

BOGGLE

RAPPORT

Développeurs : Fafa, Catalin, Lorraine, Samuel, Aranniy

Superviseur : Alexandra Rogova

3 avril 2022

INTRODUCTION :

Pour notre deuxième semestre de L2, nous avons été chargé de modéliser le jeu Boggle. Dans ce rapport, nous allons évoquer notre mode de travail, les fonctionnalités du jeu, ses atouts ainsi que les difficultés que nous avons rencontrés tout au long de notre avancée. Nous mentionnerons également toutes les idées que nous avons eu pour des versions futures du jeu.

Dans un premier temps, pour bien commencer ce projet, nous avons décidé de nous réunir assez tôt afin de discuter des différents éléments du jeu à implémenter, à savoir, la grille de lettres, la vérification des mots, le timer, les joueurs, les règles et les modes de jeu même si nous savions que cette dernière partie serait abordée de manière plus abstraite puisque nous étions encore qu'aux prémises de nos recherches. Nous pouvons également mentionner que jouer à la version de Boggle en ligne nous a fortement aidé à comprendre le fonctionnement du jeu et son environnement.

Au départ, nous avons eu du mal à démarrer notre projet. En effet, on ne savait pas vraiment si l'on devait commencer par créer les vérifications autour des mots, les joueurs, ou alors la grille en premier. Ça a été difficile de se mettre en route. C'est pour cela que nos réunions hebdomadaires avec notre superviseur nous ont été extrêmement utiles. Elle nous fournissait chaque semaines des idées, pistes de réflexions et des conseils pour ne pas se laisser submerger par le nombre de tâches à réaliser. Grâce à cela, nous avons pu nous mettre plus facilement d'accord dans les fonctionnalités à ajouter et leur temps de développement.

Nous avons commencé à modéliser le jeu avec différentes classes centrées sur une fonctionnalité particulière (une classe Jeu, Joueur, Grille etc). Une grande partie de la modélisation de départ a été modifiée après l'arrivée des utilisateurs et de tous les modes de jeu qui ont provoqué la création/fusion de nombreux fichiers. Toutefois, la base est restée identique ce qui nous a permis de ne pas nous égarer dans le développement du projet.

Pour notre première réunion nous nous sommes aussi concentrés sur la répartition des tâches le plus rapidement possible pour pouvoir bien démarrer. Nous nous sommes ainsi fixés l'objectif de chacun faire une tâche par semaine. Nous avons tenu ce rythme au départ, cependant, l'arrivée des devoirs notés, des contrôles et des cours en parallèles a ralenti notre progression. De même, certaines fonctionnalités ont été plus difficiles à implémenter que prévu et ont pris plus de temps, augmentant nos doutes sur nos capacités à finir notre projet à temps. Cela sera expliqué plus en détails lors de ce rapport. Malgré tout, nous n'avons pas abandonné et avons continué à travailler ensemble en s'aidant mutuellement, nous permettant ainsi de finir ce projet à temps avec un rendu final qui nous satisfait tous.

ENVIRONNEMENT DE JEU

Il existe deux versions : une textuelle, et une autre graphique réalisée sous JavaFX.

Version textuelle

Pour la version textuelle, le jeu de base a été reproduit fidèlement. Vous aurez accès à une grille avec la possibilité de choisir une lettre à chaque tour, la possibilité de valider le mot formé et ainsi de gagner des points ! Un timer est présent pour pouvoir mesurer le score de la partie. Des bonus ont également été rajoutés pour pouvoir aider le joueur en difficulté. La version textuelle n'a donc pas été délaissée : l'expérience de jeu ne sera pas amoindrie en choisissant de jouer à cette version plus rapide d'accès.

Version graphique

Avec la version graphique, on pourrait croire que l'on joue à une version officielle de Boggle ! En effet, cette version reprend toutes les règles du jeu de base, et bien plus. Lorsque vous lancerez le jeu, on vous demandera de vous connecter à votre compte, ou d'en créer un si c'est votre première fois. Vous pourrez ensuite choisir de jouer seul, contre une IA ou avec d'autres joueurs, de manière locale ou en ligne ! On a tenu à offrir aux joueurs le plus d'options possibles.

