TELECOM Nancy

Projet Pluridisciplinaire d'Informatique Intégrative

MADLE (a WORDLE like game)

Thibault BOISSEAU
Guillaume BOURGEON
Tanguy BOURRA
Aurélien TRONCY

Responsable de module :
Olivier Festor

Table des matières

Chapitre 1

Introduction

1.1 Contexte

Le projet présenté dans ce rapport a été réalisé dans le cadre du module PPII-2 (Projet Pluridisciplinaire d'Informatique Intégrative du second semestre) afin d'appliquer dans un cadre concret les connaissances et les savoirs acquis en gestion de projet au travers du MOOC de Centrale Lille ainsi qu'en algorithmie, en gestion et création de base de données, en développement Web et en structure de données en rapport avec les modules CS54, SD, C. Ce projet est un module à part entière de la première année de la formation sous statut étudiant du cycle ingénieur de TELECOM Nancy.

L'objectif de ce projet est de développer une application web similaire au jeu Wordle (retrouvable ici) basée sur une base de données et des algorithmes de traitement avancés. Le second objectif de ce projet est de concevoir, dans le langage de programmation C, un solveur pour l'application précédemment développée. Nous avons donc découpé notre travail selon 5 grands axes : Développement du jeu, Base de données, Développement Web, Algorithmie pour le solveur et Gestion de projet.

1.2 Organisation du document

Le rapport est composé d'une introduction (chapitre actuel), de 8 chapitres ainsi que d'une conclusion suivie d'une annexe.
Le chapitre 2 exposera les différents outils de gestion de projet que nous avons utilisés.
Le chapitre 3 mettra en exergue les différents algorithmes utilisés dans notre application concernant le fonctionnement du jeu.
Le chapitre 4 présentera la conception et l'implémention de la base de données utilisée pour notre application.
Le chapitre 5 développera la partie sur le développement web et le schéma du site.
Le chapitre 6 présentera les complexités des différents algorithmes ainsi que les tests.
Le chapitre 7 est consacré au solveur du jeu Wordle.
La conclusion consistera en un bilan du projet à la fois d'un point de vue personnel mais aussi d'un point de vue global.

1.3 Précisions légales

L'ensemble des données utilisées ont été inventées pour le test de notre application, toute ressemblance avec des personnes existantes ou ayant existées est fortuite.
Ce projet n'est pas destiné à un usage commercial et est à caractère strictement scolaire, ainsi, la réutilisation de code qui n'est pas libre de droit nous est possible en accord avec :

- Code civil : articles 7 à 15, article 9 : respect de la vie privée
- Code pénal : articles 226-1 à 226-7 : atteinte à la vie privée
- Code de procédure civil : articles 484 à 492-1 : procédure de référé
- Loi n78-17 du 6 janvier 1978 : Informatique et libertés, Article 38

Toutes les réutilisations de code sont listées ci dessous :

- Code LaTeXinspiré du rapport de projet "Jouons à Cache-Cache" de *Chanteloup Marie-Astrid, Reszetko Clément et Sepe Benjamin* réalisé en 2019 pour le module TOP à TELECOM Nancy.
- Code pour les Checkboxes du site web, inspiré par le Code de creativejuiz

1.4 Présentation du jeu WORDLE

Wordle est un jeu sorti en 2021 et développé par Josh Wardle. Il est une adaptation du jeu télévisé américain Lingo qui est l'équivalent de notre Motus national. Ce jeu est très rapidement devenu populaire autour du monde, principalement grâce à Twitter, qui permettait à tous les joueurs de montrer leurs score du jour. Car en effet, une des particularités de Wordle est qu'il n'y a qu'un seul mot à deviner par jour et ce mot est le même pour tous les joueurs. Certains dérivés de ce jeu sont très vite arrivé sur le marché avec par exemple SUTOM. En janvier 2022, le New York Times décide de racheter le jeu pour plusieurs millions de

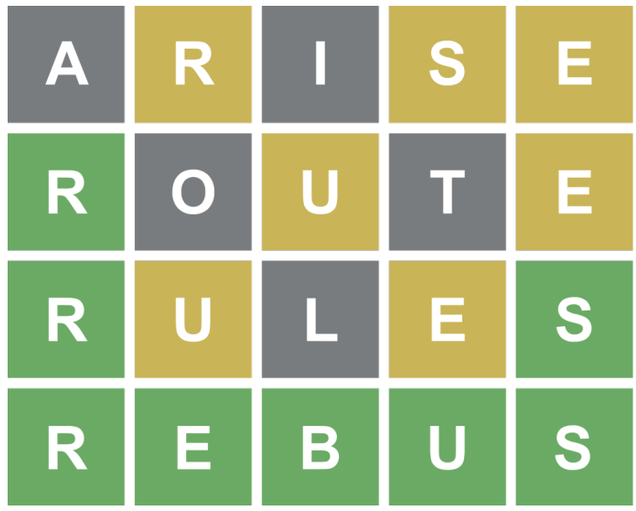


FIGURE 1.1 – Grille de Wordle (Source : https://fr.wikipedia.org/wiki/Wordle)

dollars.

Le but du jeu est de trouver un mot de cinq lettres à partir de rien et cela en moins de 6 coups. A chaque coup le joueur rentre un mot de 5 lettres qui est dans le dictionnaire. Le jeu répond à l'aide d'un code couleur. Gris si la lettre n'est pas dans le mot final, jaune si la lettre est dans le mot mais pas à l'emplacement auquel elle est actuellement et vert si la lettre est dans le mot et au bon emplacement. Néanmoins il existe quelques subtilités. Si la même lettre apparaît plus d'une fois dans le mot alors un ordre de priorité des informations se met en place. Pour la suite de l'explication nous allons prendre l'exemple de deux occurrences de la même lettre. Il est facile de généraliser cela à plus de deux occurrences de la même lettre.

1. Si le mot final ne contient qu'une seule fois la lettre et qu'une des deux occurrences est au bon endroit alors celle au bon endroit devient vert et l'autre reste grise
2. Si le mot final ne contient qu'une seule fois la lettre et qu'aucune des occurrences est au bon endroit alors la première occurrence sera colorée en jaune et la deuxième reste grise

Ainsi, une fois ces quelques règles connues, on peut rapidement prendre en main le jeu et tenter sa chance.

Chapitre 2

Gestion de projet

2.1 Équipe projet

L'équipe projet est composée de quatre membres, tous étudiants en première année à TELECOM Nancy :
— Thibault BOISSEAU
— Tanguy BOURA
— Guillaume BOURGEON
— Aurélien TRONCY

L'équipe a décidé de mettre en place une structure fonctionnelle avec coordinateur pour annimer les réunions et trancher sur certains points qui ne font pas l'hunanimité. Ce système à permis une implication importante de chaque membre tout en ne perdant pas de temps lors des questions sur les points de détails.

Les réunions ont eu lieu principalement sur la plateforme Discord, donc en distanciel. Chaque séance était présidée par Aurélien Troncy. La prise en note et la rédaction du compte-rendu était faite par un des trois autres membres, chacun leur tour.

L'environnement de travail choisi pour la partie programmation est visual studio code. Pour la base de données nous avons utilisé sqlite3, pour l'application web le framework Flask et python pour les algorithmes. En ce qui concerne le solveur, il a été écrit en C et compilé avec Clang. Les documents ont été écrit en LaTeXsur la plateforme overleaf. Enfin, pour la gestion des versions, nous avons utilisé gitlab.

2.2 Analyse du projet

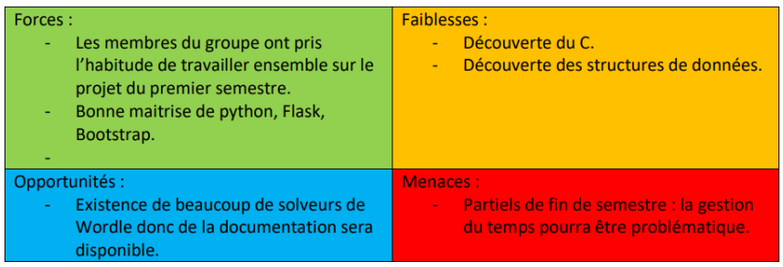
2.2.1 Définition des objectifs

L'ensemble des objectifs de ce projet a été défini à l'aide de la méthode SMART :

Abréviation	Mot abrégé	Signification
S	Spécifique	L'objectif doit être simple, clair, précis
M	Mesurable	Il doit comporter un indicateur numérique
A	Accepté	L'objectif ne doit pas être imposé
R	Réaliste mais ambitieux	Il ne doit être ni trop facile ni trop complexe
T	Défini dans le temps	Il doit avoir une date de fin

2.2.2 Matrice SWOT

La matrice SWOT à été réalisée au début du projet pour permettre à l'équipe de prendre conscience de ses forces, ses faiblesses, des oppportunités et des menaces.



2.2.3 Matrice RACI

Une fois le WBS (Work Breakdown Structure) fait, les lots de travail ont été répartis entre les différents membres de l'équipe projet et les membres extérieurs à celui-ci selon 4 attributs : Réalise (R), Autorité (A), Conseils (C) et Informe (I). Une même personne peut avoir plusieurs attributs mais il doit y avoir, pour chaque lot de travail, au moins un R et un seul et unique A.

cf.Annexe page 41

2.2.4 WBS

Le WBS (Work Breakdown Structure, Structure de répartition du travail en français) est un outil permettant de visualiser l'ensemble des tâches définies en amont d'un projet. Elles y sont hiérarchisées et la réalisation de chacune d'elles mène à la réalisation totale du projet.

cf.Annexe page 40

2.2.5 Diagramme de Gantt

Le diagramme de Gantt est un outil de gestion de projet qui permet de visualiser l'avancement de ce dernier dans sa globalité et ce pour chaque tâche. Il décrit les tâches à réaliser ainsi que le temps nécessaire à leur consacrer. Le diagramme ci-joint est le dernier réalisé pour le projet. En rouge, se trouve le chemin critique. C'est à dire l'ensemble des tâches indispensables à la réussite du projet.

cf.Annexe pages 42 et 43

Chapitre 3

Algorithmie du jeu

3.1 Introduction

Bien que l'interface web gère l'affichage et la récupération des valeurs, la gestion des différents coups donnés par le joueur est gérée en back-end avec du code python. L'objectif étant de traiter des données et de respecter les spécificités des règles du jeu Wordle.

