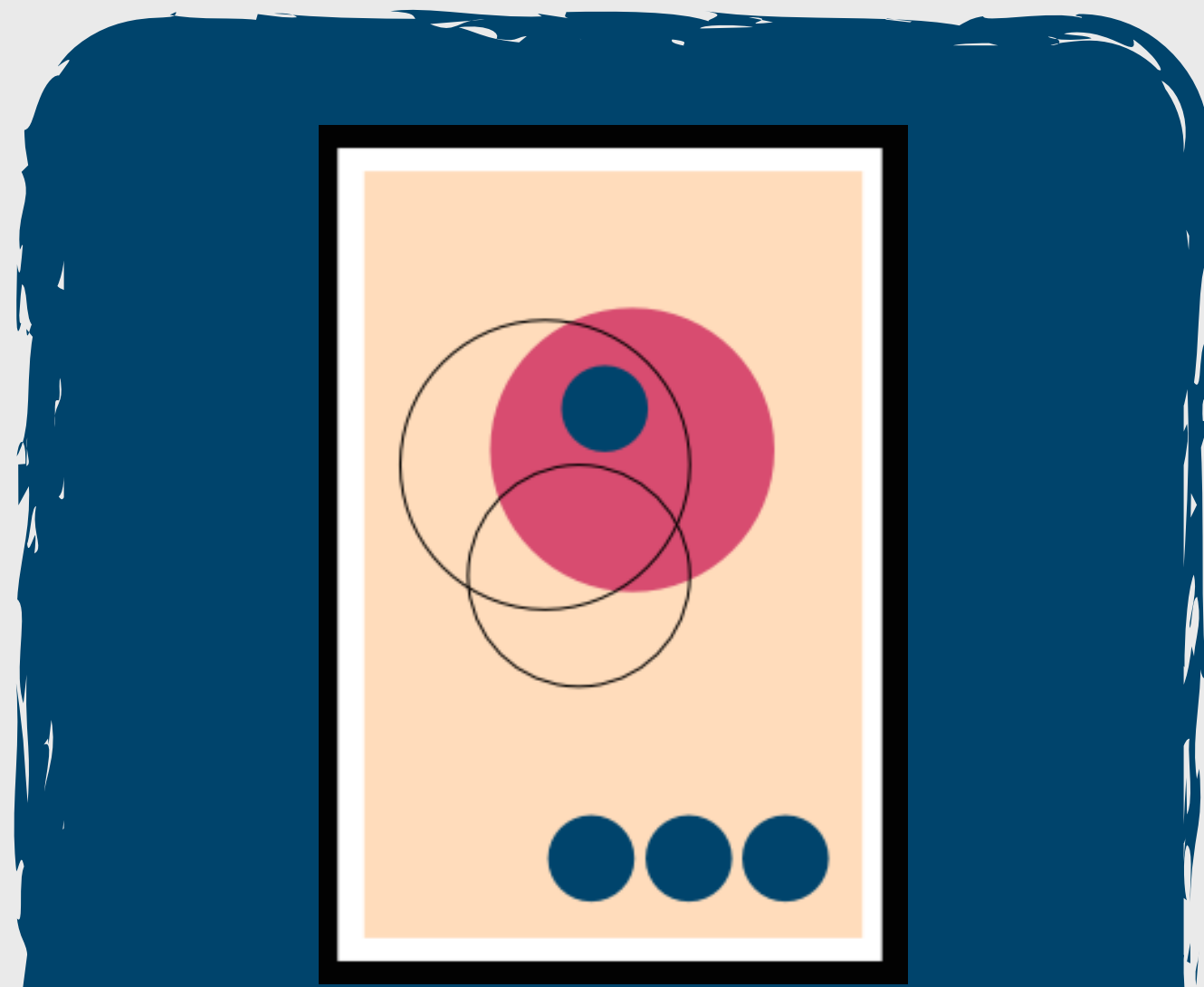


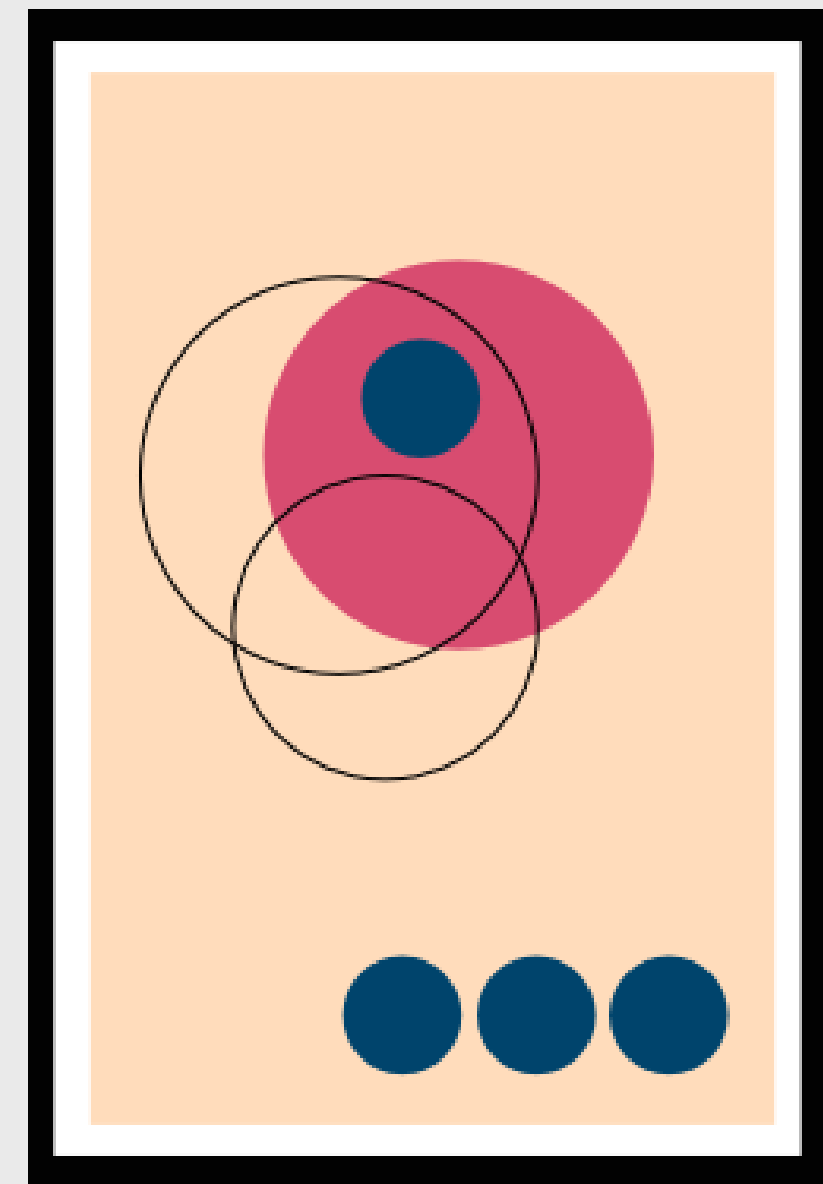
# MONA WAS ALONE



# Pourquoi Mona ?

Mona notre protagoniste s'est réveillée subitement après avoir été restaurée. Elle se rend compte qu'elle a été archivée avec de nombreux tableaux. Ne voulant pas prendre la poussière, Mona se met en quête d'une place libre où elle pourrait s'installer afin de faire partie de l'exposition.

Inspiré du jeu vidéo Thomas Was Alone, notre projet rend hommage à la peinture et prend place dans un univers digne des plus grands musées.



