

Le Mans Université
Licence Informatique *2ème année*
Module Conduite de projet
Rapport Final
Projet "Mish Mash"

Asmae Bouhandi
Guillaume Derrien
Quentin Bellanger
Guillaume Briffault

18 mai 2020

Table des matières

1	Introduction	3
2	Gestion de projet	4
2.1	Cahier des charges	4
2.2	Planning prévisionnel	4
2.3	Répartition des tâches	5
2.4	Outils utilisés	6
2.4.1	Outils de communication :	6
2.4.2	Outils de développement :	6
2.4.3	Outils de gestion de développement :	6
2.4.4	Outils de documentation :	7
2.4.5	Outils de gestion de projet :	7
2.4.6	Outils de de partage de fichiers :	7
3	Analyse et Conception	8
3.1	Choix de conception	8
3.1.1	Nom du jeu :	8
3.1.2	Particularités du jeu :	8
3.2	Fonctionnalités attendues	9
3.3	Règles de notre jeu	11
3.4	Comment jouer ?	11
4	Développement	14
4.1	Structure de données	15
4.2	Fichiers	16
4.3	Bibliothèques utilisées	18
5	Résultats	18
5.1	Résultat	18
5.2	Fonctionnalités implémentées	18
5.3	Respect du planning prévisionnel	19
5.4	Améliorations possibles	20
5.5	Bilan critique	20
5.5.1	Bilan individuel	20
5.5.2	Bilan collectif	21
6	Annexes :	22
6.1	Annexe n1 : Jeux de tests	22

Remerciements

Nous tenons à remercier toute l'équipe pédagogique qui nous a encadrés tout au long de ce projet. Nous remercions plus particulièrement Youness LAGHOUAOUTA qui nous a conseillé et guidés. Nous remercions aussi Madame Claudine PIAU-TOFFOLON ainsi que Madame PY Dominique pour toutes les conseils et informations qu'elles nous ont fournies.

1 Introduction

Dans le cadre du module “Conduite de Projet”, en deuxième année de licence Informatique à Le Mans Université dirigé par Madame Claudine Piau-Toffolon, nous devons réaliser un jeu que nous avons décidé de nommer “Mish Mash”.

Parmi les sujets proposés, notre groupe a choisi de réaliser un jeu de mots-mêlés.

Le principe de ce jeu est très simple. Il consiste à repérer dans une grille remplie de lettres, des mots connus du dictionnaire, lus dans les deux sens, horizontalement, verticalement et en diagonale. Et cela en s'aidant de la liste des mots présents dans la grille. Le but du joueur est donc de trouver tous les mots présents dans la liste.

Nous devons donc assurer la gestion d'un projet par équipe de quatre. Notre équipe est constituée de Asmae BOUHANDI, Guillaume BRIFFAULT, Guillaume DERRIEN et Quentin BELLANGER.

Des contraintes ont été imposées pour ce projet, en effet nous devons impérativement assurer la sauvegarde d'une partie mais également le chargement de celle-ci à partir d'un fichier. Nous devons également mettre en place les outils de développement vu au cours du semestre comme par exemple la gestion d'un dépôt Github ou encore la sauvegarde de la documentation à l'aide de Doxygen. Ce document a pour but de décrire le déroulement de notre projet de jeu “Mish Mash”. Ce rapport contient l'ensemble des éléments du projet, autant d'un point de vue technique que de gestion de projet. Dans une première partie, nous présentons la gestion du projet ainsi que les outils utilisés, puis nous aborderons l'analyse et la conception du jeu, ensuite nous décrirons le développement et enfin nous finirons par les résultats ainsi qu'un bilan critique ou nous présenterons entre autre le diagramme de Gantt qui a été effectivement réalisé.

2 Gestion de projet

L'équipe de projet est composée de membres provenant de différents cursus. Pour la majorité des membres du groupe, ce jeu est leur première expérience dans le domaine de conduite de projet. Suite à la présentation des sujets par Mr Antoine Laurent, plusieurs réunions ont été mises en place afin de déterminer les compétences et atouts des différents membres du groupe et ainsi choisir un sujet en conséquent. Les réunions ont aussi permis de répartir au mieux les différentes tâches entre les membres de l'équipe.

Pour accompagner le développement du projet des les premiers stades, des cours gestion de projet, de communications et de gestion d'outils de développement ont été mis en place. Ces cours ont introduit certains outils ainsi que les méthodologies nécessaire au bon déroulement d'un projet.

2.1 Cahier des charges

Pour commencer, la mise en place d'un cahier des charges à été la première tâche effectuée. En effet le cahier des charges permet de décrire les conditions attachées à l'exécution du projet, il nous a permis dans un premier temps, de définir le contexte et les objectifs techniques. En organisant nos idées, nous avons ainsi pu vérifier la concordance et la faisabilité de notre projet.

2.2 Planning prévisionnel

Après avoir réalisé notre cahier des charges, nous avons dû essayer de planifier tout ce projet. Pour ce faire, nous avons trié les fonctionnalités par ordre de priorité et de difficulté afin de pouvoir se répartir les tâches et car certaines fonctionnalités ne peuvent être réalisées tant que d'autres ne sont pas développées. Vous trouverez le planning réalisé ci-dessous :