3.2 Fonctionnement

Au début de chaque partie, un algorithme python va prendre un mot du dictionnaire de manière aléatoire en respectant les paramètres décidés par le joueur. Dans notre cas, seule la taille du mot est choisie par le joueur. Ensuite, à chaque proposition du joueur, l'algorithme va dans un premier temps vérifier si le mot appartient au dictionnaire. Si c'est le cas, l'algorithme va comparer le mot proposé à celui qu'il faut deviner. L'algorithme renvoie alors une liste de couleurs respectant la règle suivante : vert si la lettre est bien placée, orange si la lettre n'est pas bien placée et noir si la lettre n'est pas dans le mot. D'autres algorithmes permettent la gestion du jeu, notamment ceux permettant de gérer les statistiques. Tout le reste sera géré par l'application web.

3.3 Généralité

Pour fonctionner, des variables globales sont générées grâce à un fichier json contenant l'ensemble des mots du dictionnaire par taille de mots. Ces dernières sont des listes contenant des mots d'une taille précise, les mots étant triés selon l'ordre alphabétique, permettant ainsi de faciliter les calculs. Le reste est géré par les algorithmes suivants.

get_words_json(filename : str) -> dict

Prend en argument une chaine de caractère qui correspond au chemin du fichier .txt qui contient les mots du dictionnaire. La fonction retourne un dictionnaire contenant tous les mots du fichier .txt par taille de mots.

get_list(taille_mot : int, dico_json : dict) -> list

Prend en argument un entier n qui correspond à la taille des mots recherchés et un dictionnaire contenant tous les mots. La fonction retourne la liste de tous les mots de taille n du dictionnaire.

choix(longueur : int) -> str

Prend en argument un entier (la longueur du mot) et renvoie une chaine de caractères correspondant à un mot de cette taille, pris au hasard grâce à la fonction randint.

verif_dico(mot : str) -> bool

Prend en argument une chaine de caractères correspondants à la proposition du joueur et renvoie un booléen (true si le mot est dans le dictionnaire, false sinon).

occurrence(mot : str) ->tuple

Prend en argument une chaine de caractères correspondant à la proposition du joueur et renvoie deux listes. La première est une liste comprenant toutes les lettres du mot sans doublon. La deuxième donne le nombre de fois que ces lettres apparaissent dans le mot. Cette fonction est très utile pour la fonction correction.

correction(a_trouve : str, proposition : str) -> list

Prend en argument deux chaines de caractères correspondant au mot à trouver et au mot proposé. Cet algorithme permet de comparer les deux mots afin de renvoyer la suite de couleurs décrite précédemment.

Dans un premier temps, si les deux mots sont bien de même longueur, on applique la fonction "occurrence" au mot à trouver pour repérer les différentes occurrences de chaque lettre, et ainsi, pouvoir éliminer celle qui sont en trop dans le mot proposé. Ensuite toutes les lettres seront colorées en noir, puis en parcourant une première fois la liste on regardera les lettres bien placées que l'on coloriera en vert. Les lettres mal positionnées seront coloriées en orange. La fonction "occurence" permet de verifier les lettres qui sont plus représentées dans le mot proposé que dans le mot à trouver, celles-ci resteront donc noires.

moyenne_nombre_coups(parties : list) -> float

Prend en argument une liste de parties (obtenue par requête SQL) puis retourne la moyenne du nombre de coups effectués pour chaque parties gagnée.

moyenne_taille_mots(mots : list) -> float

Prend en argument une liste de parties (obtenue par requête SQL) puis retourne la moyenne des tailles de mots à deviner sur l'ensemble de ces parties.

get_chart(moyenne : float) -> str

Prend en argument un nombre flottant x qui correspond à un pourcentage (entre 0 et 100) et retourne une chaine de caractères qui correspond au chemin du fichier .png du graphique x pourcent.

Chapitre 4

Base de données

4.1 Introduction

L'ensemble du travail fait sur la partie base de données a été réalisé avec le logiciel SQLite3 ainsi qu'avec le module python du même nom. La base de données est utilisée dans notre application pour la gestion de la connexion des utilisateurs, pour l'enregistrement des différentes parties des joueurs, pour les paramètres enregistrés et les succès gagnés par ce dernier.

4.2 Principe d'une base de données relationnelle

Une base de données relationnelle peut être vue comme un ensemble de tables et de relations. Une table est un ensemble de lignes (tuples) ou de colonnes (attributs). Elle permet de stocker des données de manière structurée et d'y accéder facilement.

4.3 Conception de la base de données relationnelle

La base de données a été conçue dans un premier temps sans la table "paramètre", puis a été mis à jour suite aux remarques du responsable de projet et des besoins de l'application. Les noms des relations sont en gras, les clefs des relations sont soulignées et les clefs étrangères sont précédées d'un "#".

Utilisateur (id_joueur, pseudo, mdp, mode sombre, path_pic)
Un utilisateur possède un identifiant unique, un pseudo et un mot de passe. Il choisit s'il veut voir le site en mode sombre ou non et possède une photo de profil. Tous les attributs sont non nuls.

Partie (id_partie, #id_joueur, #id_parametre, victoire, a_trouver, coup_1, coup_2, coup_3, coup_4, coup_5, coup_6, coup_7, coup_8, coup_9, coup_10, coup_11, coup_12, coup_13, coup_14, coup_15)
Une partie est toujours associée à un joueur. Ce dernier définit le nombre de coup et le nombre de lettres du mot à trouver. Le joueur gagne s'il trouve le mot en un nombre de coup inférieur au nombre définit précédemment, il perd sinon. Chaque partie pouvant se faire en un nombre de coup différent, on se limite à 15 coups maximum dans les paramètres.

Parametre (id_parametre, nb_coups, nb_lettres)
Les paramètres sont définis de manière a laisser le joueur choisir la taille du mot à chercher (La taille d'un mot est comprises entre 1 et 15). Il peut également définir le nombre de coups qu'il veut se donner pour le trouver.

Statistiques (#id_joueur, nb_parties, nb_victoires)
Les statistiques sont liées à un joueur et sont mises à jour à chaque partie. Ainsi le joueur peut connaitre le nombre de parties qu'il a joué mais également le nombre parties gagnées.

Succes (#id_joueur, succes1, succes2, succes3, succes4)
Les succès, comme les statistiques sont associés à un joueur, et relatent des différents accomplissements du joueur (a joué plus de 100 parties, victoire en moins de 2 coups, ...).

La base de données ainsi construite a toutes ses relations en troisième forme normale, ce qui veut dire que pour chaque relation :

— chacun de ses attributs est atomique et mono-valué
— tout attribut n'appartenant pas à la clé ne dépend pas d'une partie de la clé
— tout attribut n'appartenant pas à la clé ne dépend pas d'un autre attribut n'appartenant pas à la clé

Une base de données en troisième forme normale permet de s'assurer que nous n'auront pas de problème de mise à jour ou de redondance des informations.

4.4 Implémentation

L'implémentation de la base de données a, dans un premier temps, été développé sans la table paramètre, avec seulement la table utilisateur, la table partie, la table statistique et la table des succès. La table paramètre a finalement été rajoutée pour mettre la base de données en 3ème forme normale. Cette dernière est alimentée via l'interface web. Les données relatives aux joueurs sont mises à jour à chacune des parties.

4.5 Difficultés rencontrées

L'implémentation de la base de données n'a pas posé de problème en elle-même. Cependant, la longueur de la table "parties" où sont recensés les 15 coups du joueur a posé quelques difficultés dans l'écriture du code, dans la récupération des données et dans l'enregistrement de ces dernières dans la table.

Chapitre 5

Application Web

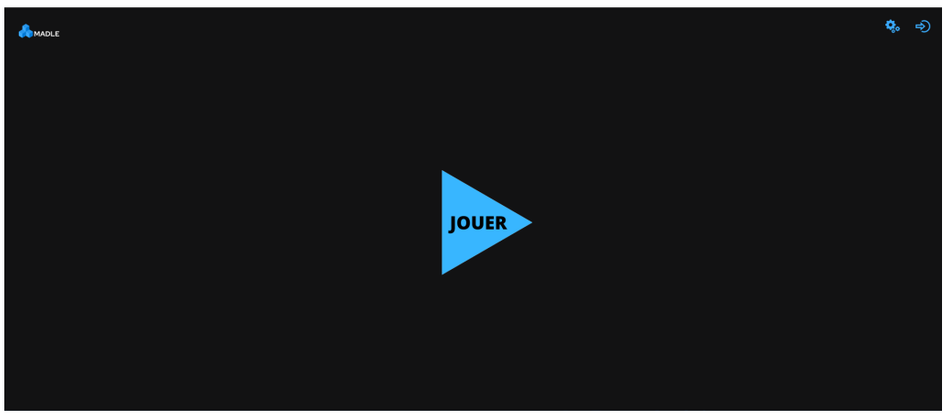
5.1 Introduction

Afin qu'elle soit la plus attrayante et efficace possible, l'application web a nécessité plusieurs outils qui ont permis de mettre en oeuvre l'affichage des données. Le site et son aspect graphique ont été créés à l'aide du framework python Flask, du langage HTML qui est complétée deux fichiers CSS pour le design du site.

5.2 Pages web

5.2.1 Page d'accueil

Lorsque l'utilisateur arrive sur le site il est directement envoyé sur cette page. Son design est très épuré pour que l'utilisateur ne se sente pas perdu en arrivant sur le site. Il est possible de jouer au jeu en cliquant sur le bouton jouer, de se connecter ou d'aller sur son profil si on est déjà connecté. Une des dernières features que cette page propose est le lien vers la page paramètre.