Vous pourrez également consulter vos statistiques de jeu et vos trophées obtenus ! En effet, pour permettre au joueur de ne pas se lasser rapidement, nous avons créé des tâches annexes à accomplir au cours d'une partie en vue de gagner des récompenses à savoir des trophées et des coins - la monnaie du jeu - ce qui lui permettra de pouvoir se payer des bonus afin de l'aider à réaliser le meilleur score possible. Vous pourrez ensuite finalement lancer une partie et jouer de manière très simple : il suffit de cliquer sur la case d'une grille pour pouvoir choisir une lettre, au lieu de devoir écrire ses coordonnées sur le terminal. De même, on prévient le joueur des cases déjà parcourues et celles non : plus besoin de devoir tout retenir ! Cette version est beaucoup plus immersive.

Comment démarrer une partie ?

Toutes les indications servant à compiler les deux versions du jeu se trouvent dans le fichier README ! L'outil Gradle a été utilisé afin de mieux structurer le projet et l'exécuter plus facilement.

GRILLE

La grille de jeu est une des premières modélisations que nous avons effectuée. Il s'agit en effet du corps du jeu, composé des 16 lettres parmi lesquelles le joueur doit repérer des mots. Nous avons donc, dans un premier temps, implémenté cela de sorte qu'à chaque lancement, une nouvelle grille de lettres se crée. Chacune des cases de cette grille reçoit de manière aléatoire une des 26 lettres de l'alphabet. Nous avons ensuite, après discussion lors d'un de nos entretiens, réalisé que cela ne convenait pas : Certaines lettres, que l'on ne retrouve que dans peu de mots (par exemple le z), apparaissent trop fréquemment et en trop grand nombre. Cela rendait la réalisation de mots appartenant au dictionnaire compliqué, voir même impossible dans certains cas. La grille de lettre obtenue n'était pas représentative car trop aléatoire. En effet, toutes les lettres de l'alphabet n'ont pas la même fréquence de distribution. Par exemple, les voyelles sont beaucoup plus fréquentes. Nous devions donc

implémenter une grille qui suivrait ce principe de fréquence de distribution des lettres. Nous avons donc, grâce à des tableaux de pourcentages d'apparition de lettres trouvés en ligne, pu corriger notre grille, et ainsi obtenir des grilles de jeu représentative, et surtout jouables.

En ce qui concerne l'affichage de la grille, nous avons commencé par une version textuelle. La grille prend alors forme d'un tableau avec des numéros de lignes et de colonnes. Pour sélectionner une lettre, le joueur renseigne donc ses coordonnées. La version graphique est quant à elle plus simple et instinctive : la grille est en fait constituée de boutons où il suffit de cliquer (ou d'entrer au clavier) pour ajouter la lettre correspondante au mot en construction.

IMPLEMENTATION REGLE DU JEU

Pour pouvoir reproduire fidèlement le jeu Boggle, il faut également implémenter les règles du jeu qui permettent le bon déroulement de la partie.

La première étant la construction des mots. Ces mots doivent être composés de lettres adjacentes sur la grille. Pour vérifier cela, nous récupérons les coordonnées de la case jouée et nous les stockons dans un attribut `LastCase` puis nous vérifions que la prochaine lettre choisie est bien une de ses voisines.

Ensuite, une case ne peut pas être utilisée deux fois dans un même mot. Ainsi, nous avons une liste de toutes les cases de la grille qui nous indique grâce à un booléen si une case a déjà été utilisée ou non. Après avoir validé un mot, toutes les cases sont remis à `false`.

Enfin, le mot sera validé s'il comporte au moins 3 lettres et s'il appartient au dictionnaire. Les accents ne sont pas importants par exemple le mot « été » pourra être validé en tapant « ETE ». Pour ce faire, une liste de mots vérifiant toutes ces conditions est initialisée à chaque début de partie et vous gagnez des points si le mot que vous jouez appartient à cette liste.

L'implémentation de cette liste nous a causé quelques difficultés. En effet, nous avons commencé par chercher toutes les combinaisons possibles de lettres dans la grille, pour ensuite vérifier si le mot formé appartenait au dictionnaire, auquel cas, il était ajouté dans la liste. Cette version était correcte mais prenait beaucoup de temps à charger surtout si la grille est plus grande. Pour remédier à ce problème, nous avons parlé avec la responsable de notre projet lors d'une réunion. Elle nous a conseillé de voir le problème à l'envers, c'est-à-dire de chercher dans le dictionnaire quels mots pouvaient être joués sur la grille. Nous avons alors implémenté cette version qui s'avère très efficace.