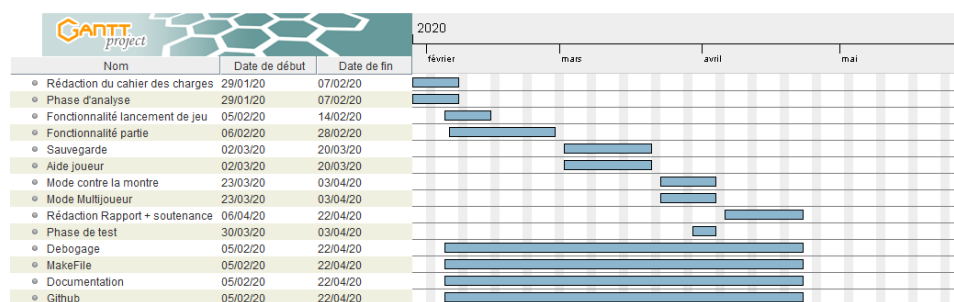


FIGURE 1 – Macro-planning V1

Afin de présenter notre planning, nous avons choisi d'utiliser la méthode Gantt. En effet, c'est une méthode efficace qui permet aux membres d'un

groupe de visualiser dans le temps les différentes tâches à réaliser. Nous pouvons ainsi nous rendre compte si notre groupe est en retard ou en avance. Cette méthode possède des avantages et des inconvénients, les voici :

- Avantages :
 - Lecture simple et accessible à tous.
 - Le suivi peut s'effectuer facilement lors du déroulement de projet en comparant les dates prévues et les dates réelles.
 - La visualisation de dates de débuts et de fins de tâche sur l'échelle de temps facilite la lecture du graphique.
- Inconvénients :
 - Le problème étant qu'il n'y a pas de mise en évidence des relations d'antériorités existantes entre les différentes tâches.

2.3 Répartition des tâches

Après avoir réalisé le cahier des charges et le diagramme de Gantt, nous avons pu vérifier la faisabilité de notre projet. Nous avons dans un premier lieu décidé de ne pas répartir les tâches par membres mais de laisser chaque membre travailler sur la tâche de son choix en respectant les priorités des tâches.

Voici les principales tâches effectuées par chaque membre.

Tâches	Asmae	Guillaume B.	Guillaume D.	Quentin
Cahier des charges				
Diagramme de Gantt				
Déterminer la listes des fonctions				
Normes et structures				
Github				
Création des fichiers thèmes				
Fonctionnalité Lancement de jeu				
Fonctionnalité Partie				
Fonctionnalité Sauvegarde				
Fonctionnalité d'aide au joueur				
Fonctionnalité "Mode contre la montre"				
Gestion des comptes rendus de TP				
Démo 1				
Démo 2				
Démo finale				
Interface graphique				
Doxygen				
Tests				
Rapport Final				

FIGURE 2 – Répartitions des tâches

Une case verte signifie que le membre du groupe à réussi à réaliser la tâche avec succès. Une case rouge signifie que le membre a tenté un essai non concluant. Et enfin, une case blanche signifie qu'aucune tentative n'a été réalisé par ce membre du groupe.

2.4 Outils utilisés

Lors de ce projet, nous avons utilisé plusieurs outils et sites internet afin de le réaliser au mieux. Voici une liste de ceux que nous avons utilisés :

2.4.1 Outils de communication :

- Discord : C'est un logiciel développé par Discord Inc., il permet de discuter oralement avec d'autres personnes ou par chat. Lors de ce projet, nous avons utilisé ce logiciel afin de garder un contact vocal malgré les circonstances du confinement lors des séances de tps ou encore hors de ceux-ci.
- Facebook Messenger : C'est un logiciel de messagerie instantané développé par Facebook. Ce logiciel nous a principalement permis de discuter lors du projet afin de mettre en place nos réunions ou même de discuter en dehors des tps de projet.

2.4.2 Outils de développement :

- Éditeur de texte pour coder (Bracket, Atom) : Afin de réaliser le code du projet, chacun a utilisé l'éditeur de texte qu'il souhaitait. Le projet a été réalisé en C. Certains ont ainsi choisi d'utiliser Atom tandis que d'autre utilisaient Brackets.
- Glade : Glade est un outil de conception d'interface graphique pour GTK+ (bibliothèque écrite en C que nous utilisons pour la création de nos interfaces graphiques). Il permet au développeur de lui simplifier la tâche de création et la gestion des fenêtres graphiques et lui permet de se concentrer sur le code qui permet à l'interface de fonctionner. Ces interfaces graphiques sont enregistrées en tant que fichiers XML, et sont ensuite chargées par le module builder de GTK+.

2.4.3 Outils de gestion de développement :

- Github : C'est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git. Outre le fait que l'on puisse y stocker les fichiers pour le jeu,

Github propose une fonctionnalité de versionning (gestion des versions d'un fichier) dans le cas où l'on souhaite revenir en arrière lors du projet suite à la prise de certaines décisions. Nous avons utilisé ce site afin de stocker tous nos fichiers pour la réalisation du projet comme demandé par les professeurs.

2.4.4 Outils de documentation :

- Doxygen : C'est un générateur de documentation sous licence libre capable de produire une documentation logicielle à partir du code source d'un programme. Comme demandé dans les consignes du projet, nous avons utilisé Doxygen afin de créer automatiquement de la documentation pour notre application.

2.4.5 Outils de gestion de projet :

- Trello : C'est un outil de gestion de projet en ligne, inspiré par la méthode Kanban de Toyota. Il repose sur une organisation des projets en planches listant des cartes, chacune représentant des tâches. Les cartes sont assignables à des utilisateurs et sont mobiles d'une planche à l'autre, traduisant leur avancement. Lors du projet, nous avons utilisé Trello afin de nous répartir les tâches, dans un premier temps, avant de basculer sur l'onglet "project" de notre projet Github.
- GanttProject : C'est un logiciel permettant de créer un diagramme de Gantt. Nous avons utilisé ce pour réaliser le macro-planning du projet. On retrouve ainsi les majeures fonctionnalités du projet sur la gauche de celui-ci et ensuite pour chacune d'entre elles, on accorde une période définie sur le calendrier.
Accès au micro-plannings : <https://github.com/Ame29/ProjetL2/projects/1>

2.4.6 Outils de de partage de fichiers :

- Google Drive : C'est un service de stockage et de partage de fichiers lancé par la société Google en 2012. Lors de ce projet, nous avons utilisé ce service afin de partager tous les documents relatifs au projet, parmi ceux-ci on retrouve les comptes rendus de réunion, ceux de tp ou encore des documents dont on pourrait avoir besoin comme la liste des consignes afin de se rappeler des contraintes exactes du projet.
- Google Docs : C'est un service de création de documents texte en ligne lancé par Google en 2006. Ce service fonctionne comme

n'importe quel autre traitement de texte, il est simplement accessible depuis n'importe quel navigateur internet. Nous avons utilisé ce service lors du projet afin de rédiger la plupart des documents nécessaires pour le projet comme le cahier des charges, les comptes rendus de tp / réunions ou encore le rapport final.