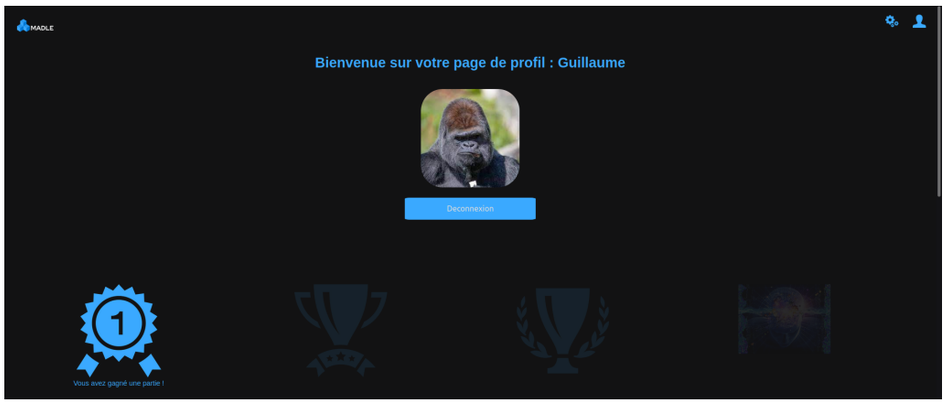


5.2.2 Page de Connexion

Cette page permet à l'utilisateur de se connecter et donc d'accéder à son profil une fois connecté. L'utilisateur a donc l'obligation de se connecter pour aller voir son profil. Certaines fonctionnalités nécessitent l'identifiant de la personne connectée, il existe un utilisateur 'NULL' qui sert à récupérer les parties que les membres non inscrit ont joués. Néanmoins, il ne possède pas de profil et personne ne peut se connecter à ce compte. Pour se connecter il est nécessaire d'écrire son nom d'utilisateur ainsi que son mot de passe. Toute erreur de non-conformité des identifiants entraîne une redirection vers la même page mais avec un message d'erreur.

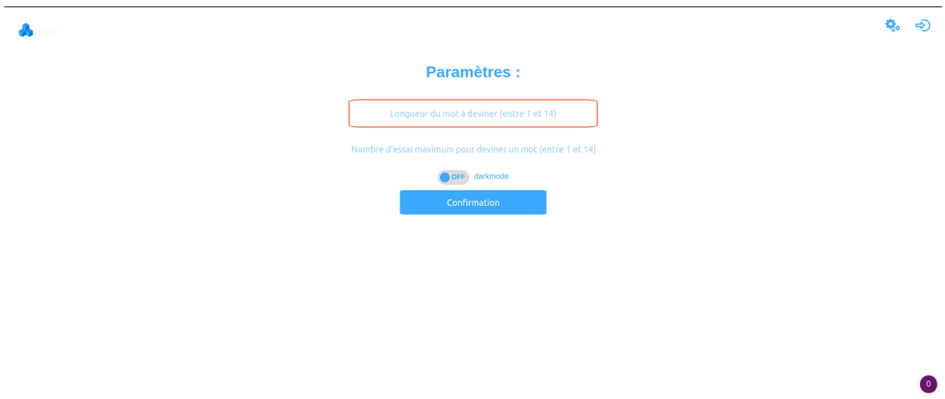
5.2.3 Page de profil

Cette page est réservée aux utilisateurs qui se sont connectés. Ils peuvent ainsi voir leur pseudo, photo de profil mais surtout le plus important, leurs succès, leurs statistiques générales et l'historique de leurs parties. L'historique rappelle le mot qui était à deviner ainsi que le nombre d'essais permis puis, enfin, les différents coups effectués par le joueur. Si l'utilisateur décide de partager une de ses parties il peut cliquer sur le N° de partie correspondant et il sera ammener sur le replay de cette partie. Ainsi il n'aura qu'a envoyer le lien de la page à ses amis !



5.2.4 Page paramètres

La page de paramètres, comme son nom l'indique, permet aux utilisateurs de choisir certains paramètres comme la longueur du mot à deviner, le nombre d'essai maximum pour deviner un mot ou encore s'il souhaite utiliser le mode sombre. Ces paramètres sont initialisés de base à 5, 6 et ON.

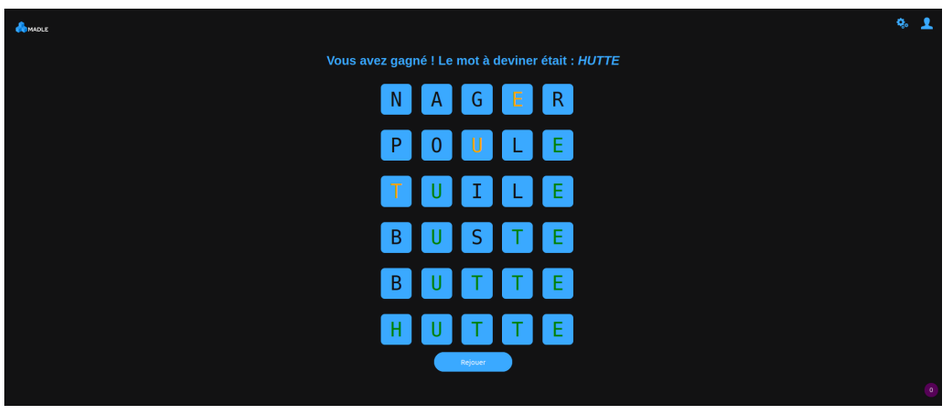


5.2.5 Page replay

Cette page permet à n'importe qui, connaissant l'identifiant d'une partie, de pouvoir voir le replay de cette dernière.Elle permet donc de revoir une partie déjà jouée de A à Z.

5.2.6 Page de Jeu

Cette page est la pierre angulaire de l'application web, c'est le jeu Wordle. Le joueur peut jouer une partie de Wordle en suivant les règles qui ont été énoncées dans la partie d'introduction. Le mot à deviner est choisi aléatoirement par l'application parmis tous les mots du dictionnaire.



5.2.7 Page erreur

Pour gérer les différentes erreurs dues aux entrées de l'utilisateur, nous avons mis en place une page qui permet à l'utilisateur de savoir quelle sorte d'erreur il a fait et donc de ne pas la refaire.

5.2.8 Design pages web

Nous avons rendu notre application web visuellement agréable en utilisant un fichier CSS standard.

5.3 Les difficultés rencontrées

Pour utiliser les différentes fonctionnalités de notre application web, nous avons dû sauvegarder nos parties dans la base de données. D'autres méthodes ont été envisagées mais n'ont jamais abouties. L'utilisation de "session" a nettement participé au code de cette application. Une légère méconnaissance du javascript à entrainée quelques complications, notamment dans l'implémentation du jeu Wordle.

5.4 Améliorations possibles

Ayant pour priorité de construire une application qui fonctionne, nous n'avons pas poussé le développement de l'esthétisme au maximum. Une amélioration serait donc à prévoir pour avoir un site encore plus attractif pour les utilisateurs afin qu'il leur soit naturel de naviguer entre les pages de notre application. L'utilisation de javascript de manière plus importante aurait ajouté de l'interactivité entre l'utilisateur et le site web.

Chapitre 6

Complexité et tests

6.1 Complexités

6.1.1 moyenne_nombre_coups(parties : list) -> float

On notera "parties" la liste des parties (réponse de la requête à la base de données) en paramètre de la fonction.
La fonction s'arrête lorsqu'une qu'elle a parcouru la liste.
On note n la taille de la liste parties. On trouve alors la complexité suivante : O(n)

6.1.2 moyenne_taille_mots(mots : list) -> float

On notera "mots" la liste des mots joués (réponse de la requête à la base de données) en paramètre de la fonction.
On note n la taille de la liste "mots".
On notera n la taille de la liste en paramètre de la fonction. On trouve alors la complexité suivante : O(n)

6.1.3 get_chart(moyenne : float) -> str

La fonction prend en argument la moyenne de victoire, notée moyenne.
La fonction génère simplement une chaine de caractères.
On trouve alors la complexité suivante : O(1)

6.1.4 choix(longueur : int) -> str

La fonction prend en argument un entier "longueur" qui correspond à la longueur du mot souhaité.
La fonction récupère une chaine de caractères au hasard dans une liste choisi grâce à l'argument "longueur".
On trouve alors la complexité suivante : O(1)

6.1.5 verif_dico(mot : str) -> bool

La fonction prend en argument une chaine de caractère qui correspond au mot proposé par le joueur.
La fonction parcourt la liste des mots de même taille que la proposition, s'arrête et renvoie True si elle trouve le mot, elle renvoie False si elle ne trouve pas le mot et qu'elle a parcourue toute la liste.
On trouve alors la complexité suivante : O(n)

6.1.6 occurrence(mot : str) ->tuple

La fonction occurence utilise deux boucles "for" qui, dans le pire des cas, font toutes les deux n tours de boucle ou n est la longueur du mot.
On trouve alors la complexité suivante : $O(n^2)$

6.1.7 correction(a_trouve : str, proposition : str) -> list

La fonction correction appelle diverses fonctions et des boucles for. Ainsi la complexité de la fonction est égale au maximum des complexités des diverses parties du programme. On retrouve notamment la fonction "occurence".
On trouve alors la complexité suivante : $O(n^2)$

6.1.8 get_words_json (filename : str) -> dict

La fonction prend en paramètre un chemin vers un fichier .txt qui comporte un mot par ligne.
La fonction parcourt le fichier texte
On note n le nombre de lignes du fichier .txt. On trouve alors la complexité suivante : O(n)

6.1.9 get_list(taille_mot : int, dico_json : dict) -> list

La fonction prend en argument la taille des mots que l'on souhaite jouer, ainsi qu'un dictionnnaire de mots.
La fonction récupère la liste des mots qui correspondent à une des valeurs du dictionnaire.
On trouve alors la complexité suivante : O(1)

6.1.10 void tableau_print(Tableau * one_tab)

La fonction prend en argument un tableau et affiche son contenu.
Pour cela la fonction effectue m tour de boucle où m est la taille du tableau et appelle à chaque ligne la fonction "list_print" qui est en O(n) avec n la taille de la ligne. Or chaque ligne peut être de taille différente donc nmax = max(n_i) pour tout i dans [1, m].
On trouve alors la complexité suivante : O(m*nmax)

6.1.11 void tableau_destroy(Tableau * one_tab)

La fonction prend en argument un tableau et libère la mémoire qui lui est reservée.
Pour cela la fonction effectue m tour de boucle où m est la taille du tableau et appelle à chaque ligne la fonction "list_destroy" qui est en O(n) où n est la taille de la ligne. Or chaque ligne peut être de taille différente donc nmax = max(n_i) pour tout i dans [1, m].
On trouve alors la complexité suivante : O(m*nmax)