# GANTT

## Répartition des tâches et gestion du temps

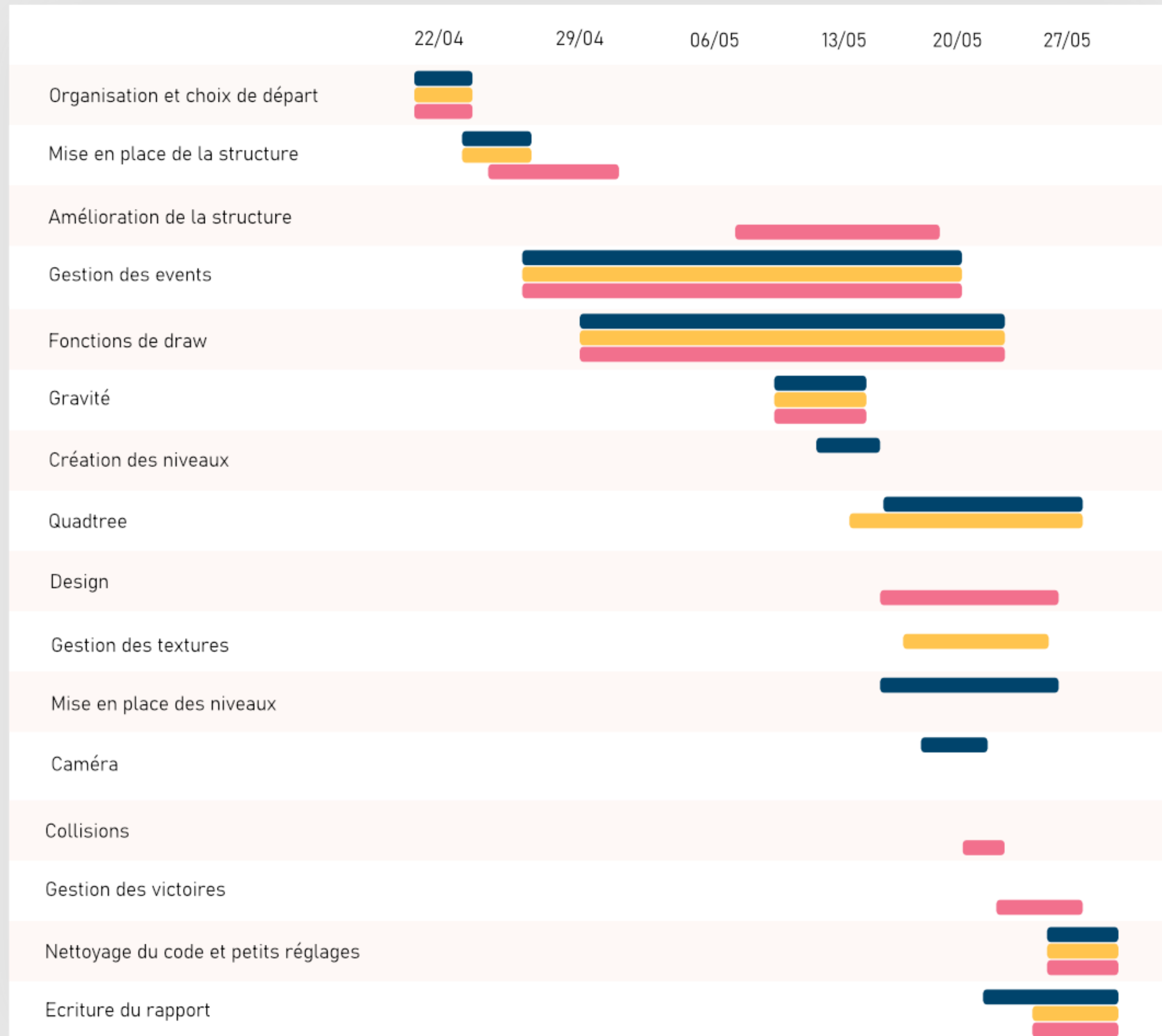
### Organisation

- \* Construire une base ensemble
- \* Répartir des tâches
- \* Sessions de code dans la même pièce
- \* Serveur discord : discussion constante

