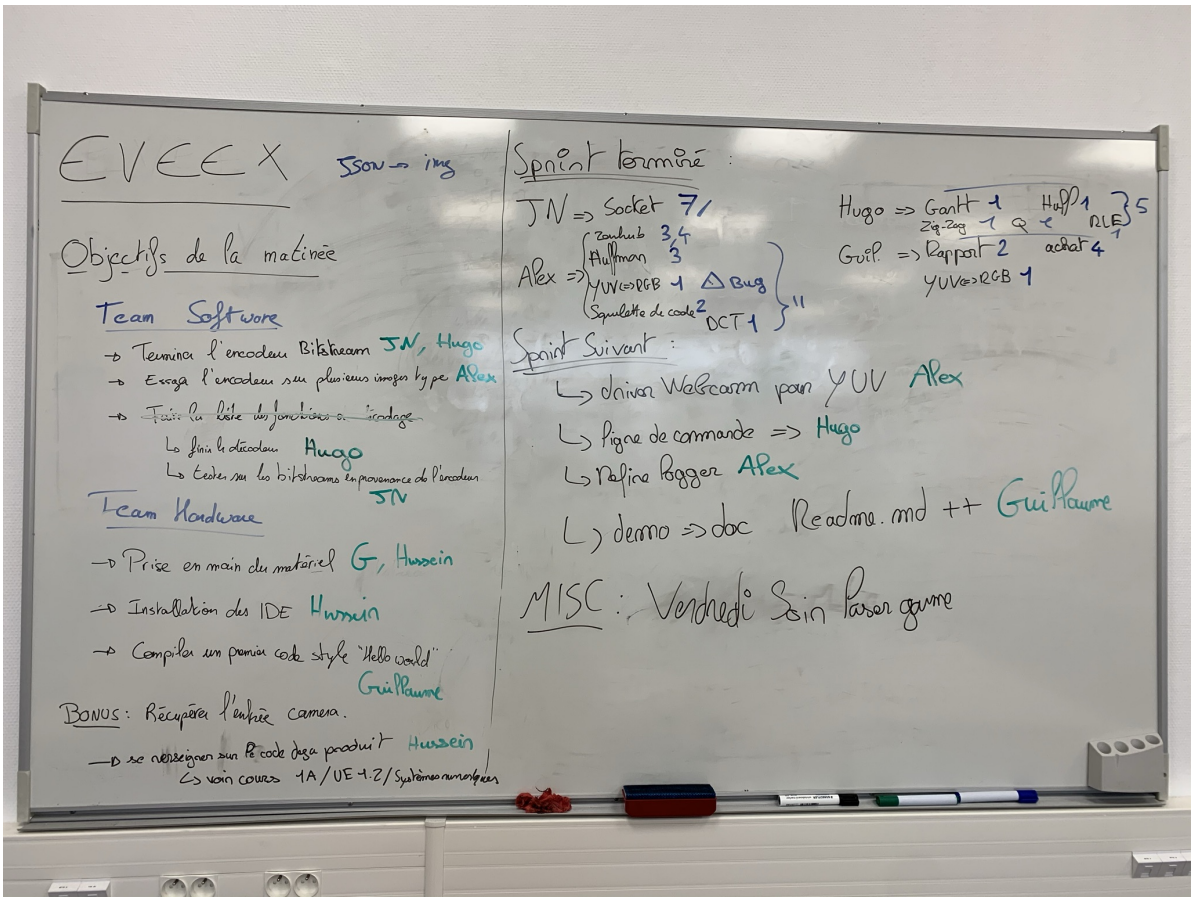
Récap sprint n°2

auteur: Guillaume leinen

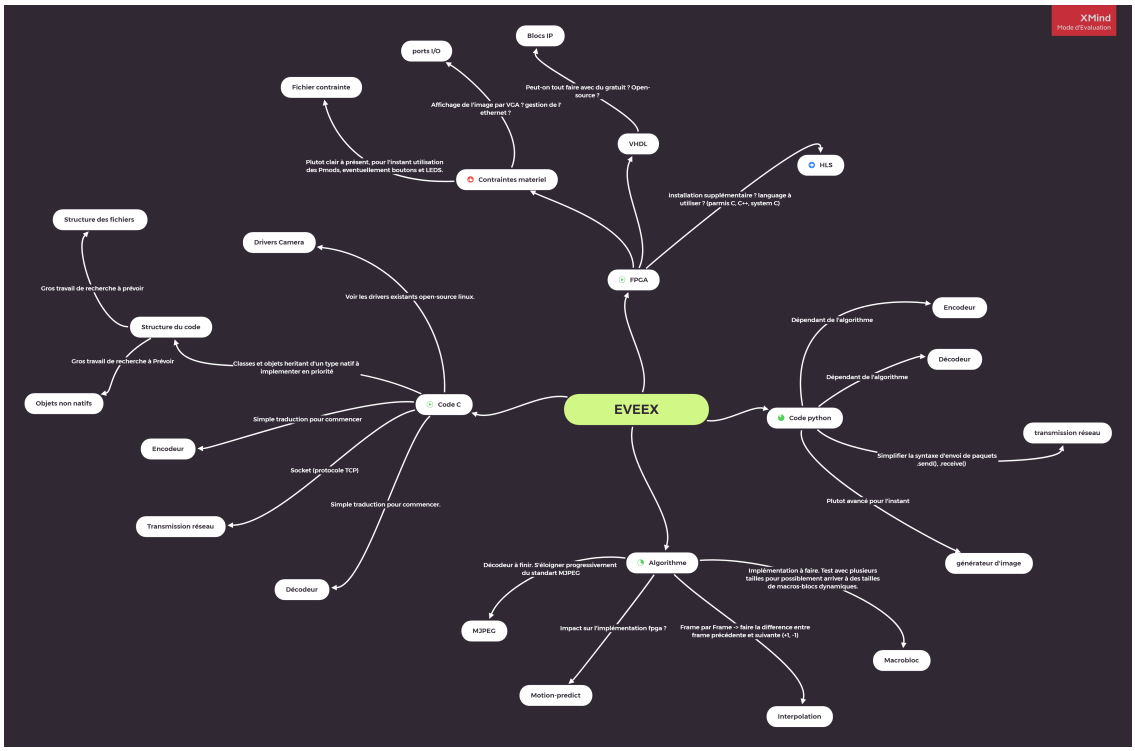
Ce qui avait été décidé au sprint précédent :



Ce qui a été fait :

- L'encodeur est terminé et opérationnel, il génère des bitstream pour des images test, dont un générateur a été créé par Alexandre en python et JSON.
- Concernant la partie FPGA, le logiciel commence à rentrer pour moi et Hussein. On a pu réaliser relativement facilement un exemple type "Hello world" en faisant clignoter une LED. La prochaine étape consiste à développer sur des exemples simples par nous même, et de commencer les choses "utiles". J'ai notamment aujourd'hui compilé un code avec l'aide de P.Cotret qui récupère le flux vidéo des cameras OV7670 et l'affiche sur un écran VGA. je n'ai pas de fils de prototypage à disposition immédiate comme je suis chez mes parents mais je devrais pouvoir me débrouiller.
- Pour la partie webcam, on peut dans un premier temps extraire directement un flux de type YUV de la webcam par la bibliothèque *open-CV* .
- La partie interface en ligne de commande avance bien, il reste quelques bug à corriger mais dans l'ensemble ça avance.
- La documentation est terminé pour une première release. Le readme est disponible sur notre github : https://github.com/EVEEX-Project/EVEEX-Code

- Concernant un autre aspect de la documentation, nous avons élaboré aujourd'hui comme vous nous l'avez demandé une carte mentale, qui m'a d'ailleurs été plutôt utile pour élaboration des issues des prochains sprints.



