

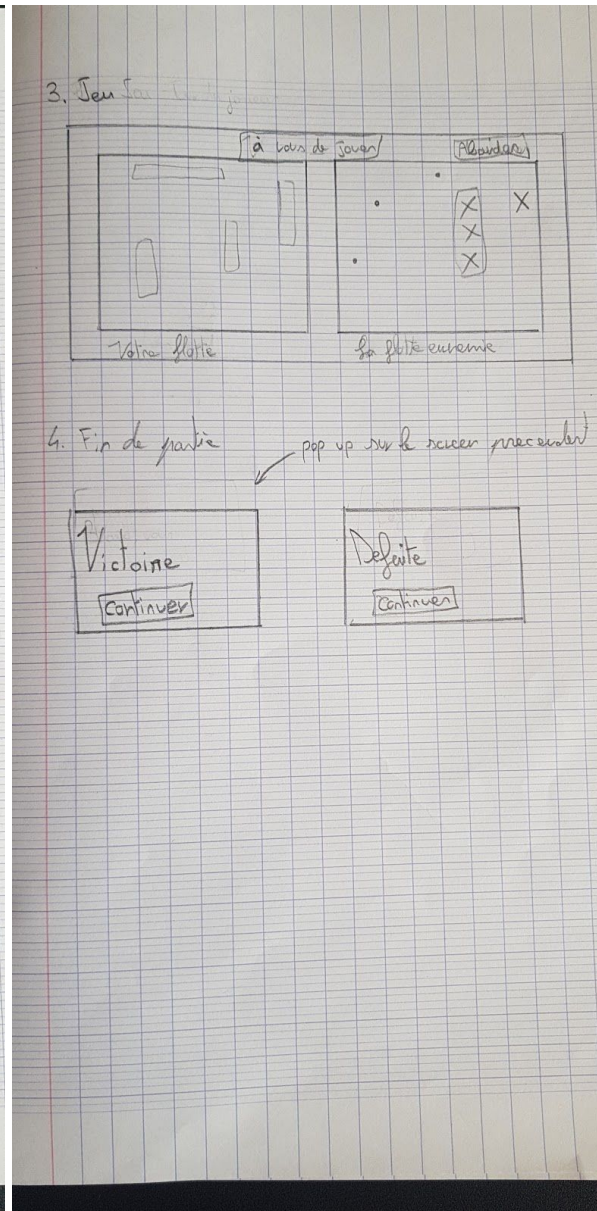
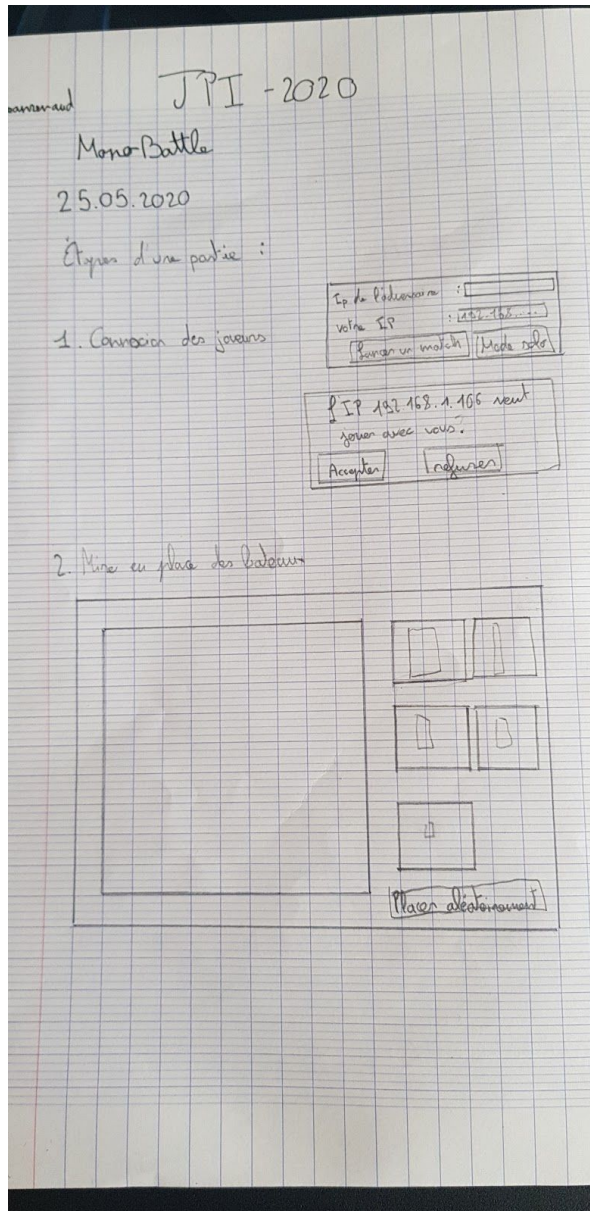
# Journal de bord