Lou

Théo

Lauriane



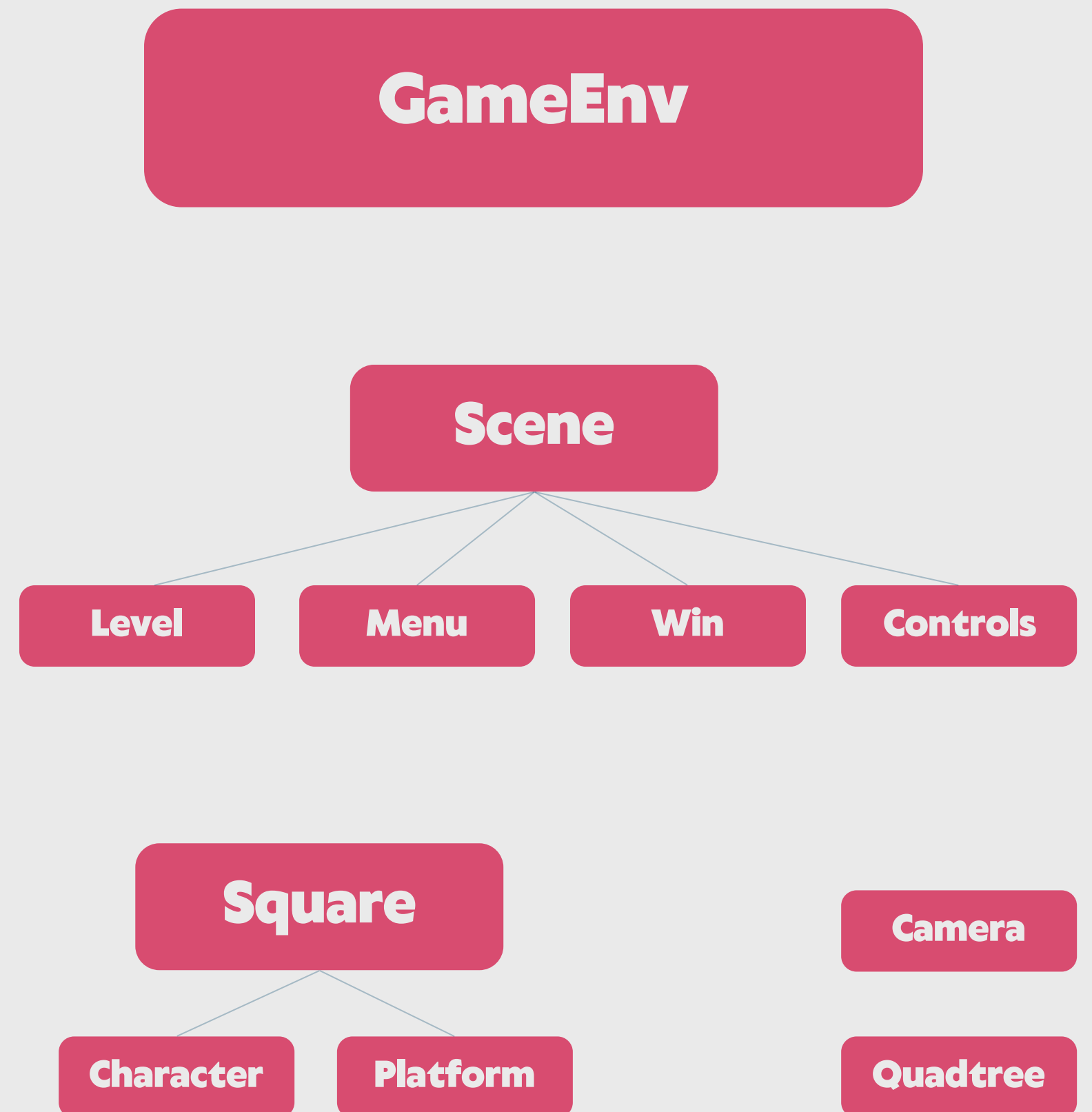
# La structure

Pour obtenir un programme le plus simple à utiliser et permettant d'implémenter de nouvelles fonctionnalités sans devoir tout remodifier, nous avons passé beaucoup de temps à penser et à coder la structure de notre jeu.

