Rapport de soutenance 1

présenté par :
Arnaud CORCIONE - Corentin BELLONI - Daniel NGOUESSY BOUSSAMBA
Ethan GIRARD - Jessy COURTEMANCHE

Table des matières

1 Introduction

Ce projet consiste à développer un jeu pour deux joueurs, dans lequel cesderniers doivent taper le plus rapidement possible les suites de mots générées par l'application. Celui qui tape les mots le plus rapidement remporte la partie. Ce jeu, dans l'esprit compétitif, vise à améliorer la rapidité et la précision de frappe des joueurs, tout en offrant une expérience de jeu amusante et interactive.

Pour atteindre cet objectif, nous allons programmer en C et utiliser des bibliothèques et des outils appropriés pour créer une interface utilisateur attrayante et une expérience de jeu fluide.

1.1 Présentation des membres

1.1.1 Arnaud Corcione

Depuis le début de ma jeunesse, grâce à mon père , j'ai été bercé dans le monde de l'informatique, ce qui m'a permis de m'intéresser à ce monde étrange. A mon entrée à Epita, j'ai eu une révélation pour le code, qui maintenant ne quitte pas mon quotidien. J'ai également pu réaliser des projets touchant à plusieurs langages. Ainsi, se creuser la tête pour apprendre et à comprendre une nouvelle technologie est passionnant pour moi.

1.1.2 Corentin Belloni

Je suis quelqu'un qui est passionné par l'informatique. J'aime découvrir de nouvelles technologies et mettre mes connaissances en pratique pour résoudre des problèmes complexes. Je suis également quelqu'un de joyeux, qui aime travailler en équipe et partager mes connaissances avec les autres. Je suis très investi dans mon travail et j'aime relever des défis. Je suis également très travailleur et je m'efforce toujours de donner le meilleur de moi-même.

1.1.3 Ethan Girard

Etant issu d'une terminale scientifique, j'ai été pris de passion par l'informatique en classe de première avec la spécialité "Numérique et sciences informatiques". Je suis quelqu'un qui aime l'informatique et le sport en général, je suis également passionné par les technologies et l'innovation. Je suis déterminé et engagé, prêt à me donner à fond pour créer un projet de A à Z.

1.1.4 Daniel NGOUESSY BOUSSAMBA

J'ai découvert cette passion qu'est l'informatique il y a quelques années et depuis, je m'efforce d'en apprendre toujours plus sur les dernières technologies et tendances dans ce domaine. Je suis également dynamique et motivé, ce qui me permet de m'investir pleinement dans tous les projets auxquels je participe. Je suis convaincu que mon enthousiasme et mon dévouement seront un atout pour réussir ce projet d'école.

1.1.5 Jessy Courtemanche

Depuis jeune, mes frères m'ont inculqué la passion du jeu vidéo, puis de fil en aiguille je me suis intéressé au domaine de l'informatique et chacun de mes centres d'intérêts me stimulaient d'avantage que les anciens. Il y a eu le hardware, le gaming que cela soit entre copains ou en mode compétition (l'eSport), mais aussi le software.

La programmation m'a toujours particulièrement intéressé car c'était une énigme pour moi, et surtout je savais que tôt ou tard j'allais pouvoir développer mes compétences techniques dans ce domaine.

A mon entrée à EPITA, j'avais seulement vaguement pratiqué le langage C pour utiliser Arduino ou bien des logiciels comme Flowcode permettant de contrôler des microcontrôleurs inclus dans des systèmes plus importants. Au-delà de ce que nous étudions en Sciences de l'Ingénieur, j'ai acheté un kit Arduino qui m'a permis de découvrir les bases du C et de la programmation en général. Mais tout ceci sans vraiment comprendre la logique camouflée derrière de tels outils. Je pense que notre projet va me faire découvrir de nouvelles choses. La programmation est pour moi une source constante d'apprentissage, le domaine étant tellement vaste.

1.2 Répartition des tâches

Tâches	Personne concernée
Serveur	Arnaud
Client	Daniel
Affichage du jeu	Corentin
Générateur de texte	Ethan
Chat	Jessy
Site Web	Jessy

2 Serveur

2.1 Introduction

Lorsqu'on entreprend un projet impliquant un réseau, il est crucial de prendre en considération l'implantation d'un serveur dès le début. Que ce soit pour un site web, une application en ligne, ou tout autre type de projet, l'utilisation d'un serveur offre de nombreux avantages. En effet, cela permet de mieux gérer le flux de données, de garantir la disponibilité et la stabilité du système. De plus, cela peut aider à réduire les coûts à long terme en évitant les problèmes qui pourraient survenir pendant le développement en local.