## Table des matières

25/05/2020	1
26/05/2020	3
27/05/2020	4
28/05/2020	4
29/05/2020	6
02/06/2020	6
03/06/2020	7
04/06/2020	8
05/06/2020	9
08/06/2020	9
09/06/2020	9

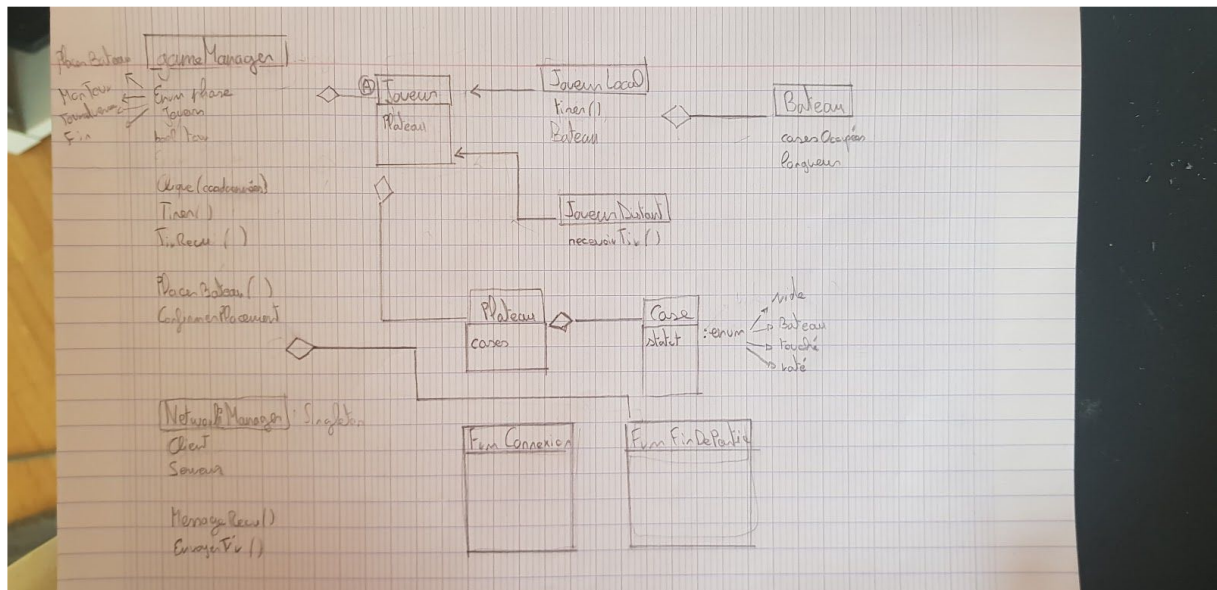
25/05/2020

Après avoir lu attentivement l'énoncé, j'ai noté les différentes étapes d'une partie type et pour chaque étape j'ai dessiné un rapide croquis de la vue. Pour m'aider j'ai fait une rapide recherche de quelque jeu de bataille navale en ligne pour me donner une idée des designs.



J'ai ensuite commencé la conception d'un premier Diagramme de classe. Une de mes décisions a été d'utiliser une machine d'état pour pouvoir séparer le jeu en phases (placement des bateaux, tour du joueur 1, tour du joueur 2, fin de la partie). J'ai également décidé que le plateau sera un tableau en 2 dimensions d'une classe square qui aura 4 états (vide, bateau, touché, raté), chaque état aura un sprite qui lui correspond.