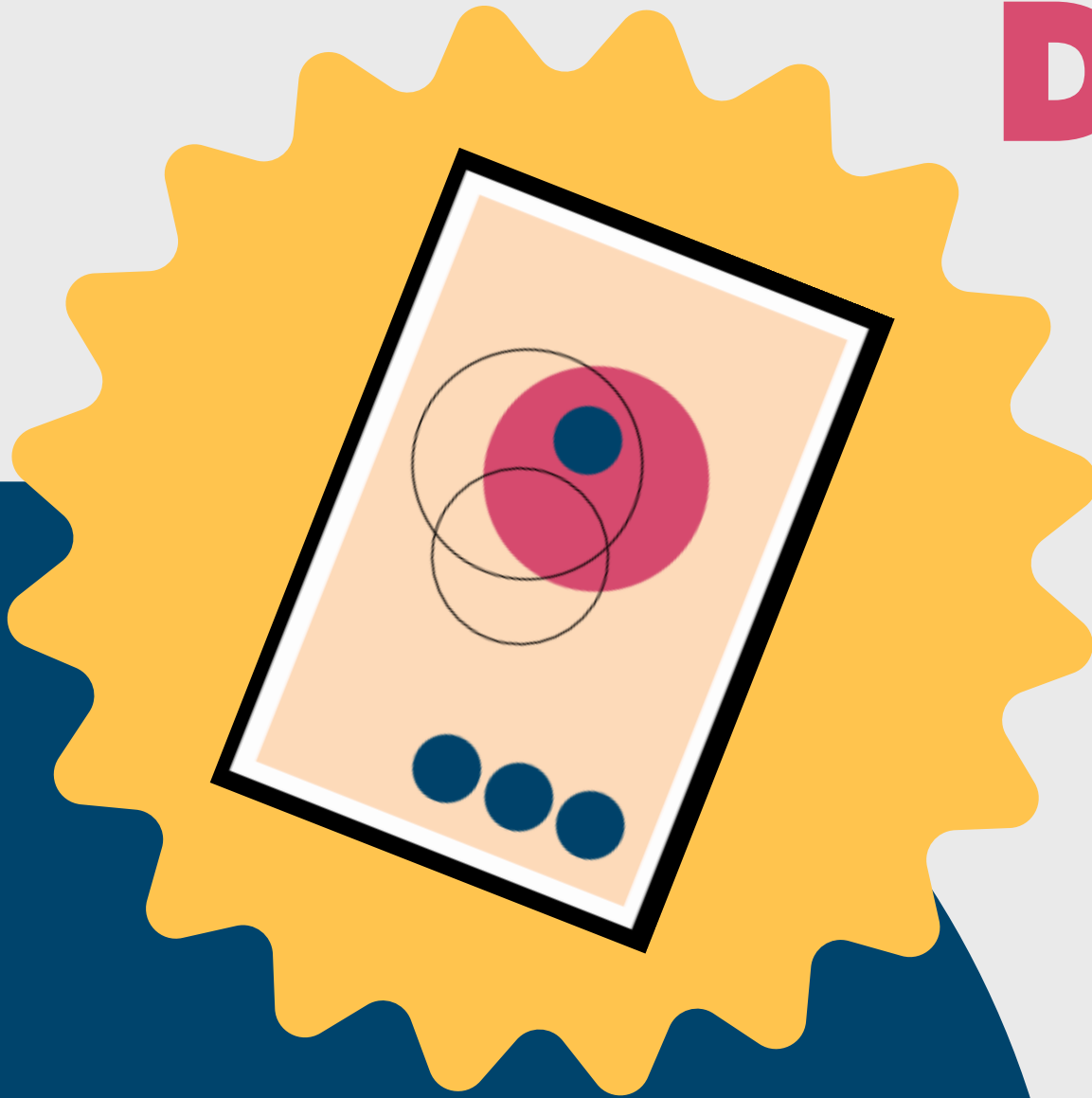
Même si nous n'avons jamais vraiment eu de cours sur les classes à l'IMAC, nous avons décidé de créer un jeu en programmation orientée objet notamment pour pouvoir utiliser les notions d'héritages.

Ainsi, le programme fonctionne via un environnement de jeu qui permet de passer facilement d'une scène à l'autre et de gérer les événements qui leur sont communs.

Cette structure permet entre autre d'implémenter très facilement de nouveaux niveaux et de nouvelles scènes.



# Difficultés rencontrées



## \* Travailler avec des classes

Comme dit précédemment la programmation orientée objet n'est pas au programme de cette première année. Pourtant au vu du travail demandé, structurer le programme avec des classes paraissait la solution adaptée. Dans notre groupe seul Lauriane en avait déjà fait, ainsi pour Théo et Lou, il était plus laborieux d'utiliser la structure du programme au départ.

## \* Reprendre le code de l'un d'entre nous.

- Pour travailler ensemble, nous avons utilisé Github. Chacun codait sur sa propre branche. Mais il était parfois compliqué de merge nos fichiers, surtout lorsque Lou et Théo avançaient le jeu et Lauriane en modifiait encore la structure.
- De plus, récupérer le code des autres ne veut pas dire le comprendre et il est arrivé que nous n'utilisions pas les fonctions créées par les autres de la bonne manière.