3 Analyse et Conception

3.1 Choix de conception

Lors de ce projet, le groupe a dû faire plusieurs choix dans la réalisation de celui-ci :

3.1.1 Nom du jeu :

Tout d'abord, nous avons dû faire un choix sur le nom de notre jeu. Pour cela, nous avons fait une liste de différents noms potentiels pour celui-ci :

- Kengo Tendo (mots mêlés en japonais)
- Méli Mélo
- Mish Mash (mots mêlés en anglais)
- Jumble

Après avoir fait un vote où la majorité l'emportait, il a été décidé que le nom du jeu serait "Mish Mash".

3.1.2 Particularités du jeu :

- Différents mode de jeu :
 - Mode "Classic" : Comme n'importe quel jeu de mots mêlés, il existe un mode dit "solo" où le joueur peut jouer une partie normale, seul. Il nous a donc semblé logique que ce mode de jeu soit inclus dans notre application.
 - Mode "Contre-la-montre" : Afin d'ajouter un peu d'originalité à cette application, nous avons eu l'idée de créer un mode contre la montre pour que le joueur puisse avoir un petit peu de challenge dans un jeu qui est d'ordinaire très calme.
 - Mode "Multijoueur" : Pour que le jeu change radicalement, nous avons eu l'idée d'intégrer un mode multijoueur à notre application. Ainsi au lieu de jouer une partie seule, ce qui n'est pas forcément amusant, le joueur peut inviter sa famille ou ses amis pour jouer une partie en local.
- Différents thèmes : Afin de pouvoir faire varier les différents mots à trouver, nous avons eu l'idée de créer plusieurs fichiers textes contenant une centaine de mots basés sur un thème précis. Par exemple,

on retrouve un thème lié aux animaux, un lié aux villes connues du monde, un aux métiers ainsi qu'un autre lié à la nourriture.

- Importance de l'aléatoire : Dans ce jeu, nous avons fait le choix que la création des grilles et des listes est aléatoire, ainsi d'une partie à une autre, il n'y a qu'une petite chance pour que le joueur ait déjà vu tous ces mots ensemble dans une partie précédentes. De plus chaque mot est placé aléatoirement dans la grille mais son sens est aussi aléatoire, il peut varier vers la gauche, vers la droite, vers le haut, vers le bas et en diagonale.

3.2 Fonctionnalités attendues

- Fonctionnalité Lancement de jeu : Lorsque l'utilisateur lance le programme, un menu s'affiche afin que celui-ci puisse choisir le mode de jeu (solo, contre-la-montre ou multijoueur) afin que le programme sache quel type de partie il va lancé. Dans le cas où c'est le mode solo qui a été choisi, on lui demande s'il souhaite commencer une nouvelle partie ou s'il souhaite charger une partie. Si démarrer une nouvelle partie est sélectionné, on demande au joueur de sélectionner un niveau de difficulté puis un thème, afin de déterminer le fichier de thème à utiliser ainsi que la taille de la matrice, via d'autres menus. Lorsque l'utilisateur a passé tous les menus, il peut enfin jouer via la fonctionnalité de partie. Dans le cas où le joueur choisi le mode contre la montre, la partie commence directement avec une taille définie.
- Fonctionnalité mode "Classic" : Une fois que le joueur a passé les différents menus, il peut jouer sa partie, la grille et la liste s'affichent, grâce à une interface graphique, et le joueur peut sélectionner la première et la dernière lettre lorsqu'il pense avoir trouvé un mot. Tant que la liste contient un mot, la partie continue et elle s'arrête une fois que le joueur a trouvé le dernier mot.
- Fonctionnalité Mode "Contre-la-montre" : Le mode Contre-la-montre est une partie solo tout à fait normale à la seule exception que le joueur doit trouver un maximum de mots voir tous les mots de la liste avant que le temps ne soit écoulé. Un système de score est aussi installé avec ce mode afin d'avoir la possibilité de battre ses propres records. Le score sera calculé en fonction du nombre de mots trouvés et du temps que le joueur a mis pour trouver chaque mot.
- Fonctionnalité "Mode Multijoueur" : Le mode Multijoueur est une partie normale, sauf que le joueur n'est pas seul à jouer. En effet, deux joueurs en local peuvent jouer grâce à ce mode. Chaque joueur a pour objectif d'avoir le plus haut score à la fin de la partie, afin que le tour

de chaque joueur soit équitable, nous mettrons en place un système de temps par tour qui sera fixe.

- Fonctionnalité de Sauvegarde (Gestion de sauvegarde) : Pendant que le joueur est en train de jouer une partie, il peut faire le choix d'interrompre sa partie afin d'y revenir plus tard. En appuyant sur un bouton de l'interface graphique, la grille et la liste sont sauvegardées dans un fichier texte. S'il souhaite reprendre sa partie, le joueur devra sélectionner "Charger une partie" dans les menus du début et il pourra ainsi reprendre sa partie là où il l'avait laissé.
- Fonctionnalité d'aide au joueur : Pendant que le joueur est en train de jouer une partie, dans le cas où il est bloqué et n'arrive pas à trouver de mots, nous avons mis en place une aide qui lui indique où se trouve la première lettre d'un mot de la liste. Il pourra accéder à cette aide via un bouton de l'interface graphique situé près du bouton pour sauvegarder.
- Interface graphique : L'interface graphique de notre jeu a été réalisée avec GTK, on peut ainsi retrouver la grille de jeu, une liste pour les mots à trouver, une liste pour les mots trouvés, un bouton pour sauvegarder sa partie et un bouton pour demander l'aide, une "case" pour voir quelle lettre ont été sélectionnées par le joueur et enfin une case pour le chronomètre et un bouton pour mettre le jeu sur pause.