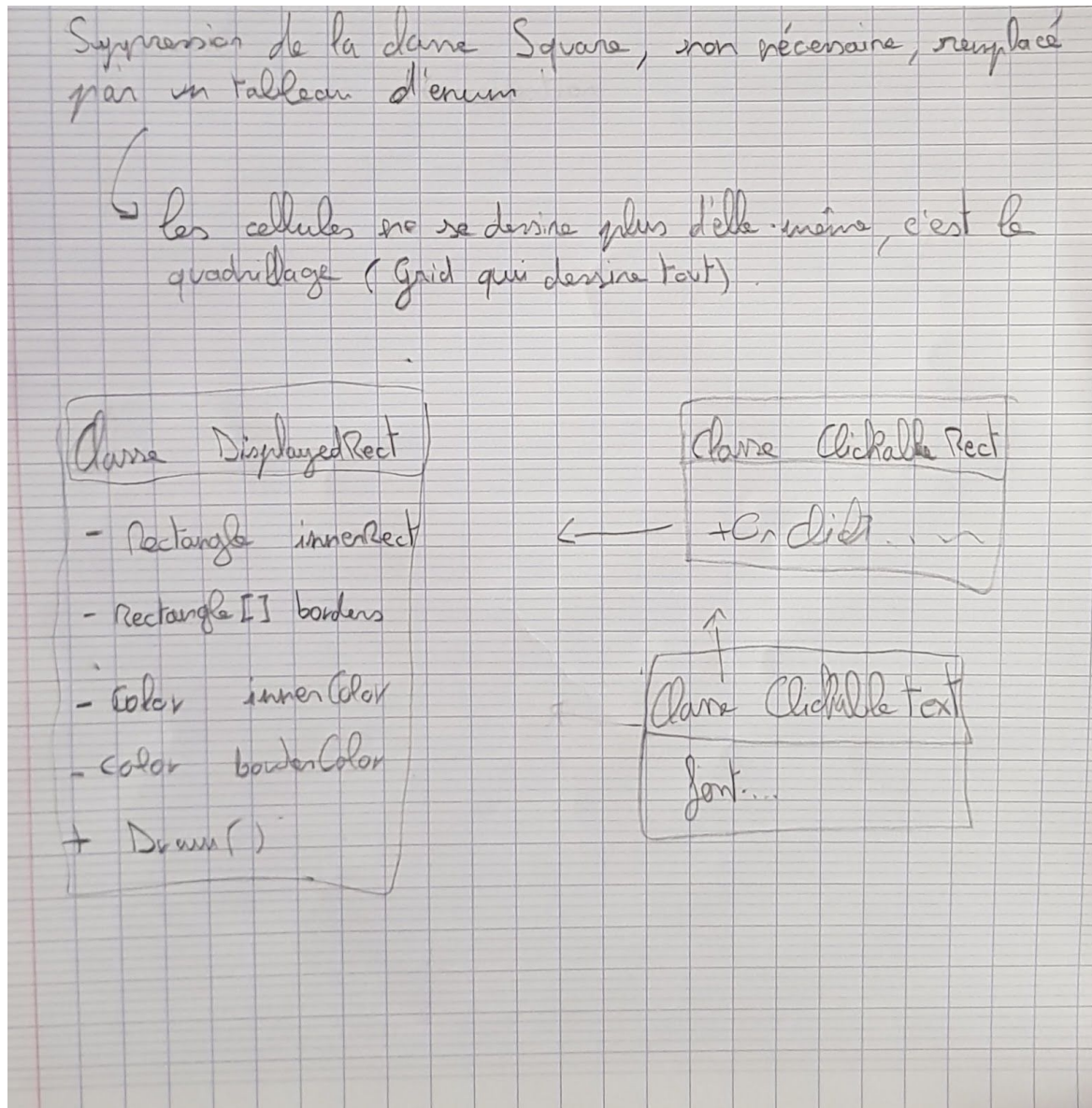
La conception du diagramme de classe m'a pris plus de temps que prévu et je n'ai pas pu faire la mise en place des tests unitaires. Je le ferai juste avant de les implémenter.



**A faire demain matin :**

Commencer l'affichage de la grille en reprenant le projet Tic Tac Toe

26/05/2020



Au final, j'ai quand même garder l'idée de classe square qui se dessine elle même plus générale pour pouvoir l'utiliser pour les éléments de l'interface (boutons, etc).