TIMER

Le chronomètre est un élément central de ce jeu. C'est en fait la limite de temps qui donne son intérêt à la partie, le but étant de trouver le maximum de mots en un temps imparti. C'est donc la fin du chronomètre qui arrête la partie. Nous verrons plus tard qu'il est également utilisé pour les bonus. Nous avons implémenté un timer en version textuelle ainsi qu'en version graphique. Il s'agit en fait d'un décompte d'un nombre x - choisi dans les paramètres - qui décroît de 1 toutes les secondes, jusqu'à 0. Ce décompte commence dès la première apparition de la grille de jeu. Nous avons implémenté cela de sorte à ce que la

version textuelle lance des rappels du temps restant toutes les 30 secondes. La version graphique est une fois de plus, plus immersive : on peut y observer le temps restant défiler, seconde par seconde. Une fois le temps de jeu terminé, la fenêtre de jeu disparaît, laissant place à un affichage de fin de partie, affichant le score obtenu, et proposant de relancer une nouvelle partie, ou de tout simplement quitter le jeu.

DESIGN

Nous nous sommes mit tardivement au design du projet, notre priorité ayant été le fonctionnement technique du jeu. Pour nous, s'y concentrer n'était pas essentiel, nous avons donc au départ juste placé un fond rose sur toutes les pages de l'interface. Toutefois, au fur et à mesure que notre projet se concrétisait, le rendu n'était pas satisfaisant. On avait l'impression que le jeu n'était pas assez « propre » et faisait un peu amateur. On s'est rendu compte que c'était le design très simple qui en était la cause. En effet, cela ne nous donnait pas envie de lancer une partie. On a donc prit un moment pour revoir l'esthétisme du projet. On souhaitait avoir un design à la fois jolie, lié au thème du jeu, et que chaque page ait un rendu différent tout en restant uni. Une page est par exemple recouverte d'éléments liés au jeu Pacman. Après avoir fini les modifications, on a tout de suite ressenti une grande différence. On a conclu qu'aucun élément ne devait être négliger et que même des tâches jugées non essentielles pouvaient avoir un énorme impact. Toutes les images, animations, couleurs que nous avons rajouté ont donné vie à notre jeu.

PARAMETRES

Après que la version classique du jeu ait été implémentée, à savoir une partie d'une durée de 3 min sur une grille de lettres de taille 4 x 4 formant des mots français, nous avons voulu aller plus loin. En effet, pourquoi se limiter à des parties classiques alors que l'on pourrait laisser l'utilisateur les personnaliser? Grâce aux paramètres, l'utilisateur ne sera pas rapidement lassé de jouer et pourra tester de nombreuses combinaisons d'options.

Pour modifier la langue et la durée, rien de plus simple, il a suffi de rajouter d'autres dictionnaires dans nos ressources et de modifier certaines données. Toutefois, la mise en place de la taille de la grille a connu quelques difficultés. En effet, après avoir créé de nouvelles grilles et les avoir ajoutées dans notre code, on s'est rendu compte que la partie ne parvenait plus à se lancer en raison d'un chargement infini. N'en comprenant pas la raison, nous avons cherché une solution pendant plusieurs semaines sans succès. Pendant un moment, on a même envisagé de supprimer cette partie. Toutefois, on a fini par se rendre compte que le problème venait du chargement de la liste des mots possibles de la grille. En effet, comme cela a été expliqué plus haut, au départ on générait une première version de cette liste au lancement d'une partie, ce qui prenait un certain temps pour une grille de taille 4x4. Donc, pour une de taille 5 ou 6, le nombre de mots était considérablement plus élevé, augmentant ainsi le temps de chargement, ce qui a causé le dysfonctionnement du jeu. On a fini par résoudre ce problème, en changeant la manière de récupérer les mots jouables, et d'implémenter les différentes grilles.

BONUS

Pour que le joueur n'abandonne pas la partie en cas de difficultés, nous avons créé un système de bonus.

Il existe cinq bonus, tous ayant un impact différent au cours d'une partie. Le premier, Vision, permet de donner un indice sur un mot pouvant être formé dans la grille. Le deuxième, Inspiration, donne un mot directement. Ces deux bonus ont été implémentés de la même manière: on crée en attribut de la grille une liste de mots jouables non validés et on en pioche un au hasard. Ensuite, on affiche soit le mot soit la première lettre ou sa taille.

Le troisième bonus est Addchrono, il rajoute 30 secondes au timer. Pour l'implémenter, il a simplement suffit d'incrémenter la valeur de la variable contenant le timer.