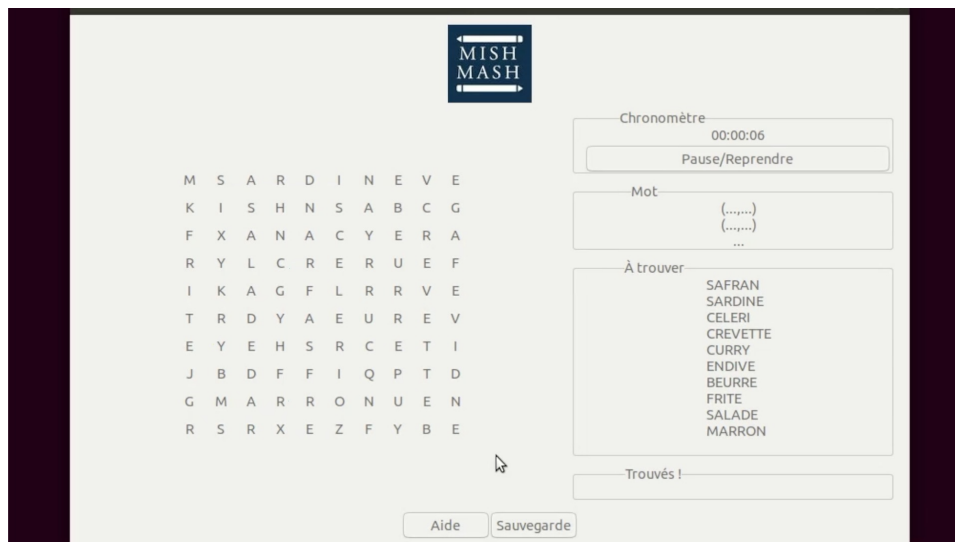


FIGURE 3 – Interface graphique lors d'une partie

- Gestion du temps : Pendant que le joueur est en train de jouer une partie, il peut décider de mettre en pause sa partie s'il souhaite l'inter-

rompre. Ainsi le joueur pourra cliquer sur le bouton “Pause/Reprendre”, pour activer la pause de sa partie.

3.3 Règles de notre jeu

Il a fallu également que nous nous mettions d’accord sur les règles du jeu. Comme nous souhaitions réaliser plusieurs modes de jeu, il a fallu faire plusieurs règles :

- Mode “Classic” : Le jeu de mots-mêlés doit permettre à un joueur de trouver des mots parmi la liste affichée à côté de la grille. Les mots peuvent être lus dans tous les sens (vers la gauche, vers la droite, en diagonale, vers le haut, vers le bas). Chaque mot peut partager une ou plusieurs lettres avec un autre mot. Pour sélectionner un mot, il faut rentrer les coordonnées de la 1ere lettre du mot ainsi que de la dernière lettre du mot.
- Mode “Contre-la-montre” : Le mode Contre-la-montre de notre jeu permet au joueur de jouer une partie en mode solo cependant il y a un chronomètre afin que le joueur ait une pression supplémentaire. Il doit donc découvrir autant de mots possible voir tous les mots de la liste avant la fin du chronomètre. Dans ce mode de jeu, la taille de la matrice sera définie et ne pourra changer.
- Mode Multijoueur : Le mode Multijoueur de notre jeu permet au joueur de jouer une partie avec 3 autres personnes maximum en local. Grâce à un système de tour et de temps par tour, chaque joueur a donc pour mission de trouver le plus de mot possible avant que son tour ne termine. Chaque joueur possède donc un score, et lorsque la partie se termine, celui ayant le plus haut score est le gagnant de la partie.

3.4 Comment jouer ?

Lorsqu’une personne lance notre programme, une interface graphique apparaît, c’est notre écran d’accueil. On y trouve un bouton permettant de lancer une nouvelle partie ainsi qu’un bouton pour charger une partie depuis une sauvegarde.

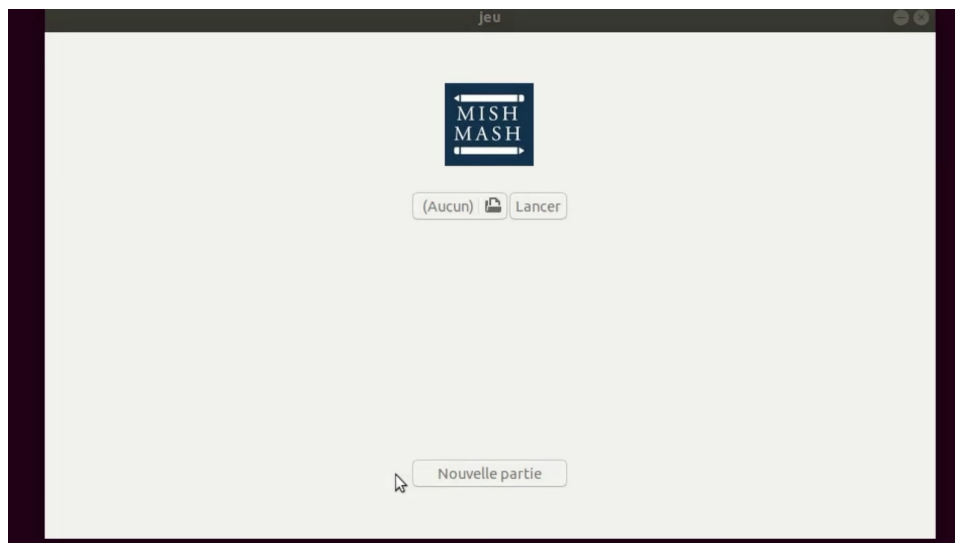


FIGURE 4 – Interface graphique lors du lancement du programme

Si le joueur choisit de lancer une nouvelle partie, il pourra ensuite sélectionner une difficulté ainsi qu'un thème pour sa partie.