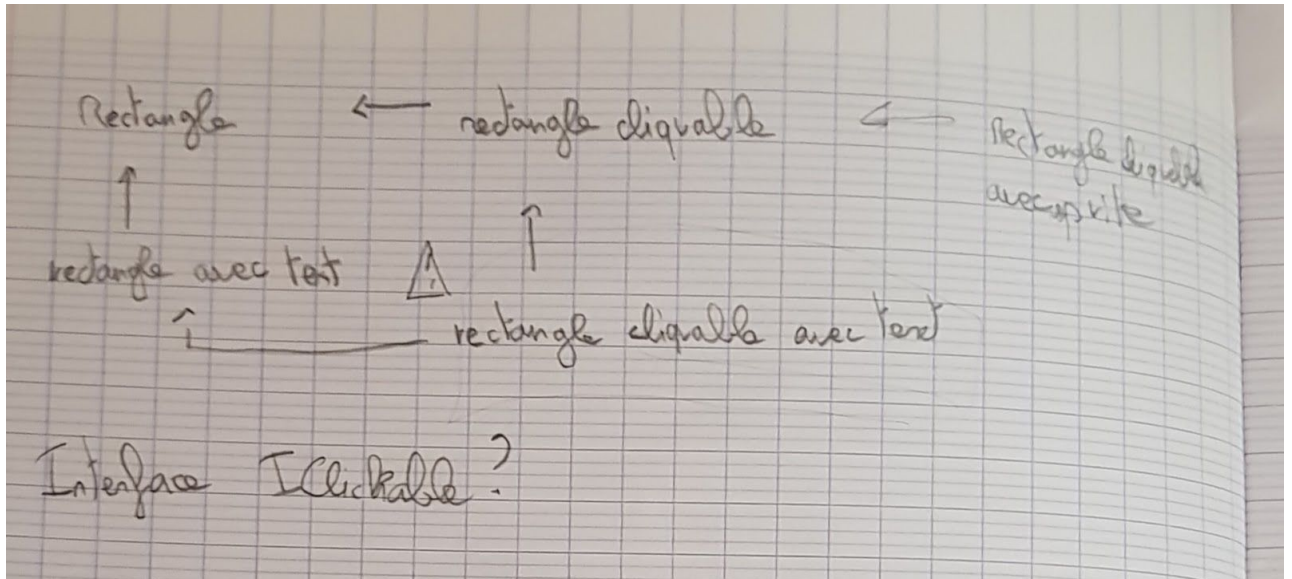
J'ai fini l'affichage des grilles et de la connexion plus tôt que prévu je vais donc pouvoir passer au placement des bateaux plus rapidement. J'ai également pu passer plus de temps sur la documentation technique.