Dans ce contexte, il est donc fortement recommandé de considérer l'implantation d'un serveur dès les premières étapes de conception de notre projet Digit-Dash. Arnaud, disposant d 'un serveur chez lui, s'est occupé de commencer l'implémentation du serveur. Daniel et Jessy se sont concentrés sur le Client.

2.2 Serveur

Lorsqu'on souhaite créer un serveur, il est important de comprendre son fonctionnement ainsi que les protocoles de communication utilisés pour transférer des données sur un réseau.

2.2.1 TCP (Transmission Control Protocol)

TCP, Transmission Control Protocol, est un protocole de communication qui assure un transfert fiable et ordonné de données entre des ordinateurs sur un réseau. Ce protocole fonctionne en établissant une connexion entre l'expéditeur et le destinataire. Cette connexion se compose de trois étapes.

Tout d'abord, le client envoie une demande de connexion (SYN) au serveur. Le serveur répond avec un accusé de réception (SYN-ACK) pour confirmer la demande et établir la connexion. Enfin, le client envoie un accusé de réception (ACK) pour confirmer la réponse du serveur et finaliser la connexion.

Une fois la connexion établie, TCP divise les données à envoyer en segments plus petits. Chaque segment est numéroté et envoyé séparément. Le destinataire reçoit les segments, les confirme et renvoie un accusé de réception pour informer l'expéditeur que les données ont bien été reçues. Si le destinataire ne reçoit pas un segment, il demande à l'expéditeur de le renvoyer.

Enfin, lorsque toutes les données ont été envoyées, TCP ferme la connexion en envoyant un message de fin de connexion (FIN). Le destinataire répond par un accusé de réception de fin de connexion (FIN-ACK), puis ferme la connexion.
En résumé, TCP est un protocole fiable et ordonné qui établit une connexion, divise les données en segments, utilise des accusés de réception pour assurer que les données sont reçues sans erreur.

2.2.2 UDP (User Datagram Protocol)

UDP (User Datagram Protocol) est un protocole de communication sans connexion qui envoie des paquets de données (appelés "datagrammes") de manière rapide et peu fiable sur un réseau.

Contrairement à TCP, UDP ne nécessite pas d'établir une connexion avant d'envoyer des données. Les datagrammes sont envoyés sans aucune garantie que le destinataire les reçoive, et aucune confirmation de réception n'est envoyée. Les datagrammes sont envoyés directement au destinataire sans suivre un chemin de communication spécifique.

UDP est souvent utilisé pour des applications qui nécessitent une transmission rapide de données en temps réel, telles que la diffusion en continu de vidéo et audio, les jeux en ligne ou les applications de voix sur IP. Cependant, cette rapidité de transmission se fait au détriment de la fiabilité, car les données peuvent être perdues ou arrivées dans le désordre.

UDP est également utilisé pour les applications qui ont besoin d'une faible latence, c'est-à-dire une durée minimale entre l'envoi et la réception des données.

2.2.3 Implémentation

Pour notre serveur, on a décidé d'utiliser le protocole TCP, en effet pour l'instant la fiabilité des données est primordiale pour nous. De plus, il est primordial de gérer plusieurs connexions simultanées, ce qui est préférable pour un jeu. Ainsi à chaque connexion, on crée un nouveau processus qui s'occupe de gérer les requêtes faites par le client indépendamment.

De plus, à chaque requête le serveur renvoie une requête au client avec le message “received” ce qui assure que le serveur à bien reçu le message du client. Ce message est la a but informatif pour nous, car bien que nous utilisons le protocole TCP nous n’avons pas la main dessus ainsi nous avons besoin de savoir si le bon serveur a reçu notre message.

