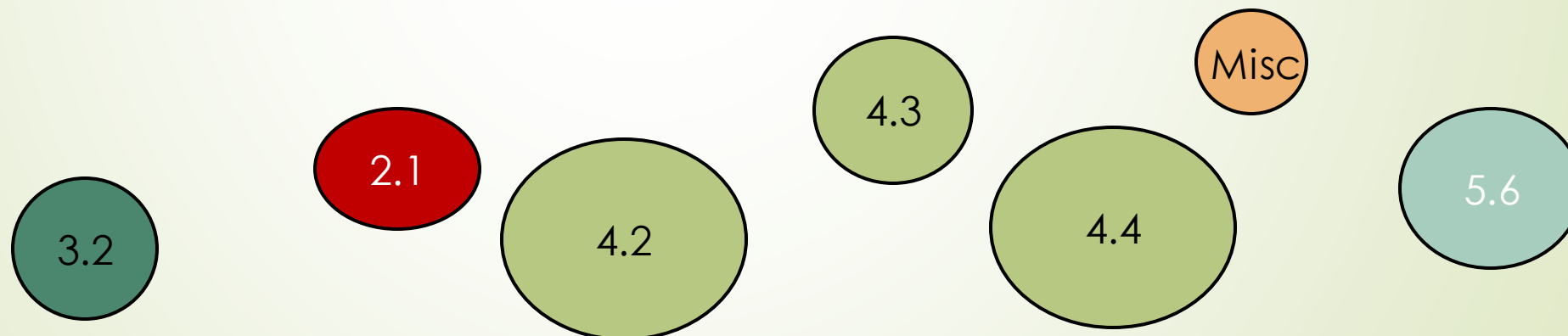


Romain



Timeline

Evann



Timeline

Miscellaneous Task 1.X Task 2.X Task 3.X Task 4.X Task 5.X

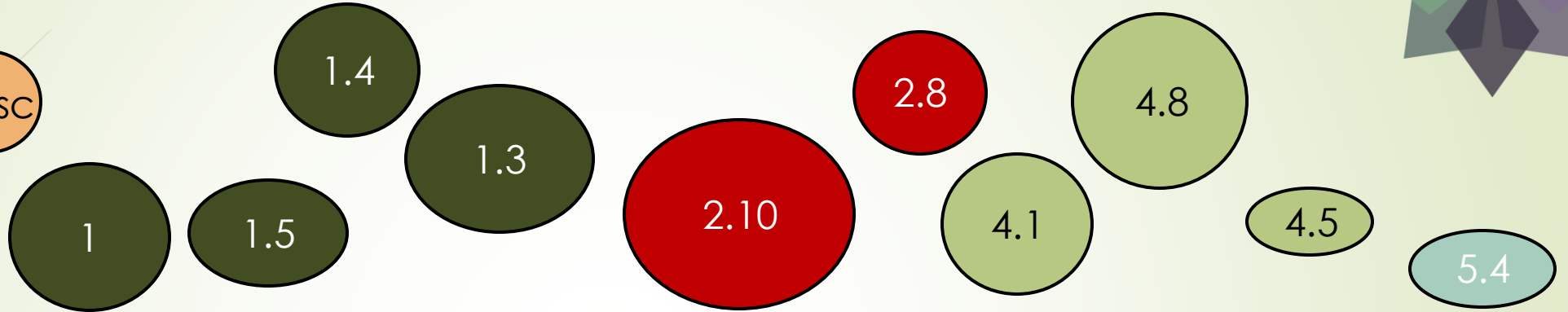
Taiga : Infographie



Elodie

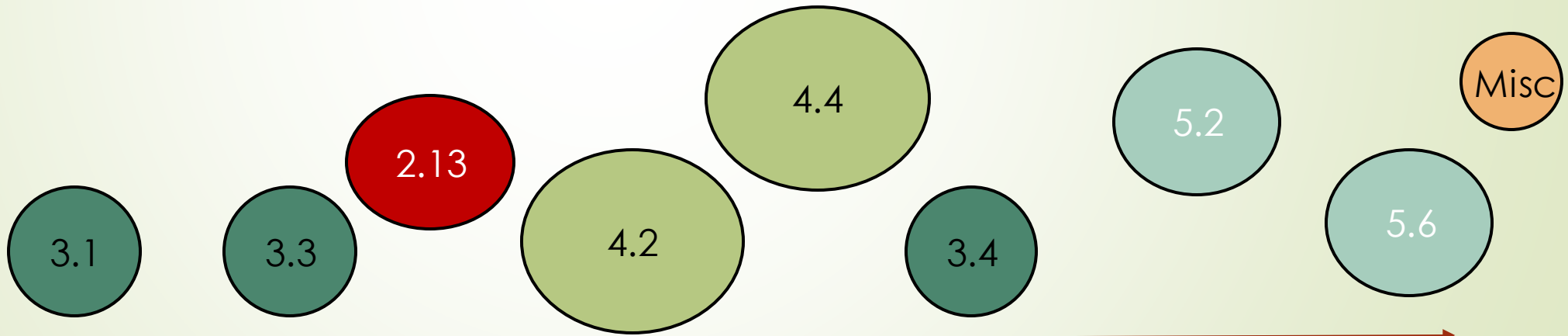