**A faire demain matin :**

Commencer la phase de placement des bateaux



27/05/2020

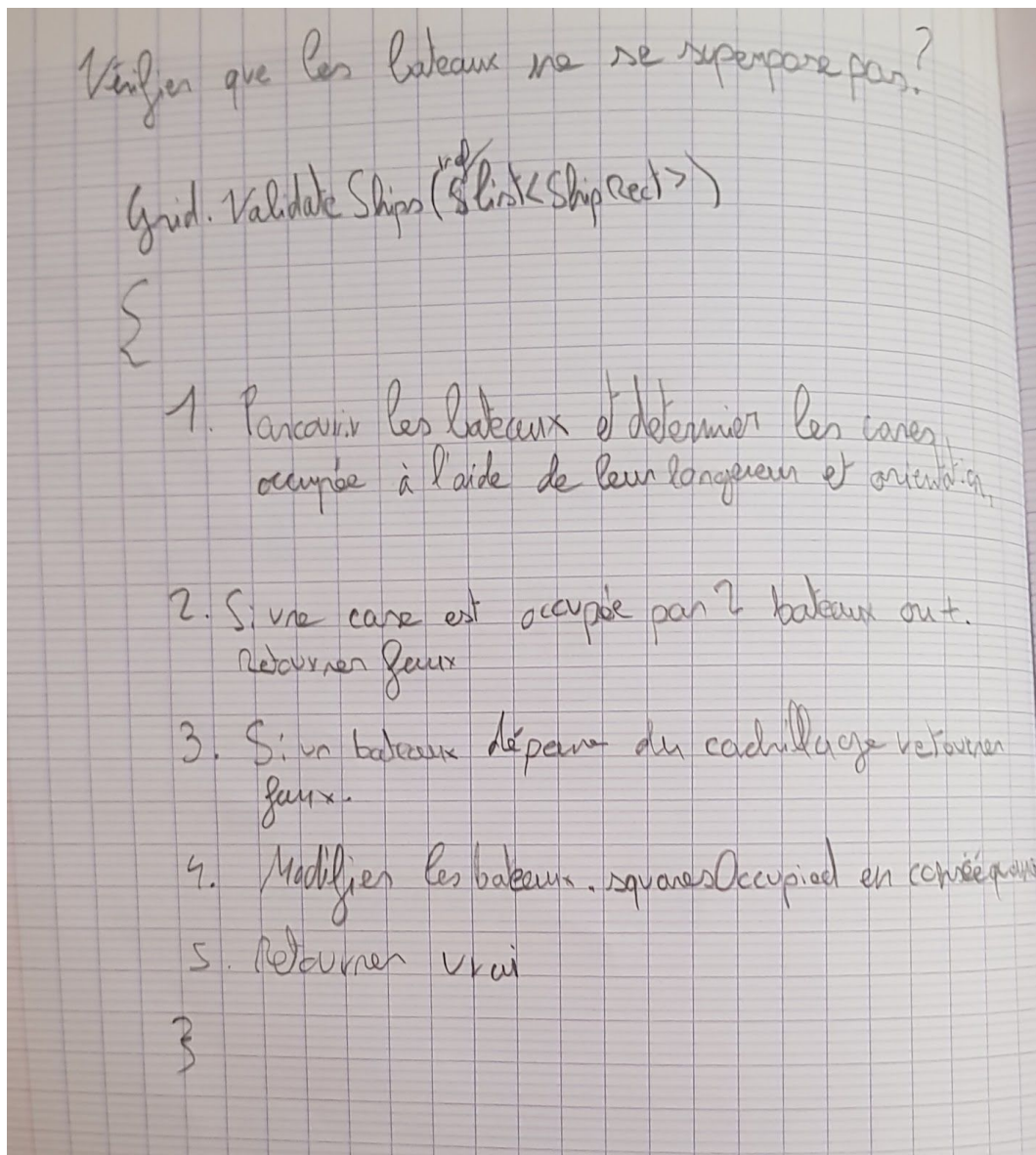


Pour les éléments d'interface j'ai remarqué que beaucoup de fonctionnalités se croisaient j'ai donc décidé d'implémenter une interface pour les objets cliquables (boutons, bateaux et cases)

**A faire demain matin :**

Mettre en place le bouton de confirmation de placement des bateaux avec la vérification de leur placement.

28/05/2020



Aujourd'hui j'ai terminé plus rapidement que prévu la vérification du placement des bateaux. Alors j'ai pris le temps de bien commenter le code et de rendre le code plus lisible.

**A faire demain matin :**

Implémenter les tours de jeu avec l'envoi de données.

## 29/05/2020

Aujourd'hui j'ai rencontré un bug bloquant, l'application crashait après environ 1 minutes d'utilisation. Après un certain temps de recherche j'ai trouvé qu'il s'agissait d'un problème de mémoire.

Maladroitement, je créais une texture à chaque frame et sans jamais libérer la mémoire de ces derniers. J'ai donc déplacé la création de la texture au lancement de l'application.

### A faire demain matin :

Trouver une solution pour que l'application communique bien que son bateau à coulé pour que l'autre puisse l'afficher.

## 02/06/2020

Pas de problème aujourd'hui, j'ai même pu finir plus de chose que prévu. Le système de tour, le bouton de reddition et la fin de partie. J'ai également pu régler certain bugs comme le fait qu'on puisse appuyer plusieurs fois sur la même case ou qu'on puisse commencer à jouer avant que l'autre joueur soit prêt.

Au niveau de la solution pour les signaler qu'un bateau à coulé, j'ai tout simplement rajouté la positions des cases du bateau après le message envoyé à l'autre application.