6.1.12 char * new_proposition_word(List_Char ** lettres_possibles, List_String * dictionnaire, char * lettres_obligatoires, int one_taille)

La fonction prend en argument une liste de liste de char, un dictionnaire (qui est une liste chaînée), une liste de lettres obligatoires et un entier. Elle renvoie un string qui est le premier mot du dictionnaire correspondant aux critères imposés par la liste de liste de char. Le string renvoyé contient les lettres obligatoires qui sont comprises entre 0 et "one_taille".
Pour cela la fonction parcourt tout le dictionnaire et pour chaque mot vérifie si le mot correspond aux contraintes qui lui sont imposées. Cette fonction utilise la fonction "mot_possible" qui renvoie un booléen dépendant de différents critères. Cette fonction est en O(n) où n est la taille du mot. Ainsi si on note m la taille du dictionnaire, on fait m tour de boucle dans le pire des cas.
On trouve alors la complexité suivante : O(n*m)

6.1.13 int main()

La fonction main gère l'interface Homme machine et utilise les différentes structures de données.
La fonction utilise différentes fonctions dont new_proposition_word.
On trouve alors la complexité suivante : O(n*m)

6.2 Testing

Pour le testing, la méthode utilisée est celle du module pytest de python3 ainsi que la méthode *Right BICEP*. La méthode *Right BICEP* permet de tester les fonctions selon plusieurs critères. Tout d'abord le *Right* incite à se demander si le résultat est correct. Le B pour Boundary, incite à tester les conditions limites (effectuer des tests sur des listes vides qui ne doivent pas l'être). Le I de Inverse est la pour savoir si l'inverse de notre fonction est réalisable. Le C de Cross-Check, conseille de vérifier que deux fonctions différentes ayant le même but renvoient bien le même résultat. Le E de Error demande si le programme peut bugger, si les disques peuvent se remplir, etc.... Et le P de Performance, car il faut vérifier que les programmes soient performants ou, le cas échéant, s'il est possible de faire mieux.

Pour les structures de données la méthode utilisée n'est pas *Right BICEP* mais la vérification des axiomes qui est la combinaison entre les observateurs et les opérations internes.

Chapitre 7

Solveur

7.1 Introduction

La seconde majeure partie du Projet consiste en l'implémentation d'un solveur Wordle. Il était donc demandé de créer un algorithme, exclusivement en C, qui prendrait une serie de nombres en entrée (02110 par exemple) et qui donnerait un mot pris intelligemment dans un dictionnaire que nous avons à disposition. Pour ce faire nous avons fonctionné en plusieurs parties, premièrement nous nous sommes renseignés sur les solveurs Wordle existant, pour voir ce qui se faisait, comment, en combien de temps et si c'était optimisé et efficace. Nous avons ensuite réfléchi a quelle structure de donnée choisir et à comment nous allions implémenter le solveur pour qu'il soit efficace, après avoir choisi une structure de donnée, l'implémentation a pu commencer. Par manque de temps nous n'avons pas pu faire tout ce que nous avions en tête, le solveur peut encore être amélioré en utilisant différentes méthodes, notamment en utilisant d'une meilleure manière la théorie de l'information.

7.2 État de l'art

La première étape à donc été de se renseigner sur les solveurs déja existants, et il en existe plusieurs. Parmis les vidéos/articles que nous avons lu se trouve la vidéo de 3Blue1Brown mais aussi différents sites comme celui de dcode qui permet de "tricher" a Wordle grâce a leur solveur. Il existe donc plusieurs solveurs en C ou dans d'autres langages (Python, etc...) et un grand point commun de ces solveurs est qu'ils utilisent tous de la théorie de l'information, ainsi que des structures de données assez simple mais efficace (comme des tables de hachages par exemple).

7.3 Premières idées

Après s'être intéressés aux différents solveurs nous avons décidés de partir sur une structure de donnée de type "table de hachage". Néanmoins la recherche de bonnes fonctions de hachages nous a freiné dans cette idée. Il était difficile de trouver de bonne fonctions de hachages qui englobaient l'ensemble des mots et des cas à traiter, c'est pour cela que nous nous sommes redirigés vers une autre structure de donnée : les listes

7.4 Notre solveur

Ainsi comme dit précédemment nous avons décidé de nous orienter vers une autre structure que la table de hachage, le solveur utilise des listes de listes chainées. Ainsi le mot que le solveur va proposer est stocké dans une liste, de taille n (n étant la longueur du mot que l'on souhaite faire deviner), ou chaque élément est une liste comportant les lettres de l'alphabet, de A à Z. Afin de trouver le plus rapidement et le plus efficacement le mot recherché, chaque liste va se réduire en fonction de l'entrée utilisateur (00210 par exemple). Pour faire plus clair si le mot à deviner est "ANTRE" et que le solveur propose le mot "CRANE", l'entrée utilisateur va être "01112". Le premier chiffre, 0, signifie que la lettre n'est pas dans le mot, ainsi la lettre C va être enlevée de toutes les listes. Pour les trois 1 suivants, les lettres RAN vont êtres enlevées de leur listes respectives, mais seront obligatoirement dans le prochain mot proposé. Pour le dernier chiffre, le 2 signifie que la lettre E est a la bonne place, donc la dernière liste perd toute ses lettres a l'exception du E. Ainsi le prochain mot proposé contiendra un E à la dernière place, pas de C mais un R un A et un N a des places différentes que dans le mot CRANE.

7.5 Utilisation de la Théorie de l'information

La théorie de l'information, dite théorie de l'information de Shannon est une théorie probabiliste permettant de quantifier le contenu moyen en information d'un ensemble de messages. L'objectif de l'utilisation de la théorie de l'information dans notre solveur est de le rendre plus efficace et optimisé, notamment en "triant les mots" et en associant a chaque mot une polarité. Plus la polarité du mot est élevée plus le mot est commun (le mot "frontière" est plus couramment utilisé que le mot "Ayurvedas" par exemple). Ainsi, si les mots ayant une plus forte polarité arrivent avant ceux avec une plus faible polarité dans le dictionnaire, le solveur aura tendance a proposer des mots plus courants, favorisant ainsi les chances de victoire. La théorie de l'information est fréquemment utilisées en cryptographie ou en codage de l'information.

7.6 Structure de données

7.6.1 Structures implémentées

On dispose de listes chainées de caractères (char) et de chaines de caractères (char*).
Nous avons également implémenté une structure de tableau qui est une liste de pointeurs qui pointent tous vers une liste chainée.

7.6.2 Fonctionnement

Notre dictionnaire est donc un tableau où chaque ligne est une liste de mot de même taille. Plus précisément, notre tableau est une liste de pointeurs et chaque pointeur pointe vers une liste chainée de chaines de caractères (char*).
En plus de cela, nous avons implémenté des listes chainées de caractères (char). L'ensemble des lettres encore possibles au cours du jeu sont stockées dans une liste de liste chainée de lettres (char). Le fonctionnement est analogue aux regexp.

7.6.3 Avantages de cette structure

Le principal avantage d'utiliser les listes chainées est que nous ne sommes pas limités en nombre d'éléments. En effet, un changement de dictionnaire n'affecte pas le fonctionnement de notre solveur.
De plus, le parcours d'une liste est en O(n) ce qui reste raisonnable pour un n qui vaut au maximum 80000 (au cours du jeu, on est aux environ de 20000 car on utilise une liste de mot de même taille).
De plus, les listes de listes chainées de char permettent de retirer une lettre sans recopier la liste ce qui représente un gain de temps de mémoire.

7.7 Améliorations à venir

Il serait intéressant de prendre en compte de manière plus correcte les redondances de lettres dans un mot, ainsi que les paires de mot courante, ou les triplets. Ceci permettrait de rendre le solveur plus efficace, et ainsi de trouver des mots en moins de coups qu'actuellement. On pourrait également supprimer les mots du dictionnaire si on sait qu'on ne peut pas tomber dessus, ainsi on gagnerait du temps de "recherche" d'un nouveau mot.

Chapitre 8

Bilan

8.1 Bilan du projet par membre

8.1.1 Tanguy Boura

Points positifs	- Meilleure compréhension du langace C. - L'expérience d'un premier projet déjà réalisé ensemble à été utile. - Cohésion du groupe renforcée. - Meilleure compréhension de Git, notamment sur les branches.
Difficultés rencontrées	- Nouveau langage, le C. Langage peu évident. - Besoin d'une compréhension du langage pour gérer les différentes structures. - Le temps et les partiels de fin d'année ont posés des problèmes d'organisation.
Expérience personnelle	- Malgré les problèmes de temps, ce dernier fut mieux géré que lors du premier projet. - Grande progression en C et en Web. - Petit imprévu pendant la partie Web, j'ai du m'absenter quelque temps (aller retour à la frontière Pologne/Ukraine pour loger une famille d'Ukrainiennes).
Axes d'amélioration	- Apprentissage du C plus en avance pour pouvoir commencer le projet plus en avance. - Mieux appréhender la gestion du travail.