Le quatrième bonus, Brasier, double le nombre de points obtenus pour les deux prochains mots validés. On a du créer un paramètre booléen dans la fonction qui met à jour le score pour qu'il double les points lorsqu'il est true, et aussi un attribut compteur qui vérifie la double utilisation du bonus.

Le cinquième bonus Rotation échange les places de toutes les lettres de la grille. Lors de son implémentation, on a connu des difficultés mineures: comme on modifiait l'emplacement des lettres de la grille à l'aide de la fonction shuffle, la liste des mots jouables était totalement différente. On a du mettre à jour cette liste puis remettre à jour la valeur des boutons de la grille au niveau Javafx.

Les bonus apportent une aide plus ou moins importante, ils n'ont pas tous la même valeur. Si le joueur pouvait les utiliser à tout moment, cela deviendrait trop facile. Il faut donc maîtriser sa consommation. C'est pour cela que nous avons créé le système de coins.

COINS

Les coins sont la monnaie courante du jeu. Ils permettent au joueur d'acheter des bonus afin de les utiliser au cours d'une partie. Toutefois, pour pouvoir dépenser de l'argent, il faut d'abord en posséder ! Lorsqu'un utilisateur crée un compte, on lui attribue 3 coins au départ. Ensuite, il peut en gagner de différentes manières : en battant l'IA qui se mesure à lui ou en gagnant des trophées.

Grâce à ce système, le joueur ne fait pas que s'amuser sur un jeu pendant quelques minutes, il peut également gérer son argent et l'utiliser à bon escient. Il y a donc une nouvelle façon de jouer qui se débloque : le joueur sera plus concentré lors d'une partie, abordera de nouvelles stratégies afin de gagner plus de coins et sera ainsi plus satisfait à la fin d'une partie : un nouvel enjeu a été créé. Il aura ensuite très envie de rejouer pour pouvoir dépenser cet argent qu'il a durement obtenu. C'est un échange positif entre l'utilisateur et le jeu.

TROPHÉES

Souvent, lorsque quelqu'un télécharge un jeu, il joue quelques heures, finit le mode histoire assez rapidement et passe ensuite à autre chose. Ce n'est pas ce que l'on veut. On souhaite que nos utilisateurs puissent jouer de nombreuses fois sans se lasser. Pour cela, on a implémenté des paramètres, des modes de jeu ainsi que des bonus. Toutefois, rien de mieux sur le long terme que des médailles à débloquer après un événement remarquable.

En effet, nous avons implémenté neuf trophées avec des conditions différentes pour les obtenir. On peut les observer depuis la page d'accueil avec, pour chacun, une apparence unique. Pour en obtenir un on peut par exemple trouver un mot sans voyelle, obtenir plus de 200 points ou encore lancer 100 parties. Il existe même un bonus sans intitulé et uniquement une image pour rajouter du défi au joueur. A chaque gain, on peut récupérer 3 coins.

Avoir la satisfaction d'avoir fini à 100% un jeu est une source de motivation pour que l'utilisateur continue à se connecter régulièrement, d'où l'intérêt de cette fonctionnalité.

USER

Les Users sont - comme leur nom l'indique - les utilisateurs qui jouent où ont joué à notre jeu. De nombreux attributs comme le score, le nombre de coins, ou encore les paramètres de la partie leur sont attribués. Les Users jouent un rôle majeur dans le code et sont en relation directe avec tous les échanges serveur que nous verrons plus tard.

La partie la plus complexe et sans doute celle qui nous a bloqué le plus longtemps est la gestion des « Opponents », soit une liste des adversaires que le joueur affrontera dans le mode en ligne. Nous ne voulions en aucun cas les gérer comme de simples utilisateurs, par un soucis de potentielle sécurité ! Il fallait donc trouver une alternative. C'est là que nous est venu l'idée des « Opponents » : ce sont des Users avec juste un pseudonyme, une liste de leur scores récents ainsi que des paramètres de partie. Chaque User possède donc une liste d'Opponents contenant tous les adversaires que l'User affronte ! Cela nous permettra de les afficher plus facilement dans l'onglet des parties en cours du mode en ligne que nous aborderons dans la dernière partie du rapport.