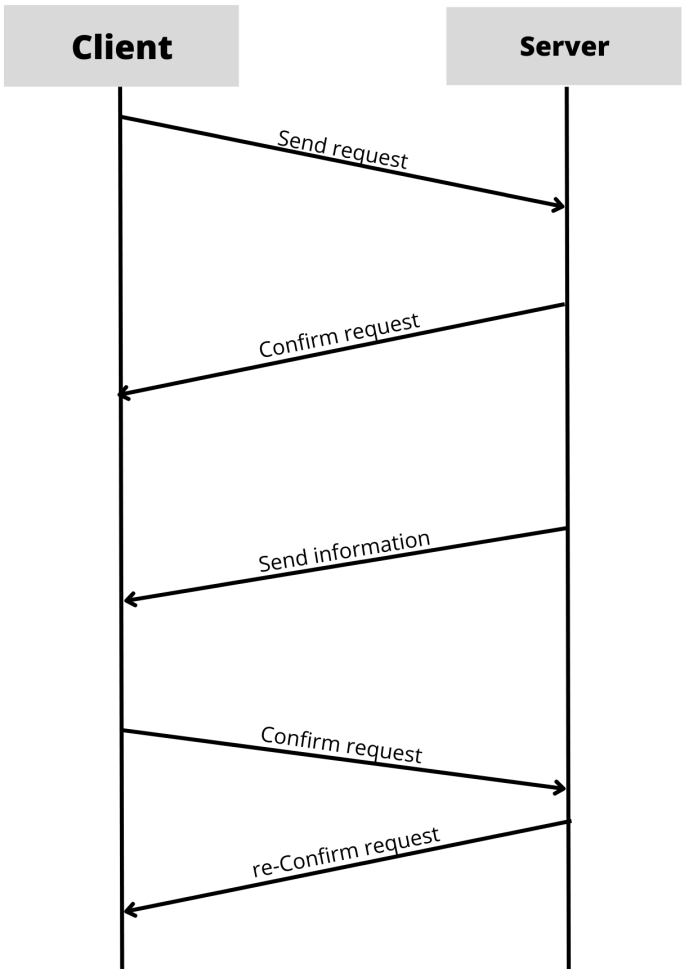

Figure 1 – Diagramme Communication Serveur-Client

3 Client

Un client est un logiciel informatique qui fait des requêtes à un serveur. Le client envoie une requête au serveur via son adresse et son port (canal de communication), qui varient selon le protocole utilisé. Comme mentionné précédemment, un serveur reçoit des requêtes et répond aux clients.

3.1 Caractéristiques d'un client

Par conséquent, le client est caractérisé comme suit :
- Il établit une connexion à un serveur en spécifiant un ou plusieurs ports réseau.
- Lorsqu'une connexion est acceptée par le serveur, il communique selon les exigences du protocole utilisé.

3.2 Caractéristiques du protocole d'échange Client-Serveur

- Client et serveur doivent bien sûr utiliser le même protocole.
- L'échange peut être effectué sur le réseau ou parfois localement.
- Les clients et les serveurs doivent définir, connaître et comprendre ce protocole (TCP dans notre cas).

3.3 Types de clients applicatifs

Notre serveur a besoin d'accueillir un grand nombre de clients, il est donc nécessaire de connaître le type de ces clients et de définir la taille de leurs échanges avec le serveur.

3.3.1 Client léger

Un client léger est une application dans laquelle le traitement des requêtes client est entièrement effectué par le serveur, et le client satisfait les réponses que le serveur calcule et envoie qu'il reçoit et formate pour l'affichage.

Les avantages sont les suivants
- Le niveau client nécessite très peu de puissance de calcul.
- La mise à jour de l'application se fait uniquement sur le serveur, à l'exception éventuelle de la mise à jour des clients.
- Les applications et les serveurs sont développés avec une plus grande indépendance vis-à-vis des clients et de leurs environnements.
- Travail de développement centré sur le serveur

3.3.2 Client lourd

Un client lourd est une application dont le traitement s'effectue principalement sur un ordinateur local appelé le client. L'objectif principal du serveur est de répondre aux demandes de données du client.

Quelques avantages :

— Le client peut parfois fonctionner même lorsque le serveur est déconnecté

— Une partie du traitement est effectuée par le client, soulageant les ressources du serveur.

— Plus d'indépendance au niveau du temps de réponse du réseau et du serveur

3.3.3 Client riche

Un client riche est une application dans laquelle le traitement des requêtes client est principalement effectué par le serveur. Le client reçoit des réponses "à moitié terminées" et les finalise. Il s'agit d'un client léger plus avancé qui permet des fonctionnalités équivalentes à celles d'un client lourd. C'est un compromis entre clients légers et clients lourds.