8.1.2 Thibault Boisseau

Points positifs	- Meilleur compréhension du langage C - Cohésion de groupe renforcée - Expérience du premier projet très utile
Difficultés rencontrées	- Nouveau langage abordé, le langage C - Nombreuses structures à utiliser
Expérience personnelle	- Meilleure gestion du temps due a l'expérience du premier projet - Amélioration sur des notions moins abordées dans le premier projet - Progression sur l'utilisation des bases de données
Axes d'amélioration	- Mieux appréhender la gestion du travail

8.1.3 Aurélien Troncy

Points positifs	- Meilleure compréhension de git au travers des branches et des autres outils pour la gestion de versions. - Méthodes de gestion de projet bien mieux maîtrisées - Sujet du projet très intéressant et la première partie qui reprenait le projet du premier semestre a permit de se rendre compte à quel point nous avons évolué.
Difficultés rencontrées	- Langage C assez compliqué à maîtriser - Superposition des partiels avec le rendu du projet - Coupure assez importante entre la fin de la première partie du projet et le début de la seconde qui est due à un méconnaissance du langage C qui ne permettait pas de commencer la seconde partie du projet.
Expérience personnelle	- Progression sur l'utilisation de structure de données avancée dans un contexte concret - Amélioration en tant que programmeur C - Première expérience en tant que coordinateur de projet assez plaisante
Axes d'amélioration	- Apprentissage du C plus en avance pour pouvoir commencer la seconde partie plus rapidement

8.1.4 Guillaume Bourgeon

Points positifs	- Meilleure compréhension du langage C et des structures de données - Nous avons amélioré notre façon de travailler en rapport avec les remarques faites à la fin du premier projet - Bonne cohésion de groupe et travail équitable
Difficultés rencontrées	- Démarrage du C en début de projet - Gestion du temps entre le projet et les partiels
Expérience personnelle	- Grande progression en C et en gestion de projet - Meilleure gestion du temps de développement
Axes d'amélioration	- Avoir une meilleure connaissance des langages utilisés en début de projet - Répartition des tâches encore plus précise

8.1.5 Bilan du projet par le groupe

Le second projet de notre groupe s'est mieux déroulé que le précédent notamment grâce à l'expérience que nous avons acquis. Le travail a été réalisé ensemble, en évitant au mieux les membres inactifs et en nous entraidant. Les réunions ont toutes été efficaces, ce qui a permis de bien nous organiser, et ainsi chacun a pu progresser dans d'autres domaines que ceux abordé dans le premier projet.
Nous avons cependant rencontré quelques difficultés. La partie solveur étant exclusivement codé en langage C, le manque d'expérience n'a pu être compensé que grâce aux nombreuses heures de travaux pratiques auxquelles nous avons participé. La période des partiels n'a de plus pas arrangé les choses puisqu'il fallait en plus des révisions avancer le projet.

Cependant le second projet a été mieux mené que le premier projet pour chacun d'entre nous.

8.2 Tableaux des temps

Domaines	Aurélien	Guillaume	Tanguy	Thibault
Compréhension du sujet	3h	3h	3h	2h
Base de données	3h	2h	2h	8h
WEB	22h	5h	5h	4h
Algorithmie	3h	11h	13h	20h
Solveur	20h	25h	19h	10h
GDP	2h	4h	5h	3h
Test	4h	2h	4h	10h
Réunion et Comptes-rendus	7h	7h	7h	7h
Rapport final	5h	12h	15h	7h
Total	69h	71h	73h	71h

Annexe A

Annexes

A.1 Comptes-rendus

A.1.1 PPII : Compte-rendu n°1

Rédacteur : Thibault Boisseau
19 mars 2022

Motif de réunion : Première réunion	Lieu : Distanciel
Participants : — TRONCY Aurélien — BOURGEON Guillaume — BOURA Tanguy — BOISSEAU Thibault	Date :19/03/2022 Heure de début : 9h00 Durée : 40 minutes

Ordre du jour :

1. Répartition des rôles.
2. Environnement de travail.
3. Gestion de projet.
4. Projet en lui-même.

Échanges :
— Désignation d'un coordinateur de projet pour une meilleure gestion.
— Répartition de la tâche de secrétaire entre les trois autres membres.
— Election unanime du coordinateur : Troncy Aurélien.
— Apprentissage des branches git pour régler le problème de push du précédent projet.
— Utilisation de Flask, Sqlite3, Pytest (le language C sera vu plus tard)
— Réalisation de la SWOT, WBS, RACI, GANTT (avoir plusieurs itérations de gantt).
— Penser à commenter le code au fur et mesure.
Déroulement :
— Présentation de Wordle.
— Implémentation du jeu en python.
— Implémentation de l'interface web.
— Implémentation des fonctionnalités optionnelles.
— Solveur en C (à voir plus tard)

Désignation d'un coordinateur de projet

Afin d'améliorer notre gestion du projet, nous avons décidé d'élire un coordinateur de projet. Contrairement au projet précédent, cela permettra un avancement plus rigoureux du projet.

Répartition de la tâche de secrétaire

Pour rédiger les comptes-rendus de façon efficace, nous nous sommes mis d'accord sur la personne qui doit rédiger le compte-rendu du jour. Les comptes-rendus suivants seront fait à tour de rôle par les membres du groupe.

Election unanime du coordinateur

À l'unanimité, c'est donc Aurélien Troncy qui a été désigné pour coordonner ce projet. Ses compétences ainsi que notre confiance envers lui ont motivé notre choix.

Apprentissage des branches git

Lors du précédent projet de nombreux problèmes de branche git sont survenus lors du dépôts du travail des membres. Une meilleure compréhension de ces branches et la création d'une branche commune sont envisagées.

Utilisation des langages

Nous avons revu ensemble les différents langages qui devront être utilisés lors de ce projet (python, C et HTML). Les différents frameworks ont également été cités, notamment Flask, Sqlite3 et Pytest. Le langage C étant nouveau pour tous les membres du groupe, une approche plus intense sur ce langage sera traitée lorsque les pointeurs auront été abordés en cours.

Réalisation des outils de gestion de projets

Nous avons revu également les différents outils de gestion de projets que nous devrons réaliser. Le GANTT devra être réalisé en plusieurs itérations contrairement au projet précédent.

Commentaire du code

Pour éviter toutes mauvaises surprises et que les différents membres du groupe se retrouvent face à une incompréhension devant le travail d'un autre, nous avons mis un point d'honneur a commenter le code à chaque fois.

Déroulement

Afin de préparer l'état de l'art, nous ferons une brève présentation de Wordle. Ce projet sera implémenté d'une part en python pour l'algorithme de jeu, en HTML nous implémenterons l'interface web permettant de jouer. Les fonctionnalités optionnelles du jeu sont envisagées mais seront abordées seulement si le temps le permet (comme par exemple la gestion de la difficulté), le solveur permettant de résoudre le jeu sera implémenté en C.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Jouer à Wordle	Tout le monde	prochaine réunion	aucun	Tout le monde
Prendre des notes	Tout le monde	prochaine réunion	Présentation rapide de wordle	Tout le monde

Prochaine réunion : Dimanche 27 Mars.

A.1.2 PPII : Compte-rendu n°2

Rédacteur : Guillaume Bourgeon
27 mars 2022

Motif de réunion : Réunion bilan sur Wordle	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 27 mars 2022 — Heure de début : 17h30 — Durée : 30 mins

Ordre du jour :

1. Mise en commun de nos recherches et parties sur Wordle
2. Conception du jeu en python
3. Réalisation du dictionnaire de mot
4. Conception de la base de données à partir d'une modèle E/A
5. Conception du site WEB
6. Bonnes pratiques à mettre en place

Échanges :
— Conception de l'application
— Bonnes pratiques à mettre en oeuvre
— Réalisation des documents de gestion de porjet.
— Faire le SWOT puis le WBS. Ensuite, faire la RACI et le Gantt.

Conception :
— On doit faire une conception (avant l'implémentation) du jeu en python.
— Conception python : découpage des fonctions dans différents fichiers (trouver le découpage et les noms de fonctions).
— Listes de listes de mots en fonction du nombre de lettres ou alors on plusieurs listes (exemple : motstaille1, motstaille2, . . .).
— Conception de la base de données (modèle E/A) : partie obligatoire et partie facultative (à faire ou à faire plus tard).
— Faire l'arborescence des pages WEB + utilisation flask et bootstrap.
— Documents de conception : seulement la partie python au début puis la partie C plus tard.

Bonnes pratiques :
— Mettre les préconditions dans les en-têtes des fonctions.
— On fait les tests sur chaque fonction (par une personne qui n'a pas développé la fonction) au moment où on les écrit, pas uniquement à la fin du projet.
— Rédiger le rapport au fur et à mesure (exemple : dès que la partie WEB est terminés, dès que le python est terminé, etc.).
— Faire des branches git pour chaque membre du groupe, et ne les push dans la branche principale qu'une fois testées.
— Faire des git push plus régulièrement

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Faire la SWOT	Guillaume	1 semaine	SWOT	Tout le monde
Conception base de donnée	Tout le monde	1 semaine	Document de conception	Tout le monde

Prochaine réunion :

Week-end du 3 avril 2022

A.1.3 PPII : Compte-rendu n°3

Rédacteur : Tanguy Boura
3 mai 2022

Motif de réunion : réunion d'avancement	Lieu : Distanciel
Participants : — TRONCY Aurélien — BOURGEON Guillaume — BOURA Tanguy — BOISSEAU Thibault	Date : 03/04/2022 Heure de début : 17h40 Durée : 45 minutes

Ordre du jour :

1. Avancement de la SWOT.
2. Avancement de la conception de la Base de Données.
3. Avancement de la conception de la page Web.

Échanges :
— La matrice Swot à été réalisée.
— Regard sur la partie conception de la partie Web
— Regard sur la partie conception de la Base de Données
— Les tâches à faire après avoir validé les documents de conceptions.
Déroulement :
— Regard sur la Matrice SWOT et commentaires.
— Discussions autour de la future page Web.
— Discussions autour de la future Base de Données.

Observation et discussion sur la matrice SWOT

Guillaume Bourgeon à pu réaliser une Matrice SWOT depuis la dernière réunion, il l'a mise sur le Drive du projet afin que tout le monde puisse la regarder. Une menace a été rajoutée, ainsi qu'une faiblesse. Tout le monde valide la Matrice SWOT telle qu'elle est.

Regard sur la partie conception de la partie Web

La partie Web a été imaginée, elle sera composée de 6 pages différentes : une page d'accueil, une de connexion, la page de jeu, un accès au profil, le choix des paramètre et une page d'erreur qui variera selon l'erreur.

— Accueil : page servant à aller sur les pages de connexion ou le profil, ou bien à lancer le jeu
— Connexion : Page permettant au joueur de s'identifier
— Le Jeu : C'est la page de Jeu, sur laquelle il faut trouver un mot
— Profil : Page avec l'ensemble des informations sur le joueur, pseudo, statistiques, historique des parties, ...
— Paramètres : Page permettant de modifier le nombre de lettre du mot à trouver, le nombre de tentative, activer ou désactiver un darkmode. Cette page dépend d'ou l'on vient, certains paramètres peuvent être accessibles ou non
— Erreur : C'est une page d'erreur, expliquant ce qui à posé problème

L'idée de créer un mode time attack a été mentionnée mais pas retenue.