Un aperçu de la carte mentale en question, un pdf sera en copie du mail

Ce qui n'a pas été terminé :

- le décodeur est quasiment achevé, il ne reste que la DCT inverse à implémenter dans le code.
- la démo n'a par conséquent pas encore été organisé.

Il reste néanmoins pas mal de choses à faire si on veut rester dans des délais raisonnables.

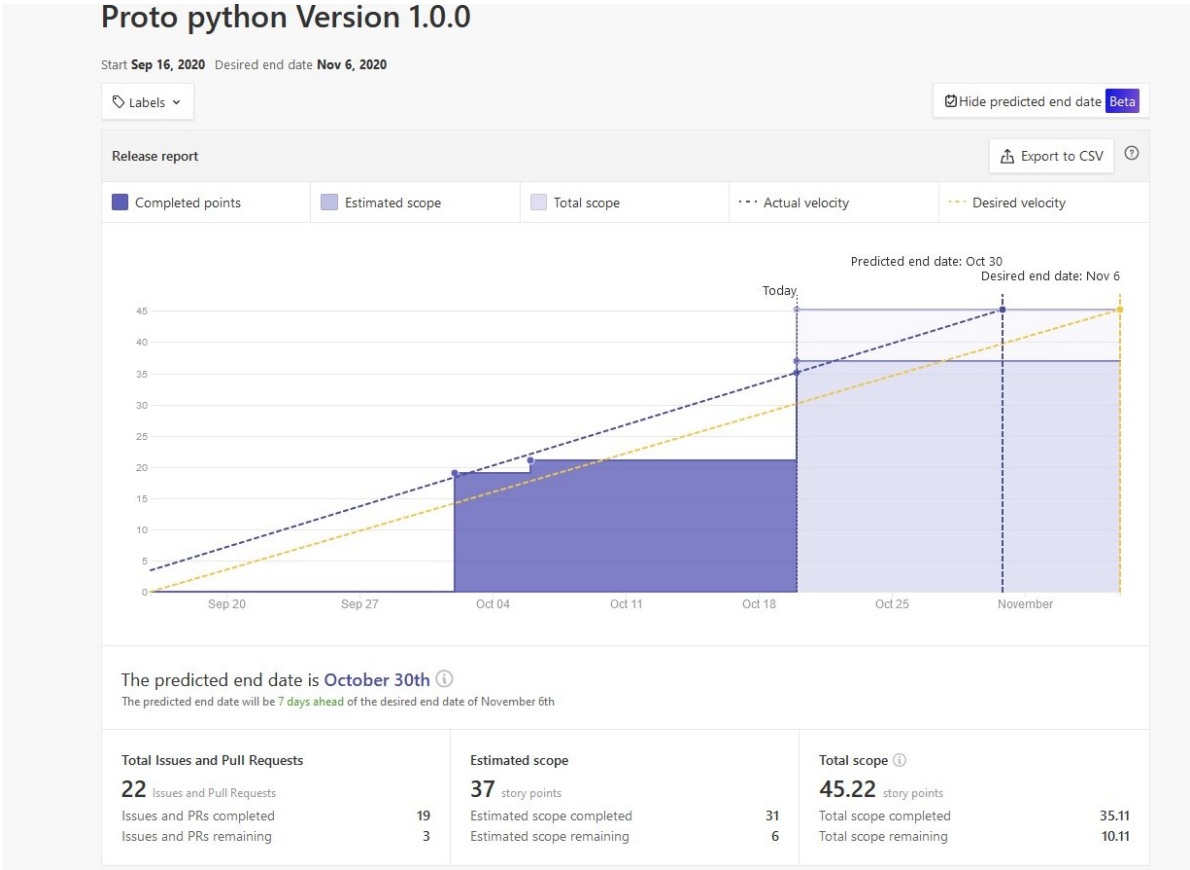
Dans un délai court (le sprint suivant), il faut faire en sorte d'avoir une première release de code fonctionnel sur des images. Cela passe notamment par la finalisation du décodeur et son test. Aussi nous voulons avoir accès à un certain nombre de statistiques de compressions notamment en fonctions des paramètres de l'image.