SAUVEGARDE

Pour pouvoir différencier les Users, il faut qu'ils se distinguent. Pour cela, il a fallu implémenter un système de sauvegarde. Pour se connecter, l'utilisateur aura besoin de renseigner un e-mail, un pseudonyme ainsi qu'un mot de passe. Ces données seront stockées et utilisées pour pouvoir leur attribuer un compte qui leur sera propre. Une page de statistiques a spécialement été créée afin d'afficher toutes les informations sauvegardées du joueur concernant le jeu comme par exemple son meilleur score, ses dix derniers scores, sa moyenne ou encore le nombre de parties jouées au total. Cette donnée sera d'ailleurs utile pour actualiser le nombre de trophées gagnés puisque l'on a besoin de son historique de parties pour pouvoir vérifier si il a réuni les conditions requises pour en débloquer une. Une sauvegarde est donc présente pour les trophées et les paramètres.

Nous souhaitons également ajouter une fonctionnalité pour pouvoir reprendre une partie en cours en sauvegardant le chrono, la grille de jeu ainsi que les mots qui ont déjà été joués. Toutefois, même si cela a été réalisé sur une branche annexe de notre git, par manque de temps et en raison des nombreux soucis de conflits entre les branches, nous ne sommes pas parvenus à l'implémenter dans la version finale.

MODE DE JEU

MODE SOLO

Le premier mode de jeu conçu a été le mode solo. Ce mode est la fonctionnalité principale de notre jeu. Il représente la base de notre projet. Pendant un temps donné, une grille de lettres est à notre disposition et le joueur doit tour à tour proposer des mots pouvant être formé à partir de cette grille et appartenant au dictionnaire de la langue choisi au préalable. Le jeu se chargera de lui indiquer l'issue de ses tentatives à travers des emojis joyeux et triste, et le score sera augmenté ou inchangé. Les bonus sont également présents pour aider le joueur bloqué et les coins sont décrémentés au fur et à mesure de leur utilisation. A la fin de la partie, le score du joueur est affiché ainsi que la liste de ses mots validés, celle de tous les mots jouables et celle de l'IA - que nous aborderons juste après.

Toutes les fonctionnalités de ce mode ont été expliquées lors de ce rapport. Sa réalisation a été notre première priorité puisqu'il rassemble pratiquement toutes les fonctionnalités obligatoires demandées dans la réalisation de notre projet. Toutefois, cela reste un mode assez simple sans réel enjeu. Nous avons donc voulu aller plus loin.

IA

Pour rendre le mode solo plus intéressant, nous avons implémenté une intelligence artificielle nommé le Master qui - en fonction du niveau et du temps de jeu choisi par l'utilisateur - va tirer au sort un nombre de mots parmi la liste des mots jouables de la grille. Plus le temps et le niveau seront grand, plus l'IA sera « forte » et obtiendra plus de mots et donc plus de points. Si le joueur parvient à obtenir un score plus élevé que celui du Master, il obtiendra entre 1 et 3 coins. Cela incitera le joueur à faire plus de parties de difficultés variés afin d'obtenir des coins plus rapidement.

Ce mode a été conçu afin de stimuler l'utilisateur qui pourrait s'ennuyer de ne pas pouvoir comparer ses scores à ceux d'autres joueurs. Grâce à l'IA il pourra tester son niveau directement à la fin de la partie et alimenter son désir d'améliorer son score lors de futures parties.

MULTIJOUEUR LOCALE

Afin d'ajouter plus de fonctionnalités à notre jeu, nous avons aussi proposé un nouveau mode de jeu : le multijoueur local. Ce mode permet à l'utilisateur de profiter du jeu simultanément avec un autre joueur et ce, localement, c'est à dire sur la même machine. Pour cela, nous avons décidé de créer une partie où les deux joueurs s'affrontent sur la même grille, chacun leur

tour. Le joueur ayant trouvé le plus de mots remporte la partie. Ce mode est donc basé sur le code de la version solo, avec comme ajout, le changement de joueur pendant la partie, ainsi que l'apparition d'un deuxième score pour le joueur adverse.

Avant de commencer la partie, ce dernier renseigne son pseudonyme. La partie commence alors; chaque joueur dispose de 15 secondes pour trouver le plus de mots possibles. Son score monte alors en fonction de ses propositions. A la fin du temps accordé au joueur, une fenêtre apparaît signalant au joueur qu'il doit laisser la place à son adversaire. Pour éviter la tricherie entre les joueurs, la fenêtre de jeu devient inaccessible le temps de changer de joueur, et la fenêtre de changement ne peut être contournée. Une fois le deuxième joueur en place, le jeu reprend, augmentant cette fois l'autre score. Ces changements de joueurs ont lieu jusqu'à la fin du chronomètre. Une fois l'échéance arrivée, une fenêtre donnant le nom du joueur ayant le score le plus élevé apparaît, et sur le même principe que le mode solo, les joueurs peuvent relancer une partie ou arrêter le jeu.