Regard sur la partie conception de la Base de Données

La partie Base de Données et sa conception ont été discutées. L'idée de rajouter des statistiques mentionnées précédemment a été ajoutée dans le document de conception de la base de données. Des discussions autour de la table dans laquelle les statistiques seront ont lieu. Les statistiques seront dans une table à part. Il faudra en faire un schéma Entité/Association pour que ce soit plus clair et plus compréhensible par tout le monde.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Faire un schéma de l'application web	Aurélien	prochaine réu	Fichier PDF	Tout le monde
Faire le schèma E/A	Tanguy	prochaine réu	Feuille	Tout le monde
Faire le schème E/A sur PlantUML	Guillaume	prochaine réu	Fichier PDF	Tout le monde
Continuer la conception de l'algorithme python	Thibault	prochaine réu	Fichier PDF	Tout le monde

Prochaine réunion : Samedi 09 Avril.

A.1.4 PPII : Compte-rendu n°4

Rédacteur : Thibault BOISSEAU
9 avril 2022

Motif de réunion : Retour sur l'envoie du mail de conception et attribtuion des rôles	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 09/04/2022 — Heure de début : 16h00 — Durée : 35 min

Ordre du jour :

1. Avancement
2. Retour sur la conception
3. Implémentation a faire pendant les vacances
4. WBS, GANTT, RACI

Échanges :
— Les schéma bd et web ont été soumis au référent du projet M. Festor.
— Retour sur la conception et le mail envoyé par le responsable de projet.
— Remarques sur la BD.
— Réalisation du jeu, du site et de la bd.
— L'esthétique du jeu.
— Répartition des outils de gestion de projet.
Déroulement :
— Bilan sur l'avancement du projet.
— Discussion sur le retour de mail de M.Festor.
— Attribution des rôles de chacun (implémentation et outils de gestion de projets).

Avancement :

Le document de conception a été finalisé par les membres de l'équipe, Aurélien ayant réalisé le schéma de l'application web, Tanguy et Guillaume ont respectivement conçu et réalisé le schéma E/A grâce à PlantUML et Thibault a continué la conception des différentes fonctions python implémentant le jeu. Une fois la conception finie, nous l'avons soumis à M. Festor pour qu'il puisse la valider ou non.

Retour sur la conception et sur le mail du responsable de projet :

Après son retour de mail, le responsable de projet nous a signalé de grosses incohérences dans le schémas E/A de notre base de données et qu'il faudrait régler cela pour le bon déroulement de l'application. N'ayant rien dit sur la partie Web du document de conception, nous la considérons valider. Nous lui avons renvoyé un nouveau schéma E/A pour qu'il puisse le valider si le temps le lui permet. En attendant nous commencerons l'implémentation pour le bien du projet.

Remarques sur la BD :

Suite aux remarques, une vérification rigoureuse sera effectuée sur la BD pour éviter que les prédictions du responsable de projet ne s'avèrent justifiées et ne nous posent problème lors de la conception.

Réalisation du jeu, du site et de la bd :

Le jeu doit être fini et fonctionnel avant la rentrée, ainsi que les tests associés aux fonctions implémentant le jeu. Pour le site, l'ensemble des pages devront avoir été créés et être fonctionnelles tout en correspondant aux attentes et à l'esthétique. La charte graphique devra également être définie. La BD sera traitée plus tard à cause de l'attente d'une éventuelle validation. Cette dernière sera réévaluée lors de la prochaine réunion.

l'esthétique du jeu :

Les pages de l'application web concernant le jeu seront, après vote, en noir et bleu (couleur majoritaire)

Répartition des outils de gestion de projet :

Les membres du groupe se sont mis d'accord pour réaliser les outils de gestion de projet. Tanguy réalisera le WBS, Aurélien le Gantt et Thibault réalisera le RACI lors de la prochaine réunion en accord avec les autres membres du groupe.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Faire le WBS	Tanguy	1 semaine	PDF	Tout le monde
Faire la matrice RACI	Thibault	faire lors de la prochaine reu	PDF	Tout le monde
Création des différente page "fonctionnel"	tout le monde	1 semaine	application	Tout le monde
Implémentation du jeu en Python	Guillaume + Thibault	1 semaine	jeu fonctionnel	tout le monde
Test des fonctions	Tanguy + Aurélien	1 semaine	fichier test	tout le monde

Prochaine réunion :

Samedi 16 avril 2022, 15h

A.1.5 PPII : Compte-rendu n°5

Rédacteur : Guillaume Bourgeon
14 avril 2022

Motif de réunion : Avancement du développement de l'application web	Lieu : Visioconférence Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 14/04/2022 — Heure de début : 18h — Durée : 30 mins

Ordre du jour :

1. Avancement
2. Base de données
3. Site WEB
4. WBS et RACI
5. Suite de l'implémentation

Échanges :
— Validation du WBS
— Avancement du projet
— Création de la RACI
— Mise en place d'une redirection vers la définition du mot à trouver
— Le Gantt est fait, il sera modifié au fur et à mesure
— On doit implémenter la base de données (qui est déjà conçue)
— Affichage du résultat de la partie sur la page de jeu
— Il faudra jouer au jeu et trouver des bugs

Avancement

Le WBS a été fait. Les pages accueil, jeu, paramètres, connexion et profil ont été réalisées et leur esthétique a été fait en CSS uniquement. Le backend de connexion et acceuil ont également été implémentés. Pour le jeu, les fonction python ont été réalisées et testées.

Gestion de projet

Après la validation du WBS, nous avons déterminé les rôles de chacun sur toutes les tâches. Nous avons fait cela pour trois des quatre étapes principales (Base de connées, application web et algorithmes de jeu Wordle). Le Gantt a aussi été crée et sera mis à jour au fur et à mesure du projet.

Développement

Après discussion, nous avons trouvé une fonctionnalité supplémentaire à implémenter. Nous allons créer une redirection vers la page de définition du dictionnaire du mot à trouver en fin de partie. Il nous reste alors à faire l'implémentation de la base de données. Nous allons tous devoir jouer au jeu pour déceler d'éventuels bugs et les règler le cas échéant.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Faire la base de données	Thibault	1 semaine	Base de données	Tout le monde
Faire le backend	Guillaume	1 semaine	Application	Tout le monde
Jouer au jeu	Tout le monde	1 semaine	Bugs	Tout le monde
S'informer sur les solveurs	2 semaine	Aucun	Tout le monde	

Prochaine réunion :

Week-end du 23

A.1.6 PPII : Compte-rendu n°6

Rédacteur : Tanguy Boura
23 avril 2022

Motif de réunion : fin du développement de l'application web et avancement	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 23/04/2022 — Heure de début : 18h — Durée : 30 mins

Ordre du jour :

1. Avancement
2. Reste à faire

Échanges :
— Avancement du projet
— Avancement du Backend
— Il faudra jouer au jeu et trouver des bugs
— les améliorations à venir

Avancement

Toute la partie Back-end a été faite par Guillaume, aucun problème de ce côté tout marche bien.

Informations sur les solveurs

Tous les membres du projet se sont un peu renseignés sur comment faire un solveur, via des vidéos, des site webs, ou en réfléchissant avec un papier et un crayon. La vidéo de 3Blue1Brown a été visionnée par tout le monde, et relativement appréciée. La tâche doit être finie dans une semaine.

Trouver des bugs

Tous les membres du projet jouent au jeu de temps en temps sur leurs ordinateurs, on recherche des bugs en faisant des manipulations "inhabituelles". Pour l'instant certains bugs ont été trouvés, et corrigés dans la foulée. La chasse au bug continue.

Améliorations à venir

Système de points Un système de point pourrait être intégré dans l'application. En gagnant une partie, un joueur remporterait un certain nombre de point selon le nombre de lettres du mots, et le nombre de tentatives. Ces points serviraient à acheter des cosmétiques dans le jeu (comme des thèmes, ou des avatars), uniquement du cosmétiques, et pas d'aide dans le jeu.

Statistiques Une page de statistique sera implémentée dans le jeu, donnant le nombre de victoire, de défaites, le nombre de partie jouées ainsi que le pourcentage de victoire.

succés Des succés seront également implantés, se déverrouillant lorsque le joueur accomplis certains haut-faits, tels que gagner une partie en deux coups, gagner sa première partie ou d'autre encore (selon le temps que nous avons)

replays de partie Dans la page "mon profil", la possibilité de voir le replays de nos partie sera possible, en cliquant sur la partie nous allons voir tout les mots tentés, dans quel ordre ainsi que le mot qui était a deviner.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Faire les statistiques	Guillaume	1 semaine	Code python	Tout le monde
Faire la déconnexion	Tanguy	1 semaine	Code python	Tout le monde
Faire les succés	Thibault	1 semaine	Code python	Tout le monde
Faire les replays de partie	Aurélien	1 semaine	Code python	Tout le monde
Chercher et corriger des bugs	Tout le monde	1 semaine	Code python	Tout le monde
Faire une boutique	Tout le monde	indéfini	Code python + PDF	Tout le monde

Prochaine réunion :

Week-end du 23

A.1.7 PPII : Compte-rendu n°7

Rédacteur : Thibault BOISSEAU
28 avril 2022

Motif de réunion : Réunion d'avancement avant la présentation du jeu devant le jury	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 28/04/2022 — Heure de début : 19h00 — Durée : 45 min

Ordre du jour :

1. Avancement
2. dernière ligne droite

Échanges :

— La base de données est finie
— Back end réalisé et fonctionnel
— Réparation des bugs
— Solveur (encore une semaine)
— Création d'un compte pour personne non connectée

Déroulement :

— Débrief sur ce qui a été accompli
— Enumération du reste du travail et des différents bugs à régler avant la présentation
— discussion sur le solveur à réaliser prochainement

Avancement :

La page statistique a été réalisé par Guillaume, qui en a profité pour ajouter des diagrammes pour observer la performance des joueurs soit, leur pourcentage de victoire ainsi que différents éléments utiles comme. Tanguy s'est occupé de la déconnexion du joueur que l'on peut changer désormais sans fermer la page. Les replays de parties sont fonctionnels, Aurélien a fait en sorte qu'ils soient accessible en cliquant dessus. Thibault s'est occupé de la partie "succès" du joueur en intégrant dans la base de données quatres succès dans un premier temps et dont les joueur remportent selon leur parties. Les membres de l'équipe ont repéré les différents bug de l'application afin de les corriger avant la présentation. La possibilité d'intégrer une boutique n'a pas été retenu faute de temps.