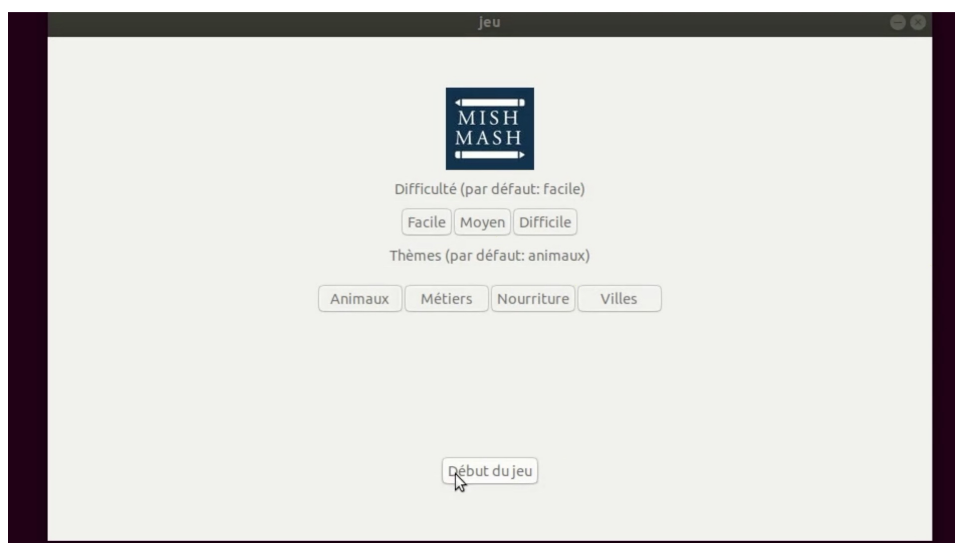


FIGURE 5 – Interface graphique : Menu difficulté et thème

Une fois ces deux conditions choisies, la partie commence. L'écran de partie apparaît alors, on y trouve la grille où les mots sont cachés, sur la droite de la fenêtre on retrouve un chronomètre ainsi qu'un bouton pour mettre le jeu en pause, un endroit où le joueur peut voir le mot qu'il a sélectionné, la liste des mots à trouver et la liste des mots trouvés. Deux

boutons sont situés dans le bas de cet écran de partie, celui permettant de sauvegarder sa partie et celui permettant de demander l'aide. Pendant une partie, le joueur doit sélectionner la première et la dernière lettre afin de créer un mot, si celui-ci est dans la liste des mots à trouver, alors il passe dans la liste des mots trouvés.

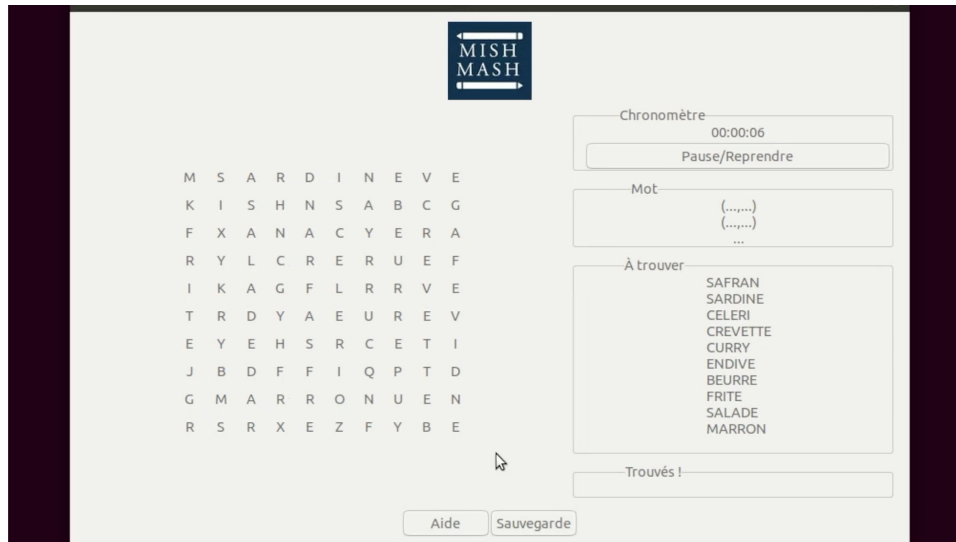


FIGURE 6 – Partie en cours



FIGURE 7 – Une partie peut être mise en pause

Pendant sa partie, si le joueur souhaite la sauvegardée pour y revenir plus tard, il n'a qu'à cliquer sur le bouton sauvegarde. De plus, il peut également demandé de l'aide via le bouton aide, le programme lui indique alors les coordonnées de la première lettre du premier mot présent dans la liste des mots à trouver.

La partie se termine quand le joueur a trouvé tous les mots, un écran de fin s'affiche alors où est indiqué un petit message de félicitations et le temps que le joueur a mis pour finir sa partie.

