|  |
| --- |
| Dolisquad |
| Party drone |
| Le drone qui va révolutionner vos soirées |

|  |
| --- |
| Louis-auguste Dumas, Gustave julien, Leo Levacher, Simon Casteran, Thibault Delerue, Thibault Jacques  07/06/2021 |

# Présentation du projet

L’objectif de ce projet est de fournir un drone riche en fonctionnalité. Ce drone offrira de nombreux moyen de control et de nombreuse fonctionnalité.

En moyen de contrôle nous trouverons :

* Reconnaissance vocale
* Client Qt et client electron
* Un client web

Les principales fonctionnalités seront les suivantes :

* Mode « toutou », le drone va suivre l’utilisateur
* Mode « patrouille », le drone va patrouiller et à la rencontre de certains visages émettre un son.
* Monde « control », ce mode laisse l’utilisateur contrôler le drone à partir de l’un des clients.

Il y aura également de présent une gestion de l’utilisateur.

# Description technique

(Voir le schéma en annexe)

L’architecture du projet est la suivante : des clients vont communiquer avec la « master API ». Cette dernière va effectuer un traitement pour que l’action ait lieu. Si par exemple l’utilisateur veut contrôler le drone, l’api master va communiquer avec l’interface « control ». Cette dernière va contenir la partie logique de ces actions et ensuite appeler la librairie qui permet de contrôler le drone.

# Les clients :

Ici nous allons rentrer dans le détail des différents clients.

## Contrôle à la reconnaissance vocal

Ici il s’agira d’un programme qui tournera lorsque l’utilisateur veut utiliser son micro pour contrôler le drone. Ce dernier lorsqu’il reconnaitra une expression enverra une requête a la master API pour faire en sorte que le drone effectue les actions.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Gustave | Implémentation de l’IA de STT (converti une commande vocale en une string) | Ko |
| Mineur | Gustave | Récupération des commandes vocales et formatage des fichiers audio | Ko |
| Mineur | Gustave | Liaison entre commande vocal et commande du drone | ko |

## Interface web pour contrôler le drone

Ici il s’agira d’une interface Web qui va permettre à l’user de contrôler le drone. Cette dernière sera faite en react js. Elle va contenir le flux vidéo du drone, les différentes commandes permettant à l’utilisateur de choisir le mode du drone et le contrôler de manière manuelle.

La prise en charge de la manette est également prévue.

La création de compter et connexion sera également disponible.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Leo | Création de la page avec un flux vidéo | ko |
| Majeur | Leo | Avoir les boutons de contrôle | ko |
| Majeur | Leo | Implémenter les contrôles clavier | ko |
| Mineur | Leo | Faire un CSS propre | ko |
| Mineur | Leo | Faire la documentation de la partie visuelle | ko |
| Mineur | Leo | Faire la librairie Selenium\* pour tester | ko |
| Mineur | Leo | Faire la documentation de la partie code | ko |
| Mineur | Leo | Documentation de la partie Selenium | ko |
| Mineur | Leo | Adapter la charte graphique | ko |

\*Selenium: Framework python de test fonctionnel

## Interface client pour contrôler le drone

Ici il s’agira de deux interfaces logiciel qui vont permettre à l’user de contrôler le drone. Cette dernière sera faite en Qt et electron. Elle va contenir le flux vidéo du drone, les différentes commandes permettant à l’utilisateur de choisir le mode du drone et le contrôler de manière manuelle.

La prise en charge d’une manette et sa configuration sont prévus.

La création de compte et connexion sera également disponible.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Thibault d | Apprentissage du framework Qt | ko |
| Majeur | Thibault d | Réalisation d’un GUI simple pour mettre en œuvre Qt |  |
| Majeur | Thibault d | Mise en place d’Electron permettant de transformer le front web en application de bureau | ko |
| Majeur | Thibault d | Affichage du flux vidéo du drone chez Qt | ko |
| Majeur | Thibault d | Gestion de la manette pour contrôler le drone | ko |
| Majeur | Thibault d | Permettre à l’utilisateur de remapper les contrôles de la manette | ko |
| Mineur | Thibault d | Gestion des routes de l’API avec des boutons | ko |
| Mineur | Thibault d | Unification de la charte graphique sur Qt | ko |
| Mineur | Thibault d | Intégration des fonctionnalités mineures de l’API | ko |
| Mineur | Thibault d | Optimisation des performances Qt | ko |
| Mineur | Thibault d | Assurance du fonctionnement cross-platform | ko |
| Mineur | Thibault d | Cleanup de l’interface Qt | ko |
| Mineur | Thibault d | Assurance du build Qt avec cmake |  |