Concernant la partie FPGA, il va s'agir de fouiller et de tabler sur les blocs IP qui vont nous être nécessaires, afin de déterminer si tout les IP nécessaires sont open-sources ou au moins gratuites. Aussi il va s'agir de chercher des petits exemples de problèmes afin de s'exercer sur la programmation en vhdl.

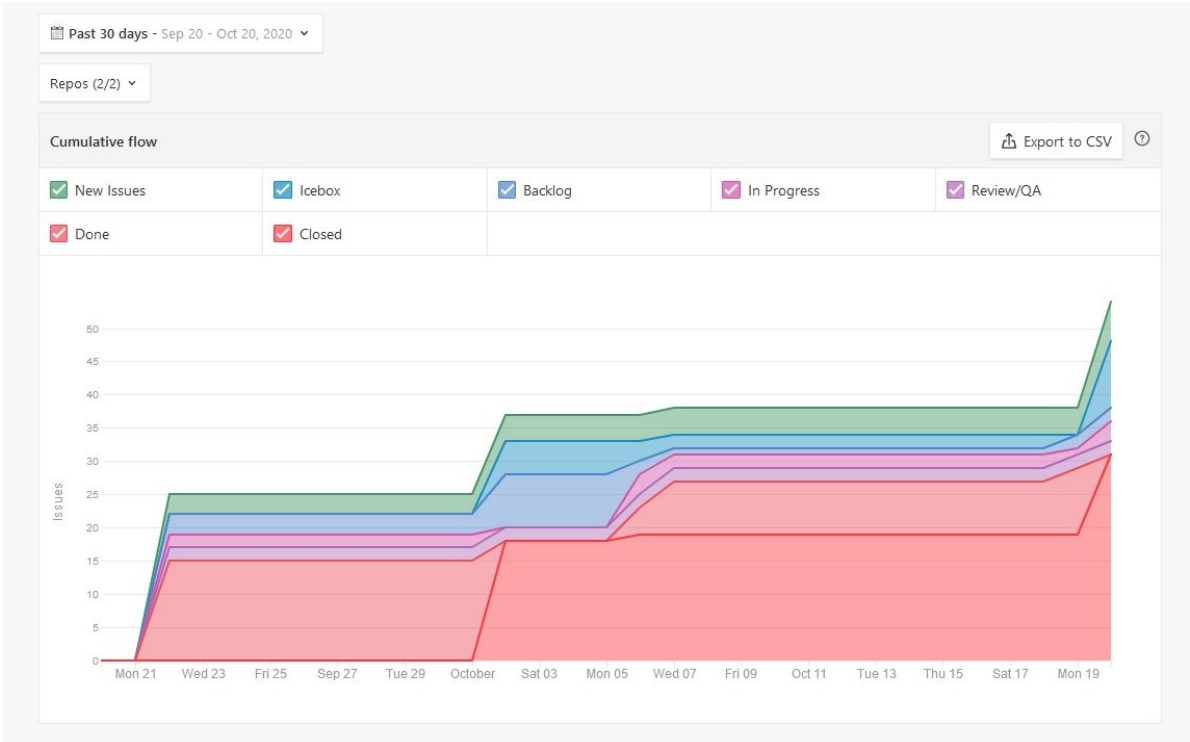
Enfin, nous commençons l'aspect programmation C du projet. Nous allons commencer les recherches sur la structure de fichiers à construire et plus important, sur les types non-natifs à implémenter (par l'intermédiaire de structures).

Le découpage de ces taches à accomplir est consultable sur la page ZenHub du projet : Zenhub

Zenhub nous permet d'avoir un certain nombre de statistiques utiles dont voici un aperçu:



le Release report nous permet d'avoir une estimation sur les délais du projet



le diagramme cumulatif nous donne aussi une évaluation de notre dynamique de groupe