3.4 Implémentation

Nous avons prévu pour de partir sur l'utilisation d'un client lourd pour le début du projet. Ce choix pourra cependant évoluer en fonction de nos différents besoins. L'implémentation prévue est la même que celle du serveur. Le client envoie une requête au serveur et reçoit une réponse de ce-dernier. Pour les tests actuels, un léger manuel d'aide a été mis à disposition du client afin que l'utilisateur se connectant puisse connaître les commandes qu'il est autorisé à entrer.

```
Enter a request : hey
Unknown request. Enter [help] to know what command is available.
Enter a request : help
Here are the commands you are allowed to use
[server] --- Will allow you to connect to the server
[exit]   --- Close the client
[help]   --- How use availables command

Enter a request : server
Client connected : 3
Enter a request : exit
Message send : exit
Answer : received
Client disconnected : 3
Application closed.
```

FIGURE 2 – Connexion d'un client

4 Affichage du jeu

4.1 Introduction

Le fait d'afficher le jeu est une partie cruciale du projet car elle établit une communication entre le joueur et les instructions que le serveur envoie.

Pour cette première étape de développement, l'aspect visuel et les fonctionnalités de l'affichage restent très simples. Le but principal était de créer une interface fonctionnelle pour permettre de mener des tests approfondis sur les aspects techniques du jeu. Toutefois, dans le futur, une librairie sera utilisée pour afficher l'interface graphique, ce qui permettra de créer une expérience utilisateur plus intuitive et plaisante.

4.2 Affichage du niveau

Lorsqu'une partie est lancée, le jeu doit afficher sur le terminal le code à écrire en gris clair, presque transparent, pour montrer au joueur ce qu'il doit écrire. Ce code prend la forme d'un motif que le joueur doit respecter pour terminer le niveau. Pour ce faire, le programme charge le niveau approprié à partir de son fichier correspondant, puis écrit le code dans la console du joueur, ligne par ligne, sans inclure l'indentation. L'indentation est actuellement ignorée dans cette version de développement pour faciliter les tests.

Pour que le jeu fonctionne correctement, le joueur doit être informé de toute faute de frappe qu'il a commise et de l'endroit où elle s'est produite. Afin de fournir cette information de manière claire, nous avons décidé d'utiliser un système de couleur simple.

Couleur	**Signification**
Gris	Aucun input.
Vert	L'input entré est correct.
Rouge	L'input entré est incorrect.

Table 1 – Tableau de correspondance des couleurs pour les entrées de l'utilisateur.

Afin d'afficher le code pour le joueur, le programme commence par lire le fichier correspondant au niveau en cours. Ensuite, il recopie le code dans la console du joueur en utilisant les couleurs qui ont été définies précédemment pour indiquer la signification de chaque entrée.

4.3 Vérification de l'input

Nous avons mis en place un algorithme pour vérifier en temps réel l'entrée de l'utilisateur lors de la partie. Cet algorithme fonctionne en parallèle avec l'affichage et a la fonctionnalité de déterminer la couleur à laquelle la prochaine lettre va être affichée dans le terminal.

Il dispose également d'une fonction qui permet de demander au joueur d'entrer un seul caractère à la fois. Ce caractère est ensuite traité et sauvegardé dans un fichier qui contient tous les inputs précédents du joueur ainsi que les futurs inputs. Ce fichier est très utile pour suivre l'avancée du joueur dans sa partie, pour savoir à quel moment il est arrivé à la fin du niveau, mais aussi pour savoir si le caractère saisi est correct ou non.

Pour vérifier si le caractère saisi est correct, l'algorithme lit et compare caractère par caractère le fichier du niveau initial ainsi que celui qui sauvegarde tous les inputs du joueur. Grâce à cette comparaison, le programme peut savoir si le caractère saisi est correct ou non, et peut donc décider d'afficher la lettre en vert si elle est correcte, ou en rouge si elle est incorrecte.

4.4 Retour en arrière

Lors de la mise en place de l'affichage, une difficulté majeure est apparue lorsque les joueurs se trompaient dans leur saisie, car il était impossible de corriger l'erreur sans tout recommencer. Nous avons donc ajouté une fonctionnalité pour supprimer une saisie erronée.

Nous avons envisagé de simplement supprimer le dernier caractère stocké dans le fichier qui enregistre toutes les saisies du joueur (le même fichier que celui évoqué précédemment). Cependant, nous avons rapidement constaté que cela était impossible en C, nous avons donc dû trouver une autre solution plus innovante.