# Le backend :

Ici nous discuterons des différentes apis utilisables

## API master

Cette Api va permettre aux différents clients d’effectuer les actions en utilisant les interfaces qui correspondent au type de fonctionnalité.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Louis | Conception de l’architecture | ko |
| Majeur | Louis | Création du socle l’api (robustesse …) | ko |
| Majeur | Louis | Ajout des urls pour les fonctionnalités (“toutou” et control) | ko |
| Mineur | Louis | Ajout des urls pour les fonctionnalités “patrouille” | ko |
| Mineur | Louis | Gestion des threads en fonction des cas et fonctionnements des fonctions plus bas niveau | ko |
| Mineur | Louis | Gestion des threads en fonction des cas et fonctionnements des fonctions plus bas niveau | ko |
| Mineur | Louis | Mise en relation avec l’api utilisateur | ko |
| Mineur | Louis | Logique pour les fonctionnalités | ko |
| Mineur | Louis | Documentations | ko |

## Api gestion user

La gestion des users est également prévu. Ici il s’agira d’une API permettant de gérer la base de données, créer des utilisateurs, leur laisser la possibilité de se connecter et vérifier que l’user est logué lorsqu’il veut intervenir sur l’api.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Simon | Création d’un template en typescript | ko |
| Mineur | Simon | Set up de la db sur Atlas et des controllers | ko |
| Mineur | Simon | Set up des models | ko |

## Interface « toutou »

Cette interface contient toute la logique de la fonctionnalité toutou. En effet elle utilise reconnaissance facile afin de suivre l’utilisateur prédéfinis.

Ici le drone ne fera que suivre l’utilisateur.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Leo | Analyser les images du drone avec de l’IA | ko |
| Majeur | Leo | Effectuer un déplacement après analyse de l’image | ko |
| Mineur | Leo | Récupérer les images | ko |
| Mineur | Leo | Effectuer un choix logique en fonction de l’analyse | ko |
| Mineur | Leo | Documentation du code | ko |
| Mineur | Leo | Faire en sorte d’analyser le minimum d’images nécessaires afin d’optimiser les performances | ko |
| Mineur | Leo | Debug (car je ne serais pas en possession du drone tout le temps) | ko |
| Mineur | Leo | Optimisation du code (car latence si le code n’est pas optimisé et potentiel crash du drone dans un mur ou sur la personne) | ko |

## L’interface « patrouille »

Ici le drone va se balader de manière autonome et émettre un son en fonction de l’utilisateur qu’il rencontre.

Le drone doit avoir connaissance de son espace.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Thibault J | Détecter une ou plusieurs présences | ko |
| Majeur | Thibault J | Analyser l’environnement du drone pour permettre d’emettre un son au bon moment | ko |
| Majeur | Thibault J | Diriger le drone de manière autonome en fonction de l’endroit voulu | ko |
| Majeur | Thibault J | Liaison entre les logiques de déplacement du drone et détection de l’environnement | ko |
| Mineur | Thibault J | Mesure de la pièce dans laquelle est diposé le drone grâce à de l’ultrason ou autres | ko |
| Mineur | Thibault J | Optimiser les mouvements directionnels du drone pour aller d’un point A à un point B | ko |
| Mineur | Thibault J | Debug environnement et détection | ko |
| Mineur | Thibault J | Debug décompression des images du drone pour les analyser | ko |
| Mineur | Thibault J | Optimisation de l’image en sortie du drone | ko |
| Mineur | Thibault J | Calibrage du son en fonction de la distance de l’utilisateur | ko |
| Mineur | Thibault J |  | ko |
| Mineur | Thibault J |  | ko |

## Drone command interface

Ici il s’agira d’écrire l’interface et la partie logique permettant à l’utilisateur de contrôler le drone, avant, arrière, droite, gauche, bat, haut et mouvements plus complexes. Partie à fortement documenter pour faciliter le travail des devs des clients.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Mineur | Simon | Commandes de mouvement simple | ko |
| Mineur | Simon | Commandes de mouvement avancé, mouvement continu dans un sens, arcs de cercle, etc | ko |
| Mineur | Simon | Manœuvres, looping si le drone le permet, danse aérienne, etc | ko |
| Mineur | Simon | Manœuvres complexes suite | ko |
| Mineur | Simon | Fonctions utilisant la caméra, prendre une photo ou une vidéo. | ko |

# Library features

Ici il sera question des librairies les plus bas niveaux sur ce projet.