MULTIJOUEUR EN LIGNE

Le multijoueur en ligne est le dernier mode à avoir été implémenté dans le jeu. Il est basé sur un RESTful Web Service avec Spring. Notre projet utilise un système serveur/client, qui peuvent communiquer entre eux à partir de n'importe quelle adresse IP sur lequel le serveur peut être lancé. Avec plus de moyens, notre jeu pourrait potentiellement être accessible à tout le monde si on louait un serveur ! Le principe est assez simple, le serveur est constitué des sauvegardes de l'utilisateur et gère les différents appel que le client pourrait envoyer. Le client quand à lui envoie des requêtes au serveur. Ces requêtes peuvent être une demande d'objet. Le client attend donc un objet en retour, ou peut en envoyer un au serveur. Ces différentes requêtes permettent donc à ce que tous les utilisateurs connectés sur le même serveur utilisent la même sauvegarde.

Lorsque l'on clique sur l'option ONLINE du jeu, une fenêtre s'ouvre et nous demande d'entrer le nom de l'utilisateur que l'on souhaite affronter. Le programme vérifiera que ce joueur existe bien. On pourra ensuite choisir les paramètres voulu et lancer une partie. Une fois terminé, le mode annonce qu'il faudra attendre que le joueur adverse fasse également une partie, sur cette même grille, pour pouvoir continuer. Un mail est envoyé au joueur en question pour le prévenir du défi lancé et du score obtenu. Il n'aura plus qu'à se connecter à son tour et à choisir le mode ONLINE pour pouvoir voir l'invitation et y répondre. Les parties multijoueurs sont infini : les joueurs peuvent se défier autant de fois qu'ils le souhaitent.

EXTENSIONS POSSIBLE

Pour de futurs mise à jour du jeu, on a pensé à créer d'autres modes de jeu. Par exemple, créer un mode où on choisit une lettre au hasard, et durant la partie tous les mots validés doivent contenir la lettre en question. On pense également à rajouter la possibilité de reprendre une partie en cours de route.

CONCLUSION

Tout le long du semestre, on s'est demandé si l'on était capable de produire un projet aussi complet en si peu de temps. Nous avons eu des hauts et des bas dans la cohésion de notre groupe, des moments de sur-confiance puis de doutes, mais aujourd'hui, on peut affirmer avec fierté que l'on a réussi à développer ce jeu par nos propres moyens. Ce projet nous a redonné confiance en nous et nous a rassuré sur nos capacités et notre volonté à avancer et progresser dans nos études.

Grâce à ce projet, nous nous sommes également nettement améliorés dans le travail en équipe, le respects des deadlines, la planification de prochaines tâches, la modélisation d'un projet : nous avons beaucoup appris. Faire ce projet nous a été hautement bénéfique et nous a donné envie de développer à notre tour des jeux plus complexes et plus uniques en son genre !

Répartition des tâches

Fafa:

Sauvegarde User (statistiques)

IA

Méthode de recherche pour récupérer la liste des mots jouables de la grille

Affichage à la fin d'une partie des mots jouables/joués

Lorraine:

Multijoueur locale

Timer

Replay d'une partie

Grille (taux d'apparitions des lettres etc.)

Samuel:

User (création de compte etc.)

Opponents

Gradle

Serveur

Multijoueur online

Aranniy:

Coins

Trophées

Paramètres (taille, langue etc.)