La nouvelle approche consiste à insérer des caractères "NUL" chaque fois que le joueur souhaite supprimer un caractère, puis déplacer le pointeur du fichier d'un cran vers l'arrière pour pouvoir réécrire sur le "NUL". Cette approche permet d'éviter le bug d'affichage qui se produit lorsque le joueur efface plusieurs caractères consécutifs, car les caractères sont toujours présents dans le fichier même s'ils sont supprimés. L'ajout des caractères "NUL" permet de spécifier au programme de ne pas les afficher s'ils sont présents dans le fichier.

4.5 Fonctionnement global

Pour finir voici un schéma simple pour comprendre le fonctionnement global de l'affichage de Digit-Dash :

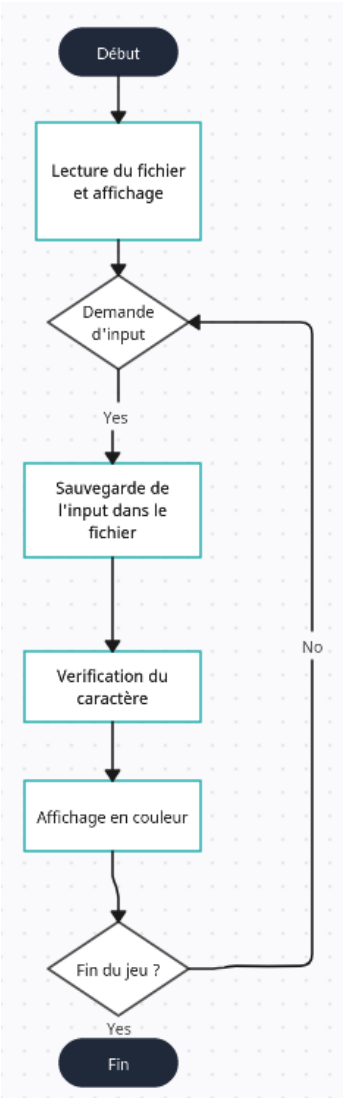

FIGURE 3 – Schéma du processus d'affichage

5 Générateur de texte

5.1 Introduction

Pour ce projet nous avions pour objectif de créer des lignes de code pour que les joueurs puissent essayer de les taper. Pour ce faire nous devions créer un système capable de générer un code via une base de donnée. L'une des tâches les plus importantes dans le traitement du langage est la détermination du sens d'un mot dans un contexte particulier.

Dans ce projet, nous avons développé un algorithme qui permet de déterminer le sens d'un mot dans un contexte particulier en utilisant un mot passé. Notre solution a été implémentée en utilisant le langage de programmation C. Dans la suite de ce rapport, nous allons expliquer en détail notre approche et comment elle fonctionne, ainsi que les algorithmes que nous avons utilisés pour développer notre solution. Nous allons également discuter des suggestions pour les améliorations futures.

5.2 Commencement

Pour commencer, nous avons dû nous pencher sur la tâche complexe de créer une liaison entre un mot futur et un mot actuel. Nous avons exploré différentes approches, notamment les chaînes de Markov. Cependant, nous avons rapidement réalisé que la modélisation des probabilités en C pour obtenir l'état suivant était une tâche ardue.

En effet, la modélisation des probabilités conditionnelles peut devenir très complexe, en particulier lorsque le nombre de variables augmente. Nous avons donc reflechi et nous nous sommes dit que pour y arriver nous allons fonctionner comme cela :

— obtenir une string via un une fichier
— créer une liste chaînée avec tous les mots comme valeur de la liste
— récupérer tous les index d'un mot recherché
— renvoyer le mot suivant grâce à aux indexs trouvés précédemment

5.3 Avancement

Pour ce qui est du programme en lui même, nous avons du mettre en place plusieurs fonction.

La toute premiere qui est utilisé dans le programme est la fonction **char *filetochar** qui permet d'obtenir une chaîne de charactère via un fichier. Par la suite nous avons donc implémenté la fonction **struct list *chartolist** qui comme énoncé précedemment sert à créer une list qui va prendre comme valeur tous les mots de la base de donnée. Cette fonction renvoie par exemple ceci :

```
ethandage@ethandage-Inspiron-5515:~/Documents/S4/Digit-Dash/game/find_word$ ./a.out "return"
[ (int; i=0;; i<50;; i++);  {; if; (i%2==0);  {; printf("Win!");; break;;
```

Ensuite, nous avons mis en place la fonction **struct list *chartolist**, qui est utilisée pour créer une liste chaînée contenant tous les mots de notre base de données.