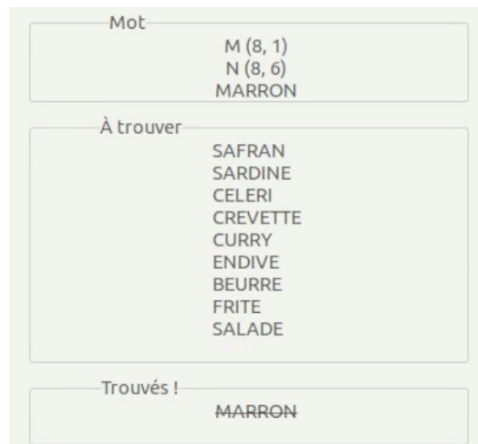


FIGURE 8 – Interface graphique : Listes des mots en cours de jeu

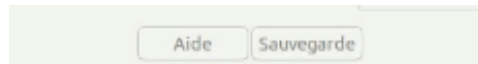


FIGURE 9 – Boutons aide et sauvegarde



FIGURE 10 – Fin de partie

4 Développement

Dans cette troisième partie, vous découvrirez comment nous avons réalisé le développement de notre jeu.

4.1 Structure de données

Nous avons trois principales structures autour duquel gravite tout le code.

- Matrice de jeu

Il s'agit de l'une des structures les plus importante. Avant d'avoir une interface graphique, il s'agissait d'une matrice de caractères de taille variable (selon la difficulté). Elle représente la grille de jeu visible par le joueur et dans laquelle on place les mots aléatoirement après les avoir sélectionnés. Une grande partie des fonctions gravite autour de cette matrice.

Pour l'interface graphique, chaque case de la grille est en fait un bouton GTK+ et contient une structure de type `info` (qu'on verra plus bas), ce qui permettra au joueur d'envoyer les bonnes informations au jeu quand le bouton est cliqué.

- Structure `mot_tire` :

Cette structure est utilisée pour les mots tirés du fichier de thème. Cette structure de données est composée d'un tableau de caractères qui permet de stocker le mot choisi par notre fonction qui tire un mot au hasard dans nos différentes bibliothèques de mots (thèmes). Elle contient aussi les coordonnées de ligne et de colonne de la première lettre ainsi que ceux de la dernière lettre du mot, lorsqu'il est placé dans la matrice dont nous avons parlé précédemment. Cela va nous permettre de stocker les mots dans un tableau de type `mot_tire`. Ce tableau sera ensuite utilisé pour la liste des mots à trouver. Un tableau du même genre est utilisé pour stocker la liste des mots trouvés (que nous créons principalement pour l'affichage de ces derniers, au fur et à mesure).

Lorsque que le joueur sélectionne un mot dans la matrice, lors du jeu, nous utilisons alors ces coordonnées dans la matrice pour déterminer si ce mot existe dans notre liste de mots à trouver. Si oui, il est déplacé dans la liste de mots trouvés, sinon, nous affichons quand même le mot sélectionné à l'écran, pour que le joueur se rende compte de son erreur.

Dans notre jeu, tel que nous l'avons actuellement, nous n'utilisons pas vraiment ces coordonnées que l'on stocke pour l'aide (nous les affichons cependant à l'écran quand le joueur clique dessus au cours d'une partie), mais il était dans notre intention de les utiliser pour "highlighter" les boutons correspondants dans l'interface graphique lorsque le joueur clique sur le bouton aide.

- Structure `info` : Cette structure de données est utilisée principalement pour l'interface graphique. Comme nous l'avons vu plus haut, chaque case de la grille de jeu est en fait un bouton GTK+. Ce bouton GTK+

est relié à plusieurs informations tel que le “label” du bouton, qui n’est autre que le nom du bouton (soit ce qui est affiché dessus, ici en l’occurrence une lettre du mot à trouver), ou encore la position du bouton dans la matrice de jeu, les coordonnées de la lettre sur la grille, etc. Ce qu’il faut savoir, c’est que lorsqu’un bouton GTK+ est cliqué, un signal “button clicked” est reçu par le programme. La fonction qui s’occupe de gérer ces signaux sous GTK+ ne prend que 2 paramètres : ledit bouton et un pointeur vers une quelconque structure.

Nous avons donc rassemblé toutes ces informations que nous voulons traiter dans une structure de données nommé “info”. Elle permet en résumé de rendre le bouton plus conscient de ce qui l’entoure, et nous permet de récupérer ses informations lorsque le joueur clique sur ledit bouton, dans le but de les traiter et faire avancer le jeu.

4.2 Fichiers

Notre projet s’appuie également sur une bonne gestion des entrées et sorties de fichiers. Voici les principaux usages que l’on en fait :

- Fichier de thème : Les matrices du jeu sont générées aléatoirement, lorsque le joueur choisit une difficulté (taille de la matrice ainsi que nombre de mots dans celle-ci), ainsi que le thème souhaité. Les thèmes consistent en 4 fichiers, contenant une centaine de mots, dans lequel un algorithme de tirage de mots vient aléatoirement piocher. Ils sont alors placés, à nouveau de façon aléatoire, dans la matrice par un algorithme de placement (directions : nord, sud, est, ouest).
- Fichiers de sauvegarde : Lorsque le joueur clique sur le bouton “sauvegarder”, en cours de partie, une sauvegarde est alors enregistrée automatiquement dans le dossier de sauvegarde.

Une sauvegarde se présente comme suit :

- sauvegarde_date_heures_minutes_secondes : ce fichier contient une sauvegarde de la matrice à ce moment. C’est sur ce fichier que l’on cliquera quand on voudra charger cette sauvegarde plus tard.
- un dossier listes qui contient les fichiers à charger dans nos deux structures de mots :
 - mots_trouves_date_heures_minutes_secondes
 - mots_trouves_date_heures_minutes_secondes

Charger une partie revient alors à charger chacun de ces fichiers dans la bonne structure. La partie date_heures_minutes_secondes est générée automatiquement à l’aide de la librairie time.h et fait office d’ID pour chaque partie que l’on veut sauvegarder/charger.