## Facial recognition

Il s’agira de faire une librairie qui va permettre de gérer la reconnaissance facile.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Gustave | Implémentation d’une IA de reconnaissance faciale | ko |
| Mineur | Gustave | Entrainement du modèle | ko |
| Mineur | Gustave | Liaison de la reconnaissance des visages avec différentes commandes du drone | ko |

## Sound API

Une api qui va permettre de lancer le son sur un RB fixé sur le drone.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Gustave | Création de l’API | ko |
| Mineur | Gustave | Création d’une banque de son à jouer par le drone et implémentation TTS | ko |
| Mineur | Gustave | Liaison de la banque de son avec les différentes fonctionnalités du drone | ko |
| Majeur | Thibault J | Créer une interface pour communiquer avec l’api | ko |
| Majeur | Thibault J | Créer une interface pour communiquer avec l’api | ko |

## Command drone

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Simon | Public interface | ko |
| Mineur | Simon | Movement private methods | ko |
| Mineur | Simon | Camera private methods | ko |

# Devops

## CI

Push lorsque les tests passent.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Simon | Setup du pipeline Github action implémentant les linters, sonar kube | ko |
| Majeur | Simon | Implémentation des tests fonctionnels dans le pipeline | ko |
| Majeur | Simon | Implémentation des nouveaux tests fonctionnels dans le pipeline | ko |
| Majeur | Simon | Implémentation des nouveaux tests fonctionnels dans le pipeline | ko |
| Mineur | Simon | Linter typescript for User API | ko |
| Mineur | Simon | Linter on API master | ko |
| Mineur | Simon | Linter on front ts + js | ko |
| Mineur | Thibault J |  | ko |
| Mineur | Thibault J |  | ko |
| Mineur | Thibault d | Implémentation d’une norme de commit avec husky | ko |
| Mineur | Thibault J | Déploiement automatisé via Ansible | ko |
| Mineur | Thibault J | Déploiement automatisé via Ansible | ko |

## Docker

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Mineur | Thibault d | Dockerisation de l’api master | ko |
| Mineur | Thibault d | Dockerisation de l’api user | ko |
| Mineur | Thibault d | Dockerisation du front | ko |
| Mineur | Thibault d | Docker-compose | ko |
|  |  |  |  |

# Test fonctionnel

Tests sous robot framework. Sur la partie web, api, librairie et interface.

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Gustave | Conception architecture de test | ko |
| Majeur | Gustave | Librairie principale pour tester les apis user | ko |
| Majeur | Gustave | Librairie principale pour tester les apis master | ko |
| Majeur | Louis | Conception architecture de test | ko |
| Majeur | Louis | Librairie de test librairie bas niveau | ko |
| Majeur | Louis | Librairie de test librairie bas niveau | ko |
| Mineur | Gustave | Sonar kube | ko |
| Mineur | Gustave | Sonar kube | ko |
| Mineur | Gustave | Ecriture des fiches de test qui correspond aux tests des fiches de tests | ko |
| Mineur | Gustave | Ecriture des fiches de test qui correspond aux tests des fiches de tests | ko |
| Mineur | Gustave | Ecriture des fiches de test qui correspond aux tests des fiches de tests | ko |
| Mineur | Gustave | Création des fiches de test excel et des procédures de test | ko |
| Mineur | Louis | Création des fiches de test excel et des procédures de test | ko |
| Mineur | Louis | Création des fiches de test excel et des procédures de test | ko |
| Mineur | Louis | Création des fiches de test excel et des procédures de test | ko |
| Mineur | Louis | Ecriture des fiches de test qui correspond aux tests des fiches de tests | ko |
| Mineur | Louis | Ecriture des fiches de test qui correspond aux tests des fiches de tests | ko |
| Mineur | Louis | Ecriture des fiches de test qui correspond aux tests des fiches de tests | ko |

# Charte graphique :

Notre projet disposant de 4 interfaces client il est vital de garder la même charte graphique sur l’ensemble des clients. Travail sur les maquettes le premier jour pour définir les

|  |  |  |  |
| --- | --- | --- | --- |
| Fonctionnalité | Personne en charge | Description | Statuts |
| Majeur | Leo | Maquette | ko |

# Annexe