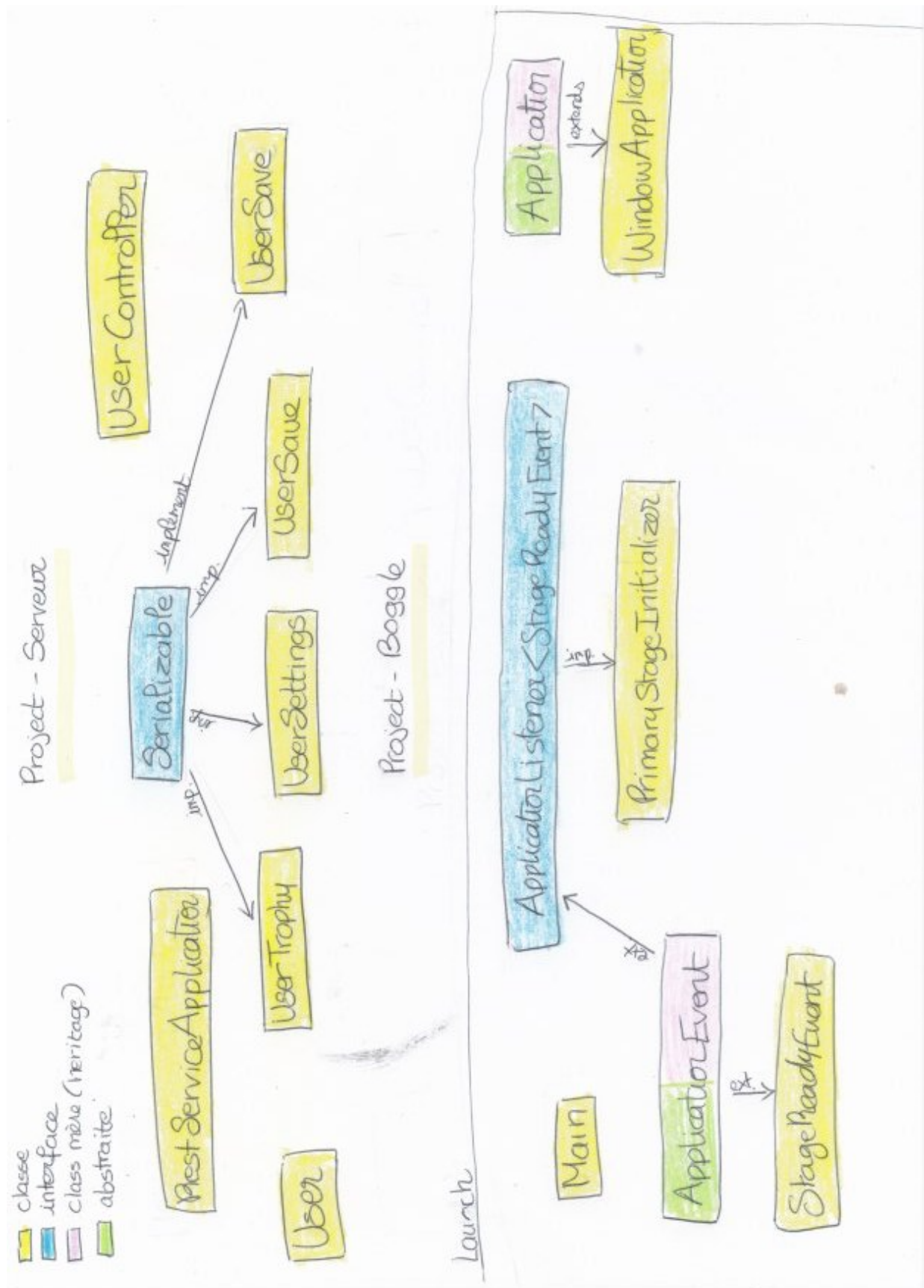
Design

Bonus

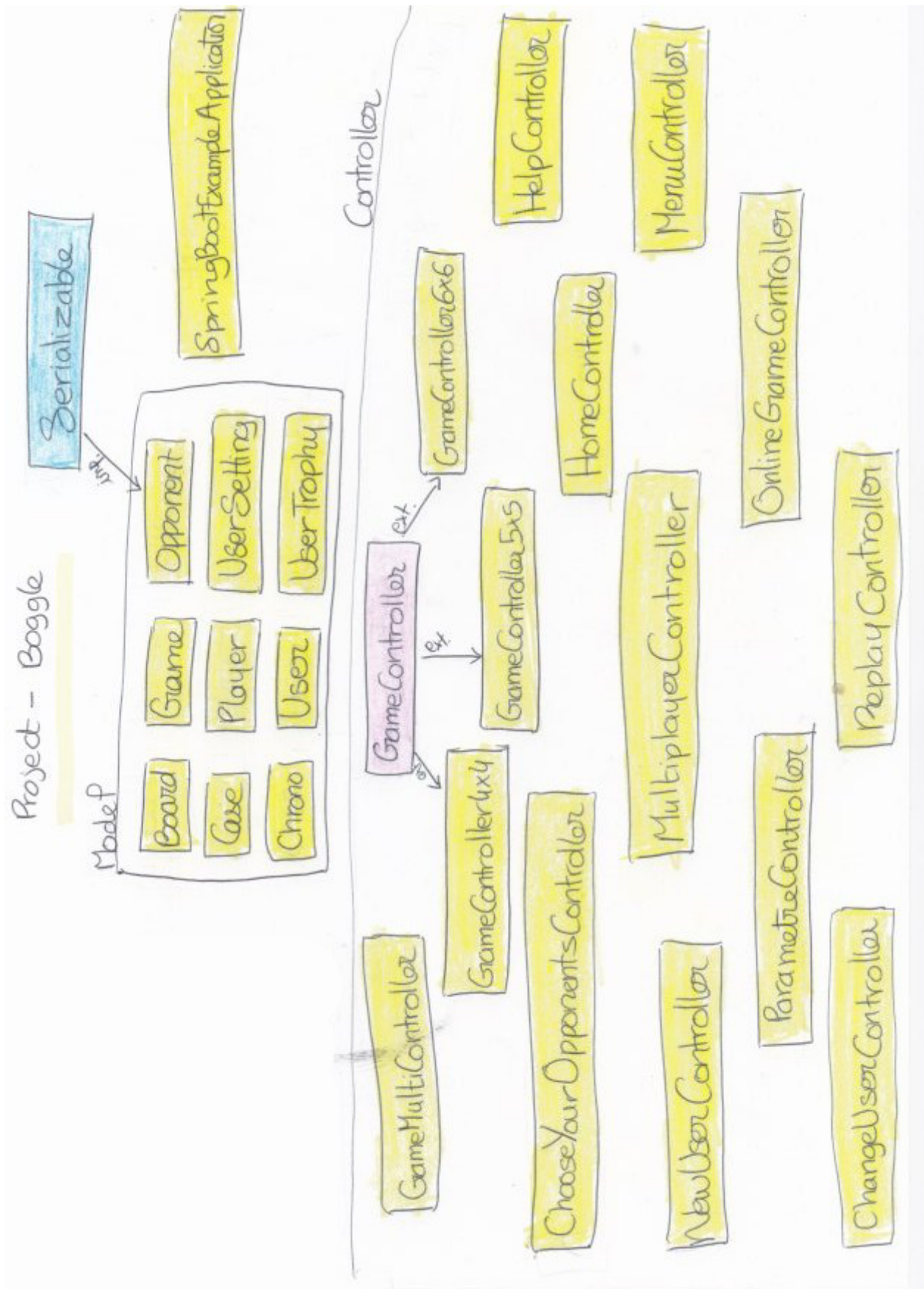
Tous :

Implémentation des règles

DIAGRAMME



DIAGRAMME



BIBLIOGRAPHIE

Liens des png utilisés :

Fond : <https://www.pinterest.fr/pin/443393525819183345/>

GG bleu : <http://pixelartmaker.com/art/de49826ec5e9dff>

Trophy : https://www.pngitem.com/middle/xToTbh_trophy-soccer-pixel-pixelart-trophy-pixel-art-hd/

Feu : https://www.clipartmax.com/middle/m2i8N4N4N4G6K9N4_pixel-art-small-pixel-art-fire/

Ghost : https://www.kindpng.com/imgv/JhwRJh_cute-ghost-png-cute-ghost-pixel-art-transparent/

Kirby : https://www.pngitem.com/middle/miTJho_transparent-cool-png-images-kirby-with-sunglasses-png/

You Lose : https://www.kindpng.com/imgv/hThibRi_you-lose-png-transparent-png/

Heart red : <https://www.teepublic.com/fr/art-mural/pixel-heart>

Heart orange : <http://pixelartmaker.com/art/51f64a437107e68>

Heart green : https://www.pngfind.com/mpng/hRhxmob_8-bit-heart-green-tata-bt21-pixel-art/

Multi ghost : https://www.seekpng.com/ipng/u2q8a9o0e6u2y3u2_pac-man-ghost-png-photos-pacman-and-ghosts/

Ghost orange : https://www.pnclipart.com/pindetail/ibhmwhJ_pac-man-png-pacman-ghost-transparent-background-clipart/

Cat angel : <https://www.pinterest.fr/pin/535928424392637593/>

Cat demon : <http://pixelartmaker.com/art/ce5924096ba8505>

Help : <https://www.deviantart.com/bumibunny/art/Help-Potion-Pixel-Animation-839970299>

Cherry : <http://pixelartmaker.com/art/e067ae2e3700165>

Pacman : <https://gifer.com/en/XOsf>

Fonctions :

Fréquence lettres : https://fr.wikipedia.org/wiki/Fr%C3%A9quence_d%27apparition_des_lettres_en_fran%C3%A7ais

Enlever accents : <https://www.developpez.net/forums/d460531/java/general-java/langage/supprimer-accent-chaine/>