Cette liste est ensuite utilisée pour déterminer le mot futur en utilisant la fonction **struct inlist *create_index_list**. Cette dernière fonction retourne une liste chaînée qui contient les index du mot donné. Cette liste nous est utile pour déterminer le mot futur en récupérant un de ces index de manière aléatoire, en lui ajoutant 1, et en récupérant ensuite le mot correspondant dans la liste du début (avec les mots).

Toutes nos fonctions nous donne ce résultat :

```
ethandage@ethandage-Inspiron-5515:~/Documents/S4/Digit-Dash/game/find_word$ ./a.out "if"
(i*y>i+y)
ethandage@ethandage-Inspiron-5515:~/Documents/S4/Digit-Dash/game/find_word$ ./a.out "if"
(i-y!=0)
ethandage@ethandage-Inspiron-5515:~/Documents/S4/Digit-Dash/game/find_word$ ./a.out "if"
(i-y!=0)
ethandage@ethandage-Inspiron-5515:~/Documents/S4/Digit-Dash/game/find_word$ ./a.out "if"
(i==0)
```

FIGURE 4 – Résultat de notre fonction via la chaîne de characatère "if"

5.4 À l'avenir

En conclusion, notre projet consistait à créer un programme en C capable de déterminer le sens d'un mot dans un contexte particulier en utilisant un mot passé. Nous avons dû mettre en place plusieurs fonctions pour arriver à cet objectif, notamment la fonction *filetochar, *chartolist et create_index_list.

En effet, une des limitations de notre programme est liée à la base de données que nous avons utilisée. Nous avons besoin d'une grande base de données pour obtenir les meilleurs résultats possibles, ce qui peut être difficile à obtenir.

Ainsi, dans une perspective d'amélioration, nous avons l'intention d'implémenter d'autres bases de données pour rendre le jeu plus complet. Nous pourrions proposer aux joueurs de choisir si ils veulent coder en C en Python etc.

En somme, nous sommes satisfaits du résultat obtenu jusqu'à présent et nous sommes impatients d'explorer de nouvelles voies pour améliorer notre programme à l'avenir

6 Chat

6.1 Introduction

L'objectif de cette partie est de créer un chat entre les deux utilisateurs pour qu'ils puissent communiqué après la partie. Cette partie va nécessité plusieurs connaissances déjà accise tels que les listes chainées, les structures et surtout le protocole réseaux TCP, cependant il sera aussi intéressant de faire des recherches sur les connaissances qu'on ne connait pas mais aussi pour peaufiner celles qu'on a déjà.

FIGURE 5 – Avancement du chat

6.2 Avancement

A la fin de cette première soutenance, le programme est capable de se connecter au serveur en utilisant les sockets, puis il demande le nom de l'utilisateur, l'étape d'après il demande le message que l'utilisateurs veut envoyer après cela la requête en voyage sous forme d'une chaine de caractère du type « Name : message ». Pour confirmer la réception le serveur renvoie « Received ». Etant dans une boucle While l'utilisateur peut envoyer autant de message qu'il veut au serveur.

6.3 Objectif pour la prochaine soutenance

Finir le système de chat, pour qu'un utilisateur puisse réellement communiquer avec un autre utilisateur cela va passé par l'utilisation de structure qui sera gérer pour le serveur pour le moment la structure à été créer et est fonctionnel coté client mais pas encore coté serveur donc pour l'instant l'utilisation d'un chaine de caractère est plus simple.

Ensuite il faudra un système de liste chainé pour gérer les conversations et pour finir l'utilisation d'un double pointeur regroupant pour chaque caractère l'adresse du nom de l'utilisateurs.

Enfin, pour la soutenance finale il faudra lier le système de chat à l'interface graphique.

7 Site web

7.1 Introduction

Dans le domaine des jeux vidéo, le site web de téléchargement est souvent considéré comme la première interface d'interaction avec les utilisateurs. En effet, c'est via cette plateforme que les joueurs pourront accéder et télécharger le jeu vidéo en question.

Par conséquent, la création et la conception d'un site web de téléchargement de jeu vidéo sont des étapes cruciales dans tout projet de développement de jeu. La qualité de ce site web est primordiale, car elle peut influencer l'opinion des utilisateurs sur le jeu dans son ensemble. Ainsi, il est essentiel de prendre en compte les besoins des utilisateurs et de fournir une expérience utilisateur optimale sur le site web de téléchargement.