# Difficultés rencontrées

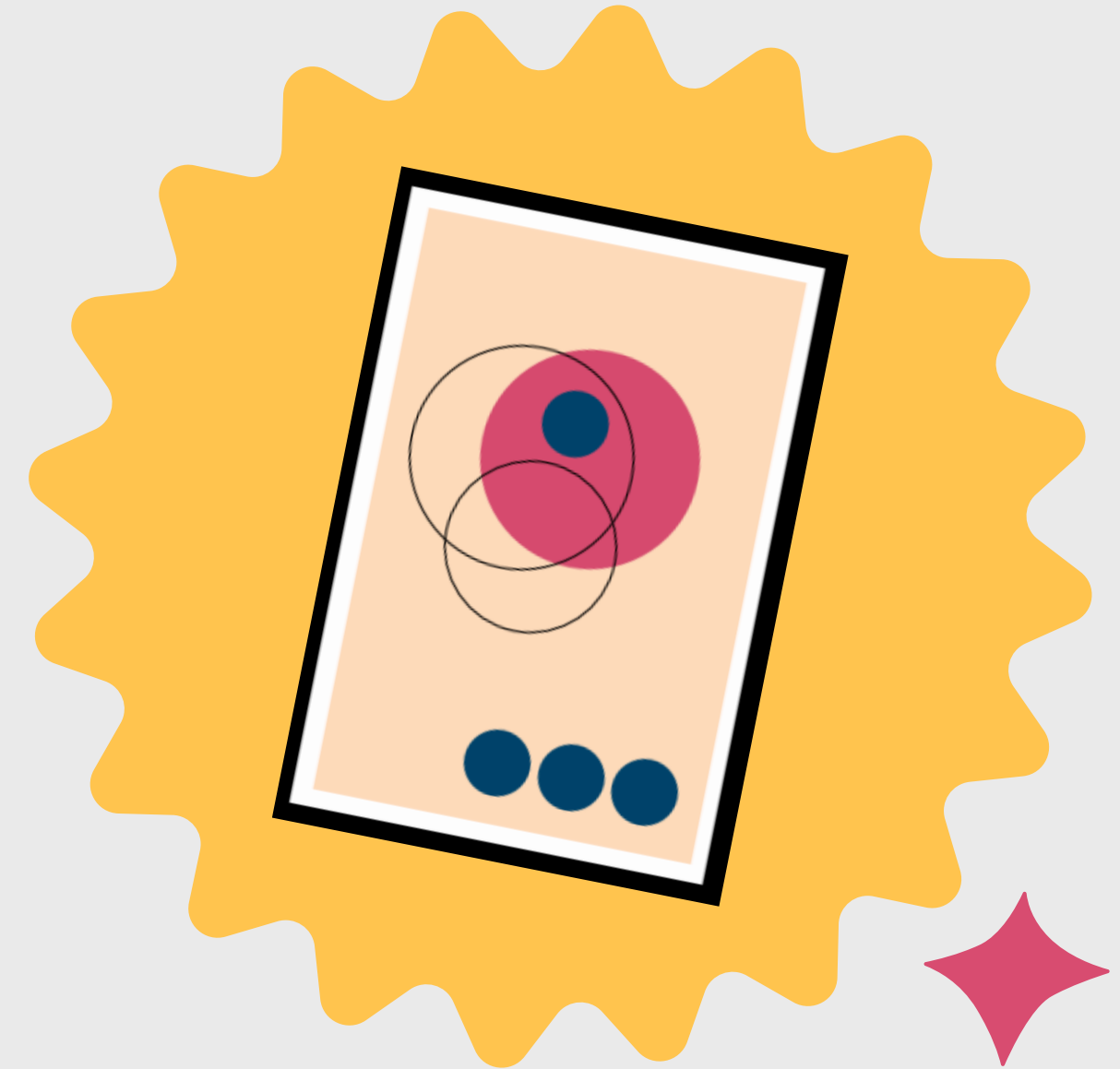
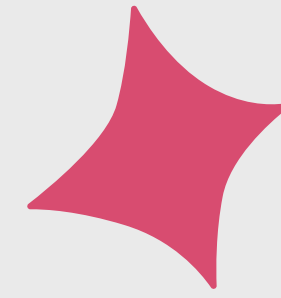
## \* Premiers pas avec un débogueur

Installer CMake, réécrire le makefile avec CMake, faire en sorte qu'il reconnaisse les librairies, se rendre compte que sur Ubuntu la version n'est pas à jour, faire en sorte qu'il trouve les images. Après toutes ces étapes, est-il judicieux de demander à tout le groupe d'installer CMake sur sa machine à cette période du projet ? Pour le reste du projet une seule personne s'occupera du débogage, mais à l'avenir il peut être judicieux de travailler d'office avec CMake.