Réparation des bugs :

Pour éviter toutes mauvaises surprises lors de la présentation, les membres de l'équipe ont joué de nombreuses fois au jeu suivant plusieurs paramètres ainsi que parcouru le code, afin de repérer toutes les anomalies, ainsi les bugs seront tous réglé avant la présentation.

Solveur :

Des premières recherches on été faites en ce qui concerne la réalisation du solveur en c, plusieurs pistes on été partagés, d'autres recherches vont tout de même être menées pour que tous les membres s'accorde sur une méthode.

création d'un compte pour personne non connectée :

Le sujet demandant de laisser un joueur non connecté pouvoir joué librement, un compte spécial est alors créé et utilisé pour les personnes jouant sans être préalablement connecté.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Test fonction auxiliaire	Tanguy	1 jour		Tout le monde
Création utiisateur dans la bd + impossibilité de se connecté avec lui + sauvegarde des parties joué par une personne non connecté par cette utilisateur	Aurélien	1 jour		Tout le monde
commentaire python (app.py, auxiliaire.py)	Thibault + Guillaume	1 jour		Tout le monde
rédiger le readme	Tout le monde	1 jour		Tout le monde
lire le code	Tout le monde	1 jour		Tout le monde
modification couleur historique	Tout le monde	1 jour		Tout le monde

Prochaine réunion :

a definir mais rapidement

A.1.8 PPII : Compte-rendu n°8

Rédacteur : Guillaume BOURGEON
8 mai 2022

Motif de réunion : Conclusion de la partie 1 et début de la partie 2 du projet	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 08/05/2022 — Heure de début : 16h40 — Durée : 30 min

Ordre du jour :

1. Avancement
2. Début de la partie C pour le solver

Échanges :
— Avancement
— Structure de données : arbre (à envisager), liste et tas (probablement pas idéal), table de hachage (à envisager)
— Relecture de l'énoncé pour la partie C
— Quelques fonctions peuvent être faites avant le solver (lecture dans le fichier wsolf.txt, interaction en lignes de commandes, proposition des mots, retour de l'utilisateur, . . .)
Déroulement :
—

Avancement :

La partie WEB est terminée. La soutenance s'est bien passée.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Se renseigner sur les solvers	Tout le monde	7 jour	Rien	Tout le monde
Rapport (partie WEB)	Tout le monde	7 jour	Rapport	Tout le monde
commentaire python (app.py, auxiliaire.py)	Thibault + Guillaume	1 jour		Tout le monde
Réfléchir à la structure de données	Tout le monde	7 jour	Rien	Tout le monde
Faire les fonctions en lignes de commandes et de lecture de fichier texte	Tout le monde	7 jour	Fichier C	Tout le monde

Prochaine réunion :

Week-end du 14 mai

A.1.9 PPII : Compte-rendu n°9

Rédacteur : Tanguy Boura
8 mai 2022

Motif de réunion : Avancement	
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 15/05/2022 — Heure de début : 18h40 — Durée : 30 min

Ordre du jour :

1. Avancement

Avant propos : les partiels nous ont ralentis dans le projet, mais on à pu travailler quand meme
Échanges :
— Avancement
— Se renseigner sur les solvers
— Réfléchir à une structure de données : arbre (à envisager), liste et tas (probablement pas idéal), table de hachage (à envisager)
— Quelques fonctions peuvent être faites avant le solver (lecture dans le fichier wsolf.txt, interaction en lignes de commandes, proposition des mots, retour de l'utilisateur, . . .)
Déroulement :
—

Avancement :

Les commentaires pour la partie python ont été réalisés dans l'ensemble, et tout le monde s'est renseigné sur les solveurs, ainsi que sur les différentes structures de données pouvant être utilisées pour réaliser le solveur.

Renseignement sur les solveurs

Une des choses que tout le monde a retrouvé dans l'ensemble des solveurs Wordle existant est qu'il faut des connaissances en Théorie de l'information. Pour voir les lettres ayant la plus grande occurence dans les mots français, les mots les plus courants etc...

structure de donnée

Aurélien s'est beaucoup renseigné sur quel genre de structure de données peuvent être utilisées. Les listes ne semblent pas être le meilleur choix. Pour ce qui est des automates on nous a dit que ça ne serait pas la manière la plus optimisée mais c'est une structure qui fera beaucoup progressé en C. Tandis que les tables de hachages semblent être la meilleure des solutions, tant d'un point de vue optimisation que entrainement en C. La difficulté sera de trouver de bonnes fonctions de hachages. Il faudra également prouver que les fonctions atteignent tout les mots possible et existants

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Rapport	Tout le monde	jusqu'à la fin	Rapport	Tout le monde
réfléchir a des fonctions de hachages	Tout le monde	1 semaine		Tout le monde
Faire les fonctions en lignes de commandes et de lecture de fichier texte	Tout le monde	7 jour	Fichier C	Tout le monde

Prochaine réunion :

Week-end du 21 mai

A.1.10 PPII : Compte-rendu n°10

Rédacteur : Thibault BOISSEAU
21 mai 2022

Motif de réunion : Avancement et mise au point	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 21/05/2022 — Heure de début : 18h00 — Durée : 40 min

Ordre du jour :

1. avancement
2. reste à faire / sprint final

Échanges :
— liste chaînée = dictionnaire
— regexp »> all
— liste de liste contenant toutes les lettres de l'alphabet puis on enleve celle qui ne conviennent pas au fur et à mesure
— La structure sera detaillée dans le rapport
— Matrice RACI
— écriture des fonctions :
-recupérer le dictionnaire -> liste de taille fixe triée
-structure de liste chaînée implementer (rajouter supprimer")
-faire une liste etant donne un dico renvoie le premier suivant l'ensemble -des critere donne
Déroulement :
— Réflexion sur les strucutures.
— Raci compléter.
— Récapitulatif du travail a faire

Avancement :

Tous le monde a participé à l'écriture du rapport de projet, il faut donc continuer ainsi. Les fonctions en ligne de commade on été faites. La table de hashage n'a été retenu que pour créer la structure dictionnaire qui comprends donc les mots de notre dictionnaire.

Réflexion sur les structures.

La table de hashage a donc été gardé uniquement pour réaliser le dictionnaire. Pour réaliser le solveur, notre choix c'est porté sur une liste de liste contenant toutes les lettres de l'aphabet, chaque sous liste représente une lettre du mot à trouver, on éliminera les lettres inutiles et on cherchera donc les mots possibles comme du regexp. D'autres modification seront apporté si nous manquons de temps ou si notre idée n'est pas réalisable.

Raci compléter.

La partie de la matrice RACI sur l'algorithmie du solveur a enfin été écrite.

Récapitulatif du travail à faire.

tous les membres de l'équipe projet ont revu ensemble le travail restant. Le rapport et les outils de gdp seront donc complétés et les différents algorithme et structure de données sont répartis entre les membres.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Rapport	tout le monde	jusqu'a la fin	Rapport	tout le monde
écriture des algo	tous le monde	prochaine reu	solveur en c	tout le monde

Prochaine réunion :

Vendredi 27 Mai

A.1.11 PPII : Compte-rendu n°11

Rédacteur : Guillaume BOURGEON
8 mai 2022

Motif de réunion : Rush final	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 27/05/2022 — Heure de début : 18h45 — Durée : 30 min

Ordre du jour :

1. Avancement
2. Rush final

Échanges :
— Avancement
— Reste à faire : algo pour ouvrir un fichier wsolf.txt et qui récupère la taille des mots que le solver doit trouver.
— Algorithme de Tanguy : finir la structure et déboger
— Algorithme : parcourir les mots du dictionnaire et vérifier si un mot est possible avec la liste des listes des lettres.
— Faire l'algo final qui assemble toutes les fonctions.
— Partie C et SD du rapport
— Faire le Gantt

Avancement :

Rapport : on a bien avancé, Thibault s'y est attelé en priorité. Algorithme pour récupérer le dictionnaire : fait et fonctionnel

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Rapport	Tout le monde	Fin	Rapport	Tout le monde
Ouvrir fichier wsolf.txt	Guillaume	Fin	fichier .c	Tout le monde
Programme de Tanguy : structure et debog	Tanguy et Aurélien	Fin	fichier .c	Tout le monde
Algorithme de parcours de dico et vérifie si un mot est possible	Guillaume	Fin	fichier .c	Tout le monde
Faire l'algo final qui assemble toutes les fonctions	Tout le monde	Fin	Fichier C	Tout le monde
Gantt	Aurélien	Fin	Gantt	Tout le monde
Test des fonctions	Thibault	Fin		Tout le monde

Prochaine réunion :

Lundi 30 mai

A.1.12 PPII : Compte-rendu n°12

Rédacteur : Thibault BOISSEAU
21 mai 2022

Motif de réunion :Avancement et mise au point	Lieu : Discord
Participants : — Tanguy BOURA — Thibault BOISSEAU — Guillaume BOURGEON — Aurélien TRONCY	— Date : 05/06/2022 — Heure de début : 10h00 — Durée : 30 min

Ordre du jour :

1. avancement
2. reste a faire / sprint final

Échanges :
— revue des choses qui ont été faites
— revue du reste a faire
Déroulement :
— bilan sur le travail réaliser (avancement)
— bilan sur le travail restant

Avancement :

Depuis la dernière réunion, le rapport a bin avancé ne reste plus qu'a rédiger la partie concernant le solveur, Guillaume a réaliser les fonctions relatant du fichier wsolf contenant un entier correspondant a la longueur de mot. Les algorithme permattant de parcourir le dictionnaire on également été réalisé ce qui est essentielle pour la réalisation de l'algorithme final concentrant tout ceux réalisé jusqu'à lors. Aurélien a terminé la réalisation du gantt. Tanguy a réalisé le programme de débugage. Thibault a bien avancé les tests mais doit ajouter pour les structures de données des tests sur les axiomes.