**Avant :** Sink

**Après :** Sink;1,5;2,5;3,5

### A faire demain matin :

Avancer la documentation et commencer le mode solo

03/06/2020

Aujourd'hui j'ai eu la rencontre avec les experts, nous avons discuté de l'avancement du projet, de la documentation et ils ont vérifié le contenu du journal de bord.

Ensuite j'ai repris le développement du mode solo. Où je suis tombé face à plusieurs problématiques :

1. Comment implémenter le bot ? De quelle classe hérite-il (Player, LocalPlayer, OpponentPlayer) ?
2. Pour le placement aléatoire des bateaux j'ai créé une fonction récursive pour s'assurer que le placement est correct. Mais elle génère une exception Stack Overflow parfois.

```
void PlaceRandomShips() {  
    Random rnd = new Random();  
  
    Pour chaque Bateau du joueur {  
        si (rnd.Next(2) < 0) {  
            bateau.rotation();  
        }  
  
        bateau.position = (nbAléatoire ; nbAléatoire)  
    }  
  
    // entre point le plus à gauche et  
    // plus à droite du cadrillage.  
  
    si (configuration Invalid) {  
        PlaceRandomShip();  
    }  
}
```

A faire demain matin :

Trouver des solutions au 2 problématiques qui bloque le développement du mode solo



04/06/2020

Pour la problématique du [Stack Overflow de hier](#), j'ai trouvé [ce thread sur stackoverflow.com](#) qui parle de ce sujet. D'après le commentaire de FarmerBob, C# n'est pas optimisé pour de la "tail-recursion" (Quand la dernière action d'une méthode rappelle cette dernière). La solution proposée est d'implémenter une boucle à la place d'une fonction récursive.

```
public void PlaceShipsRandom()
{
    do
    {
        Random rnd = new Random();
        foreach (ShipRect ship in _ships)
        {
            if (rnd.Next(2) == 0)
            {
                ship.Rotate();
            }
            ship.Position = FindClosestSquareAnchor(new Point(
                rnd.Next(_playerGrid.AnchorPoint.X, _playerGrid.AnchorPoint.X + ( NB_SQUARE_ROW * SQUARE_SIZE )),
                rnd.Next(_playerGrid.AnchorPoint.Y, _playerGrid.AnchorPoint.Y + ( NB_SQUARE_ROW * SQUARE_SIZE ))));
        }
    } while (!ValidateShips(false));
}
```

C'est ce que j'ai fait et le problème semble résolu, j'ai quand même implémenté un bloc try catch au cas où il n'y aurait pas de configuration possible.

```
try
{
    _localPlayer.PlaceShipsRandom();
}
catch (StackOverflowException)
{
    WF.MessageBox.Show("Error", "No possible configuration");
}
```

Ensuite, j'ai terminé l'implémentation du mode solo. Au final, j'ai créé une classe BotPlayer qui hérite du localPlayer avec une fonction shoot qui imite le tir d'un joueur. Je lui ai donné un champ positionDejaTesté qui est une liste de point dans laquelle se rajoute les positions après avoir été "shoot".

Cela termine l'intégralité des fonctionnalités du programme, ne reste plus que la documentation et les tests.

**A faire demain matin :**

Relire l'intégralité du code, refactoriser si nécessaire, en particulier le clic des boutons.

**05/06/2020**

Aujourd'hui j'ai relu l'intégralité du code et j'ai ajouté plus de commentaires. J'ai également supprimé certaines méthodes redondantes comme FindClosestAnchorPoint de la classe player et j'ai supprimé la classe bouton car elle était inutile au final. J'ai pu bien entamer les tests unitaires qui devraient être finis lundi matin.

**A faire demain matin :**

Finir les tests unitaires et finaliser la documentation

**08/06/2020**

J'ai pu finir la totalité des tests unitaires et une grande partie de la documentation technique, j'ai laissé quelques détails comme la correction de mise en page, l'ajout de meilleures images à faire pour le dernier jour.

**A faire demain matin :**

Finaliser la documentation (encore) et faire le résumé du rapport TPI.

**09/06/2020**

J'ai pu terminer les modifications de la documentation et j'ai terminé de rédiger le résumé du TPI avant midi. J'ai pu prendre l'après-midi pour vérifier les documents et préparer la reddition.