## \* Le quadtree

Etant donné que Lauriane avait travaillé sur une bonne partie de la structure et des collisions, il nous semblait important que Lou et Théo se penchent sur le quadtree. Après de nombreux efforts, ils n'ont pas réussi à intégrer le quadtree dans le code :c



Mais croyez-nous le cœur y était... Ne pas avoir de quadtree n'est pas dramatique pour notre projet, dans le sens où les niveaux ne sont composés seulement que d'une dizaine de carré mais si la structure des niveaux venait à se complexifier, un quadtree serait essentiel.





# LE QUADTREE



- \* Nous avons un quadtree propre à chaque niveau. Au lancement d'un niveau le quadtree est appelé. Avec une fonction récursive, il divisera la zone du niveau en plusieurs parties ne contenant au plus quatre objets.
  - \* Ce quadtree aura pour but de minimiser les vérifications pour les collisions. Nous avons donc créé une méthode qui vérifie, pour chaque coin du personnage qui est en train de bouger, dans quelle "feuille" il se trouve.
  - \* La difficulté était de créer un.e tableau/vecteur/liste qui contiendrait les objets se situant dans la ou les feuilles où est situé le personnage principal. En effet si deux coins du personnage se situent dans la même feuille, alors notre tableau contiendrait deux fois les mêmes informations.
- 
- 



# Améliorations possibles

- Tout d'abord il serait vraiment agréable d'avoir un quadtree fonctionnel. Ensuite, afin d'optimiser notre code nous pourrions utiliser un quadtree qui nous retournerai seulement les éléments à dessiner. Il n'est pas nécessaire d'afficher des plateformes qui ne sont pas visibles dans la fenêtre. Aussi si la complexité de la construction des niveaux est vouée à augmenter, il peut être judicieux de stocker les emplacements des rectangles constituant le niveau dans un fichier texte et de les instancier grâce à une boucle. Nous pourrions d'ailleurs créer un champs ID\_block qui permettrait de retrouver et de stocker les blocs plus rapidement.
- Pour ce qui est du jeu, il serait intéressant de travailler sur l'éclairage de la scène, ajouter de la musique et de créer des dalles qui déclencheraient des actions (dalle qui éteint la lumière ou qui inverse les couleurs), nous pourrions aussi travailler sur des plateformes sur lesquelles les mouvements seraient chamboulés (ex: plateformes glissantes ou collantes). Pour ce qui est de l'expérience utilisateur nous pourrions faire des niveaux à temps limité ou créer un tableau qui afficherait le temps mis par le joueur à réussir le niveau.





# RETOURS SUR LE PROJET

- \* Théo: Ce projet était l'occasion de pouvoir manipuler les classes. Une notion nouvelle pour moi, non abordée en cours. Leurs fonctionnements n'est pas encore simple à assimiler, mais cela permet au moins d'en avoir une première approche.
- \* Lou: Ce projet fut le fruit d'énormément de frustration, étant novice en programmation, je n'arrive pas encore à très bien retranscrire ce que j'ai en tête. Même si je pense avoir compris la logique, j'ai énormément de soucis avec la syntaxe. Nous avons eu la chance de travailler avec Lauriane qui, très carrée, n'hésitait pas à pointer du doigts nos erreurs et mauvaises pratiques. Au final je pense avoir beaucoup appris et progresser même s'il me semble toujours avoir un train de retard en prog.
- \* Lauriane: J'ai beaucoup apprécié ce projet, même si au moment où je travaillais sur la structure je vous aurais plutôt dit le contraire. En effet, même si j'ai passé de trop nombreuses heures à traquer des erreurs, j'ai aussi beaucoup appris. J'ai consolidé mes connaissances en C++ et en programmation objet. Et quand je regarde ce que nous avons réalisé au final, je suis assez fière du résultat. J'espère juste ne pas avoir trop maltraité mes camarades en critiquant leur travail.



**Mona vous remercie!**