Todo list :

Description	Responsable	Délai	Livrable	Validé par
Rapport	tout le monde	jusqu'à la fin	Rapport	tout le monde
Algo final	tout le monde	jusqu'à la fin	solveur en c	tout le monde
test fonction	thibault	jusqu'à la fin		tout le monde
Développement algo améliorer	tout le monde	jusqu'à la fin	solveur	tout le monde

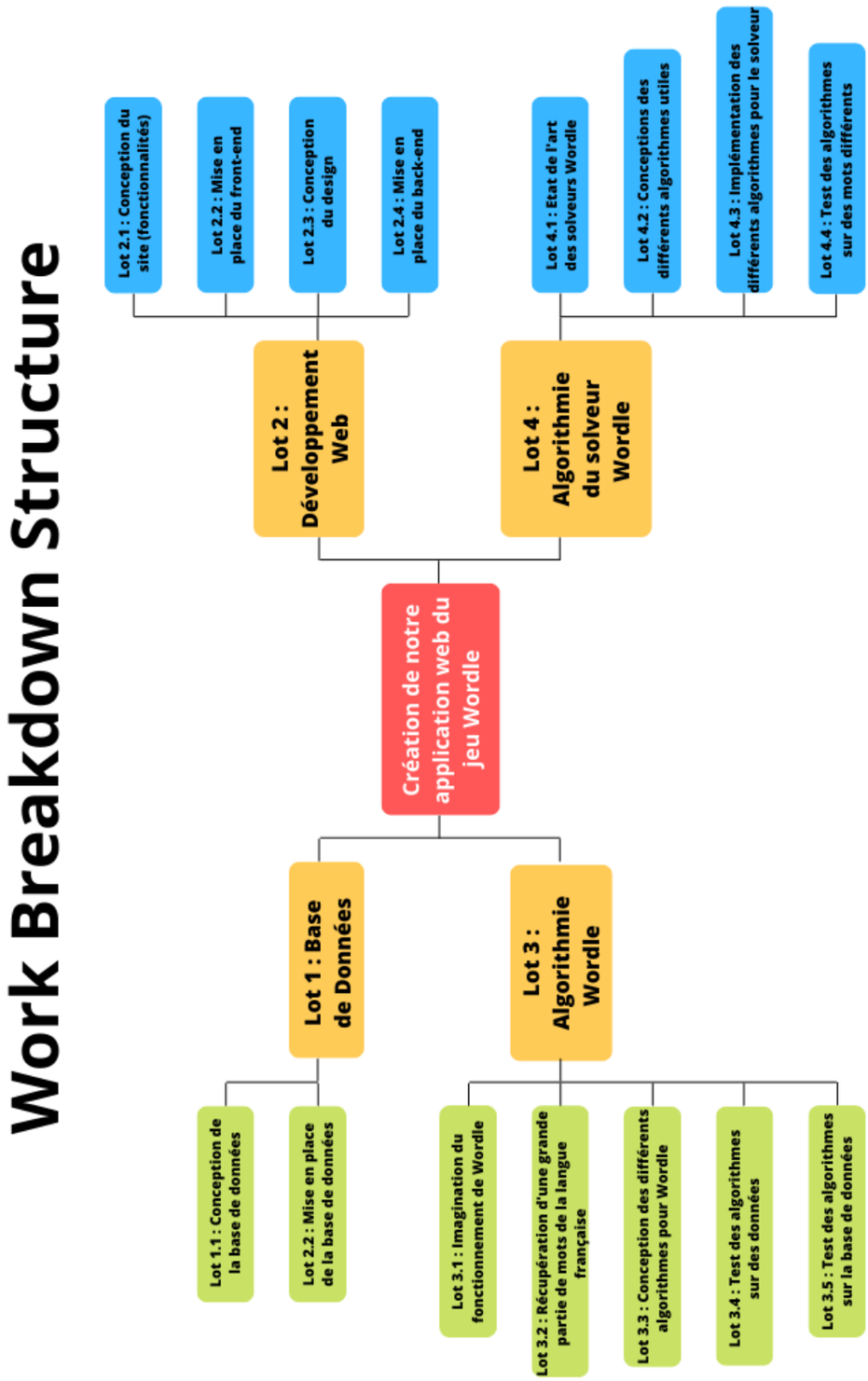
Prochaine réunion :

8 juin 2022 pour relecture du rapport

A.2 SWOT

Forces : - Les membres du groupe ont pris l'habitude de travailler ensemble sur le projet du premier semestre. - Bonne maitrise de python, Flask, Bootstrap. -	Faiblesses : - Découverte du C. - Découverte des structures de données.
Opportunités : - Existence de beaucoup de solveurs de Wordle donc de la documentation sera disponible.	Menaces : - Partiels de fin de semestre : la gestion du temps pourra être problématique.

A.3 WBS

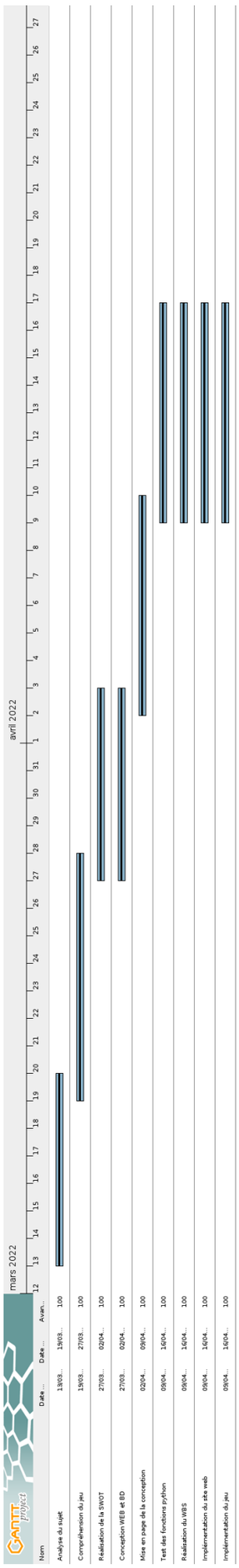


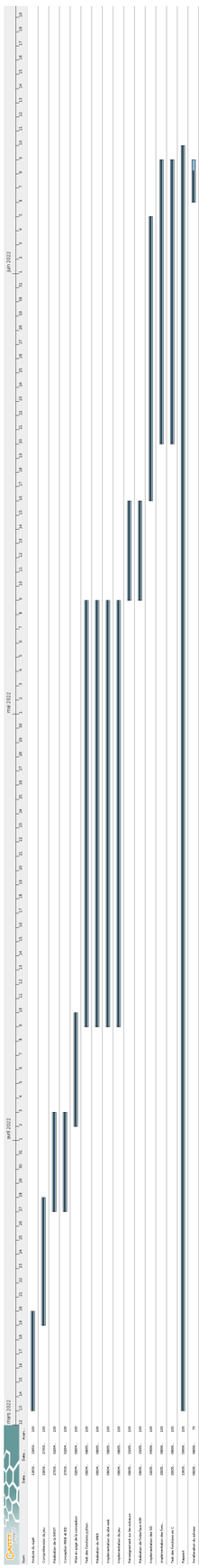
A.4 RACI

	Livrables ou Tâches	Troncy Aurélien	Boura Tanguy	Boisseau Thibault	Bourgeon Guillaume	M.Festor
		Equipe projet				Equipe pédagogique
	Lot 1 : Base de Données					
	Tâche 1.1	R/C	R/A	R/C	R	I
	Tâche 1.2	R	R	R/A	R	
	Lot 2 : Développement Web					
	Tâche 2.1	R	R/A	R	R	I
	Tâche 2.2	R/A	I	I	I	
	Tâche 2.3	R/A	I	I	I	
	Tâche 2.4	R	R	R	R/A	
	Lot 3 : Algorithmie Wordle					
	Tâche 3.1	R	R	R	R/A	
	Tâche 3.2	I	R	I	R/A	
	Tâche 3.3	I	I	R/A	R	
	Tâche 3.4	R/A	R/C	I	I	
	Tâche 3.5	R	R	R/A	R	
	Phase 4 Algorithmie du Solveur Wordle					
	Tâche 4.1	R	R/A	R	R	
	Tâche 4.2	R/A	R	R	R	
	Tâche 4.3	R	R	R	R/A	
	Tâche 4.4	R	R	R/A	R	

R Réalisé
A Autorité
C Conseils
I Informé

A.5 Gantt


GANTT project
mars 2022
avril 2022
Nom
Date ...
Date ...
Avan...
Analyse du sujet
13/03...
13/03...
100
Compréhension du jeu
13/03...
27/03...
100
Réalisation de la SWOT
27/03...
02/04...
100
Conception WEB et BD
27/03...
02/04...
100
Mise en page de la conception
02/04...
09/04...
100
Test des fonctions python
09/04...
16/04...
100
Réalisation du WBS
09/04...
16/04...
100
Implémentation du site web
09/04...
16/04...
100
Implémentation du jeu
09/04...
16/04...
100


GANTT
mars 2022
avril 2022
mai 2022
juin 2022

Bibliographie

[1] Custom checkboxes. https ://www.creativejuiz.fr/trytotry/custom-checkbox-full-css3-flat-ui.html.

[2] Dictionnaire. https ://www.freelang.com/dictionnaire/dic-francais.php.

[3] Dictionnaire. http ://step.ipgp.fr/index.php/Fichier :Liste$_m ots.txt$.

[4] Dictionnaire petit larousse. http ://www.3zsoftware.com/fr/listes.php.

[5] Plantuml. http ://www.plantuml.com/.

[6] Raci. https ://www.vertex42.com/ExcelTemplates/raci-matrix.html.

[7] Wbs. https ://www.canva.com/fr$_f r$/.

[8] Wordle, 2022. https ://www.nytimes.com/games/wordle/index.html.

[9] 3Blue1Brown. Solving wordle using information theory, 2022.

[10] Marc Baudoin. Apprends latex !, 2012.

[11] Tanguy Boura ; Aurélien Troncy ; Thibault Boisseau ; Guillaume Bourgeon. *Rapport Projet Pluridisciplinaire d'Informatique Intégrative*. 2021.

[12] Codecademy. A wordle type game using javascript. https ://discuss.codecademy.com/t/a-wordle-type-game-using-javascript/646680.

[13] Grafikart.fr. Comprendre git (7/18) : Les branches, 2015.

[14] Wikipédia. Wordle, 2022. https ://fr.wikipedia.org/wiki/Wordle.