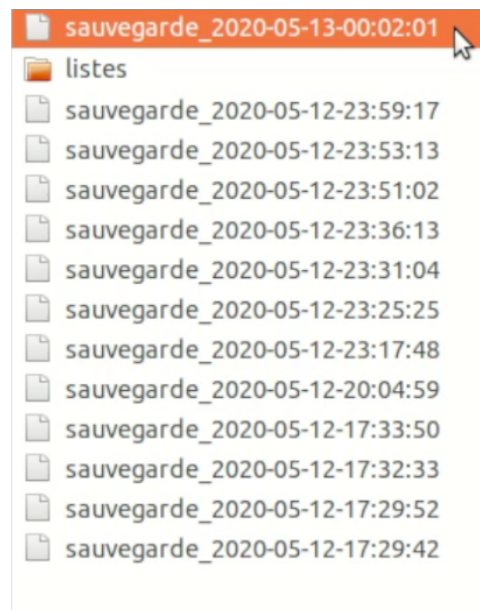


FIGURE 11 – Les fichiers de sauvegarde



FIGURE 12 – On peut alors lancer la sauvegarde

4.3 Bibliothèques utilisées

En plus des bibliothèques standards et connues du C (time.h pour la gestion du temps, string.h pour la gestion des chaînes de caractères), la principale bibliothèque que nous devons présenter est celle utilisée pour l'interface graphique.

GTK+ est une bibliothèque pour la création et gestion d'interface graphique. Cette bibliothèque a été développée pour les besoins du logiciel de traitement d'images GIMP ainsi que pour les environnements de bureau GNOME. Même si elle a été principalement utilisée sous Linux à ses débuts, elle peut aussi être utilisée sous Windows ou Mac. Cette bibliothèque est écrite en C et a une très bonne documentation et un joli rendu final, ce qui nous a grandement facilité la tâche au moment de l'implémentation, en sachant que nous n'avons commencé à travailler sur l'interface graphique qu'assez tard dans le projet. De plus, cette bibliothèque, bien qu'écrite en C, repose sur un paradigme plutôt orienté objet, ce qui rend la programmation des différents modules de l'interface graphique bien plus facile, puisque l'on manipule de façon très intuitive des objets aux comportements bien définis.

5 Résultats

5.1 Résultat

Comme vous pourrez le voir sur la vidéo de démonstration, disponible en téléchargement ici : https://github.com/Ame29/ProjetL2/blob/master/D%C3%A9mo/TP2_MishMash.mp4. Notre groupe a réussi à réaliser la plupart des fonctionnalités auxquelles nous avions pensé lors de la rédaction du cahier des charges au début du projet.

Cependant dans notre cahier des charges, nous avons souhaité réaliser certaines fonctionnalités originales mais par manque de temps et car nous avons perdu du temps sur la réalisation d'autres fonctionnalités qui nous ont posé soucis, nous n'avons pas réussi à les rendre concrètes.

5.2 Fonctionnalités implémentées

Parmi toutes les fonctions présentées plus haut dans la partie "Fonctionnalités attendues", nous n'avons malheureusement pu réaliser toutes celles qui étaient prévues. Vous trouverez ci-dessous une liste des fonctionnalités qui ont pu être réalisées :

- La fonctionnalité "Lancement de partie"
- La fonctionnalité mode "Classic"
- L'interface graphique
- la fonctionnalité de sauvegarde

Ainsi le mode Multijoueur et le mode Contre-la-montre n'ont pas pu être réalisés suite à des retards pris lors du développement du mode Classic.

Cependant, malgré le fait que certaines n'aient pas pu être développées, d'autres fonctionnalités ont été développées (la gestion du temps/pause) mais l'intégration dans la version avec l'interface graphique a été compliquée puis finalement abandonnée. C'est notamment le cas de la fonctionnalité d'aide au joueur.

5.3 Respect du planning prévisionnel

En nous basant sur les différents comptes rendus réalisés lors des séances de TP, nous avons pu comparer les dates prévues aux dates réelles. Voici le diagramme de Gantt pour visualiser comment le projet c'est réellement déroulé.

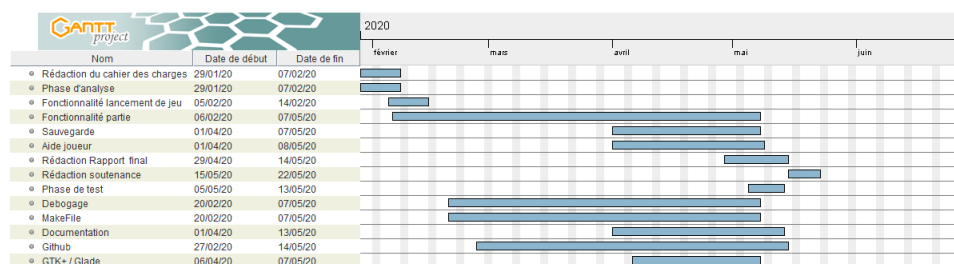


FIGURE 13 – Macro-planning V2

En comparant ce planning avec celui qui a été réalisé en début de projet, on peut remarquer que plusieurs choses ont changé. Tout d'abord, les dates de certaines fonctionnalités ont été modifiées, c'est le cas de la plupart des fonctionnalités de partie, sauvegarde, lancement du jeu ou encore aide au joueur. Ensuite, certaines fonctionnalités ont été effacées de ce Gantt car elles ont été abandonnées au cours du projet. C'est notamment le cas du mode Multijoueur et Contre-la-montre qui figuraient sur l'ancien planning. Enfin, certaines fonctionnalités ont été rajoutées lors de ce planning suite à des choix faits lors du projet. C'est le cas de la fonctionnalité GTK+ / Glade. Il y a aussi une fonctionnalité qui a été rajoutée, la rédaction de la soutenance, qui dans l'ancien planning était compris avec la rédaction rapport.

Ainsi ce planning montre que malgré le fait que certaines fonctionnalités aient changées voir aient été supprimées, notre groupe a su s'adapter aux retards qui avaient été pris.