Misc



Timeline

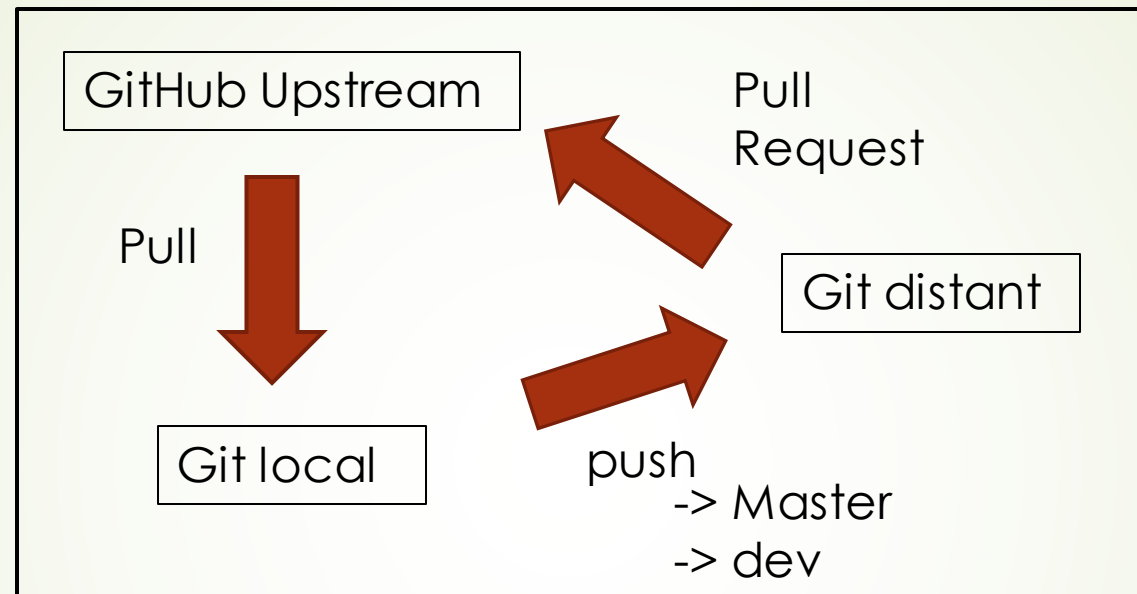
Alexandre



Misc

Timeline

GitHub



GitHub Repository:

desherborator_control

Launch pour lancer
l'ensemble

desherborator_description

Modélisation du robot
(mécanique & algorithmes)