En outre, des considérations telles que la sécurité, la vitesse de téléchargement, l'accessibilité et la compatibilité des navigateurs doivent également être prises en compte pour garantir une expérience utilisateur fluide et satisfaisante. En somme, la conception et le développement d'un site web de téléchargement de jeu vidéo sont des étapes clés dans la réussite d'un projet de développement de jeu vidéo. Pour sa création nous allons utiliser du HTML et du CSS.

7.2 Avancement

Après avoir choisi le thème rouge, nous avons consacré du temps et des efforts pour finaliser la première page de notre projet. Cette page comprend plusieurs espaces réservés pour des illustrations et du texte, qui seront ajoutés ultérieurement une fois que nous aurons finalisé la partie graphique de notre

projet.

Nous avons décidé de laisser ces espaces vides pour le moment, car nous souhaitons nous assurer que les illustrations et le texte finaux sont parfaitement cohérents avec le reste du projet et qu'ils se marient harmonieusement avec le thème rouge que nous avons choisi.

Malgré le fait que ces espaces soient encore vides, nous avons veillé à les placer stratégiquement sur la page pour qu'ils aient le plus d'impact possible lorsqu'ils seront remplis. Nous sommes impatients de voir le résultat final, qui devrait être esthétiquement plaisant et attractif pour nos lecteurs.

En outre, la première page de notre projet comporte un en-tête qui permettra aux utilisateurs d'accéder facilement aux différentes pages secondaires. Certaines de ces pages secondaires ne sont pas encore disponibles, car nous travaillons encore sur leur contenu. Toutefois, elles seront ajoutées dès que possible, et nous sommes convaincus qu'elles apporteront une grande valeur ajoutée à notre projet global.

En résumé, notre première page est prête et nous sommes impatients de la remplir avec des illustrations et du texte qui correspondent parfaitement à notre thème rouge. Nous sommes également en train de travailler sur les pages secondaires, qui viendront bientôt enrichir notre projet global.

8 Conclusion

En conclusion, notre projet Digit Dash sera un jeu en ligne innovant qui visera à améliorer les compétences de frappe des utilisateurs. Nous allons définir une variété de fonctionnalités pour offrir une expérience de jeu interactive et amusante, telles que la génération aléatoire de ligne de code, un système de chat pour une expérience interactive entre les joueurs, un menu pour choisir le mode de jeu et une room pour connecter les joueurs avant le début de la partie.

8.1 Objectif pour la suite

Notre prochaine étape consistera à atteindre deux objectifs principaux pour nos prochaines soutenances. Tout d'abord, nous nous concentrerons sur la finalisation des pages secondaires, qui expliqueront en détail les différents aspects et les règles du jeu. Nous voulons nous assurer que les joueurs ont toutes les informations dont ils ont besoin pour jouer et profiter du jeu.

Ensuite, nous nous concentrerons sur la préparation de la dernière soutenance, qui aura lieu ultérieurement. Lors de cette dernière étape, nous avons prévu de créer un formulaire pour recueillir les commentaires et les retours des joueurs, afin de pouvoir améliorer le jeu en conséquence.

De plus, nous allons travailler sur la création d'un exécutable qui permettra aux joueurs de télécharger le jeu directement à partir des boutons intitulés "jouer". Nous voulons que le processus de téléchargement soit le plus simple et le plus fluide possible pour les utilisateurs, afin qu'ils puissent commencer à jouer rapidement et facilement.

En somme, nos objectifs pour les prochaines étapes sont de finaliser les pages secondaires du jeu pour fournir aux joueurs toutes les informations dont ils ont besoin pour jouer et de travailler sur la création d'un formulaire pour recueillir les commentaires des joueurs afin d'améliorer le jeu. Nous allons également faciliter le téléchargement du jeu pour les joueurs en créant un exécutable téléchargeable directement à partir des boutons "jouer".

9 Bibliographie

— MonkeyType (https ://monkeytype.com)
— 10FastFingers (https ://10fastfingers.com)
— WebSocket (https ://en.wikipedia.org/wiki/WebSocket)
— Protocole UDP (https ://fr.wikipedia.org/wiki/User_Datagram_Protocol)
— Protocole TCP (https ://fr.wikipedia.org/wiki/Transmission_Control_Protocol)