5.4 Améliorations possibles

Comme pour n'importe quel projet, des améliorations sont possibles aussi bien au niveau de l'équipe, que du jeu en lui-même. Tout d'abord, au niveau du groupe de projet, nous aurions pu améliorer la communication au sein de celui-ci, car ne nous connaissant pas de base, il a fallu que chacun s'adapte aux autres, ce qui a eu pour effet de rendre la communication hors des TPs assez compliquée. De plus, notre groupe a peut être visé un peu trop haut, au niveau de ce que nous pouvions réaliser en code, au début du projet, il faudrait donc que l'on améliore cela également.

Ensuite, au niveau du jeu en lui-même, il faudrait intégrer les fonctionnalités qui n'ont malheureusement pas pu être réalisées (voir ci-dessus dans la partie fonctionnalités implémentées). Mis à part ces fonctionnalités, on peut dire qu'une amélioration majeure de ce jeu serait que les joueurs puissent jouer en ligne en multijoueur, une autre amélioration possible pour notre jeu serait une amélioration au niveau de l'esthétique de l'interface graphique.

5.5 Bilan critique

Afin de réaliser un bilan le plus critique et objectif possible, chaque membre du groupe va réaliser un bilan individuel sur son ressenti pendant le projet. Puis un bilan collectif sera réalisé en combinant tous les points qui sont ressortis.

5.5.1 Bilan individuel

- Asmae Bouhandi : Ce projet a été pour moi une expérience intéressante. En effet, ce fut ma première expérience de réalisation de projet. J'ai beaucoup appris à la fois en programmation (notamment avec la bibliothèque GTK+), en gestion de projet ainsi qu'en communication relationnel. L'organisation a été assez difficile. Nous n'avions pas tous la même motivation ni le même investissement tout au long du projet et cela a impacté sur le bon déroulement du projet sur la durée. En effet, certains membres ont fini par endosser plus de responsabilités pour corriger le déséquilibre occasionné et mener le projet à bien. Mais pour un premier projet, je suis très satisfaite du résultat obtenu..
- Guillaume Briffault : Ce projet m'a beaucoup appris. Tout d'abord, parce que celui-ci est le premier réel projet réalisé. Nous avons donc rencontré quelques problèmes techniques, des problèmes de temps mais aussi liés au confinement. Il a fallu s'organiser pour faire le maximum pour ce projet. Cette expérience m'a beaucoup appris et j'ai été ravi de travailler avec mes camarades.

- Guillaume Derrien : Je suis au final assez content de ce que mon groupe a pu produire lors de ce projet, bien évidemment nous aurions pu faire plus mais je pense aussi que nous avons visé un peu haut par rapport à nos compétences. J'ai principalement pu m'améliorer en gestion de projet, domaine dans lequel j'avais des lacunes. J'ai enfin été heureux de travailler avec mes camarades sur ce projet.
- Quentin Bellanger : À l'origine peu emballé par l'idée de mener à bien un projet en groupe, que ce soit en raison de mes lacunes ou par peur de ne pas le mener à bien, j'estime à contrecœur que mes craintes étaient fondées car je ne suis pas autant satisfait du résultat que je le voudrais, bien qu'il reste plus que décent. De plus malgré les quelques problèmes, soit d'organisations, soit techniques que nous avons pu rencontrer, je considère que l'expérience acquise durant ces quelques mois n'est cependant pas à jeter et qu'elle me permettra de ne pas reproduire des erreurs passées. Finalement, d'un point de vue ambiance de groupe, je remercie mes camarades motivés avec lesquelles je suis content d'avoir pu travailler.

5.5.2 Bilan collectif

En conclusion, on peut voir que le groupe est satisfait du travail rendu, néanmoins il aurait pu être meilleur avec une meilleure gestion de projet et une meilleure communication au sein du groupe. Cela a eu pour effet d'occasionner une baisse de motivation chez certains membres du groupe ainsi qu'un sentiment de déception en cours de projet.

Lors de ce projet, on peut cependant noter que chacun est content d'avoir beaucoup appris sur comment se passe un projet dans des conditions réelles, c'est-à-dire souvent bien différent des prévisions que l'on se fait au début et avec beaucoup d'ajustements en cours de route.

6 Annexes :

6.1 Annexe n1 : Jeux de tests

Vous trouverez les jeux de tests dans le dépôt git dans la partie test.

Cas valides	Jeu de test	Résultats
Caractère spécial	.	Caractère valide
Caractère lettre minuscule	a,b,z	Caractère valide
Caractère lettre majuscule	A,B,D	Caractère valide

Cas invalides	Jeu de test	Résultats
Caractère chiffre	0,5,7	Caractère invalide
Caractère spécial autre que '.'	!, ?,)	Caractère invalide

FIGURE 14 – Jeu de test du fichier tests AffGrille.c

Cas valides	Jeu de test	Résultats
Caractère spécial	.	Caractère valide

Cas invalides	Jeu de test	Résultats
Caractère chiffre	0,5,7	Caractère invalide
Caractère spécial autre que '.'	!, ?,)	Caractère invalide
Caractère lettre minuscule	a,b,z	Caractère invalide
Caractère lettre majuscule	A,B,D	Caractère invalide

FIGURE 15 – Jeu de test du fichier tests initGrille.c

Cas valides	Jeu de test	Résultats
Fichier existant	fichier!=NULL	Fichier valide

Cas invalides	Jeu de test	Résultats
Fichier vide		Fichier invalide
Fichier inexistant	fichier==NULL	Fichier invalide

FIGURE 16 – Jeu de test du fichier tests TirerMot.c

Cas valides	Jeu de test	Résultats
Caractère lettre majuscule	A,B,D	Caractère valide

Cas invalides	Jeu de test	Résultats
Caractère chiffre	0,5,7	Caractère invalide
Caractère spécial	!, ?,)	Caractère invalide
Caractère lettre minuscule	a,b,z	Caractère invalide

FIGURE 17 – Jeu de test du fichier tests voidToChar.c