desherborator_gazebo

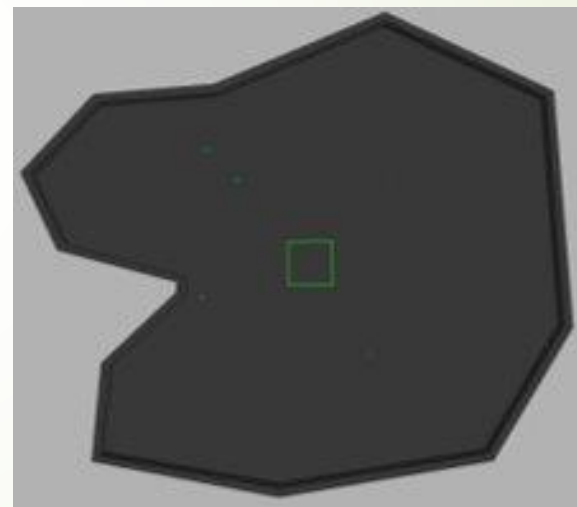
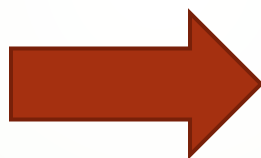
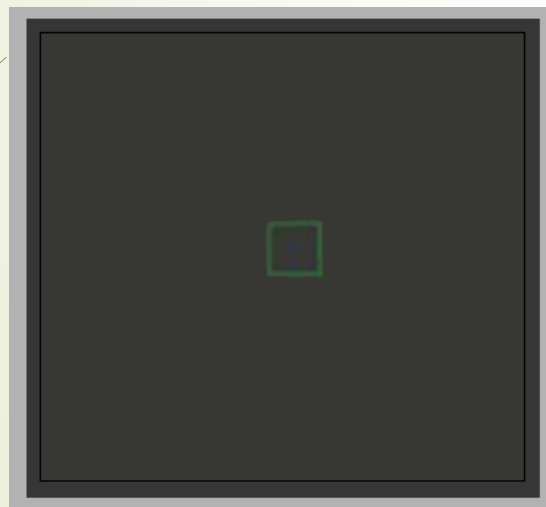
Modélisation du terrain
précédent

gazebo_client_garden

Modélisation du terrain
suivant

Modifications du cahier des charges

Changement de map:



Défauts du produit final

- Simulateur dépendant de la version de ROS
- Action du LASER conditionnée par un temps constant et non un résultat sur l'aspect de l'herbe
- Action du LASER parfois déclenchée dans le vide



Retour d'expérience



➤ Gazebo:

- Beaucoup de problèmes rencontrés liés à l'origine de la version de ROS ou linux et conflits entre ROS et python
- Problème de gestion des collisions entre les éléments (départ du robot dans un mur)

Retour d'expérience



➡ Git:

- ➡ Problèmes au niveau du git workflow: nombreux conflits gérés par le product owner
- ➡ Mauvaise séparation entre la branche de développement et la branche master (éléments non fonctionnels présents sur la branche master)
- ➡ Gestion git des fichiers dépendant des machines (fichiers de compilation: CMakeList)
- ➡ Mise à jour sous forme de packets et non le workspace complet

Retour d'expérience



➤ Outil de gestion de projet Taiga:

- Répartition claire et détaillée des tâches
- Suivi en temps réel de l'état du projet
- Recommandations établies initialement non respectées en totalité au fil du projet

Conclusion

➤ Bilan:

- Robot désherbeur conçu avec succès et produit final correspondant aux exigences initiales
- Adaptation aux changements d'exigences du client grâce à l'architecture organisée choisie

➤ Recommandations/suggestions pour la prochaine promo:

- Proposer un Workflow pour Github et gestion AGILE si demandé par les étudiants
- Format de projet très intéressant et positif

■ Apportez des cookies pour les gagnants!

