

Rapport final

Projet Analyse des besoins et Application Web

par
Aryamaan KUNWAR
Axel MAÏSSA

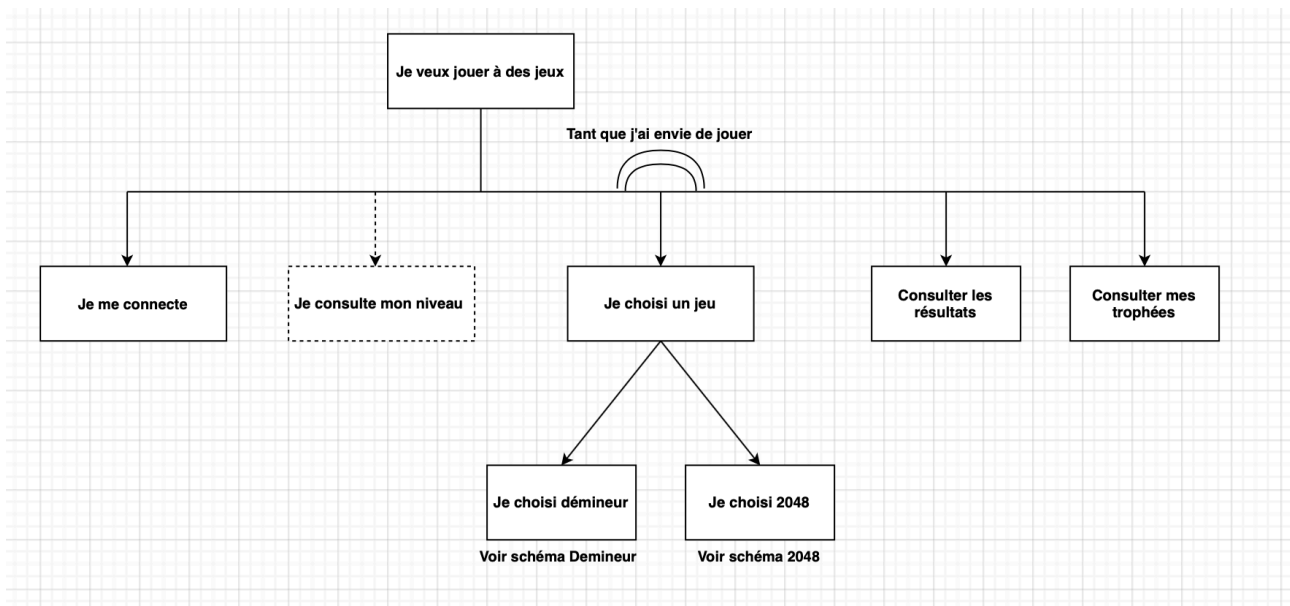
Monsieur M. BUFFA - Monsieur P. RENEVIER - Madame M. TIKAT

Mars 2021

Modèle des tâches	2
Modèle Général	2
Modèle 2048	2
Modèle Démineur	3
Maquettes	4
Architecture logicielle	5
Barre de navigation	5
Gestion d'un clic sur le canvas du Démineur	5
Gestion des tours dans 2048	7
Envoi des données au profil, trophées et localStorage	7
Planning des réalisations des scénarios	9
Techniques utilisées	10
Langages	10
Techniques organisationnelles	10
Comparaison	12

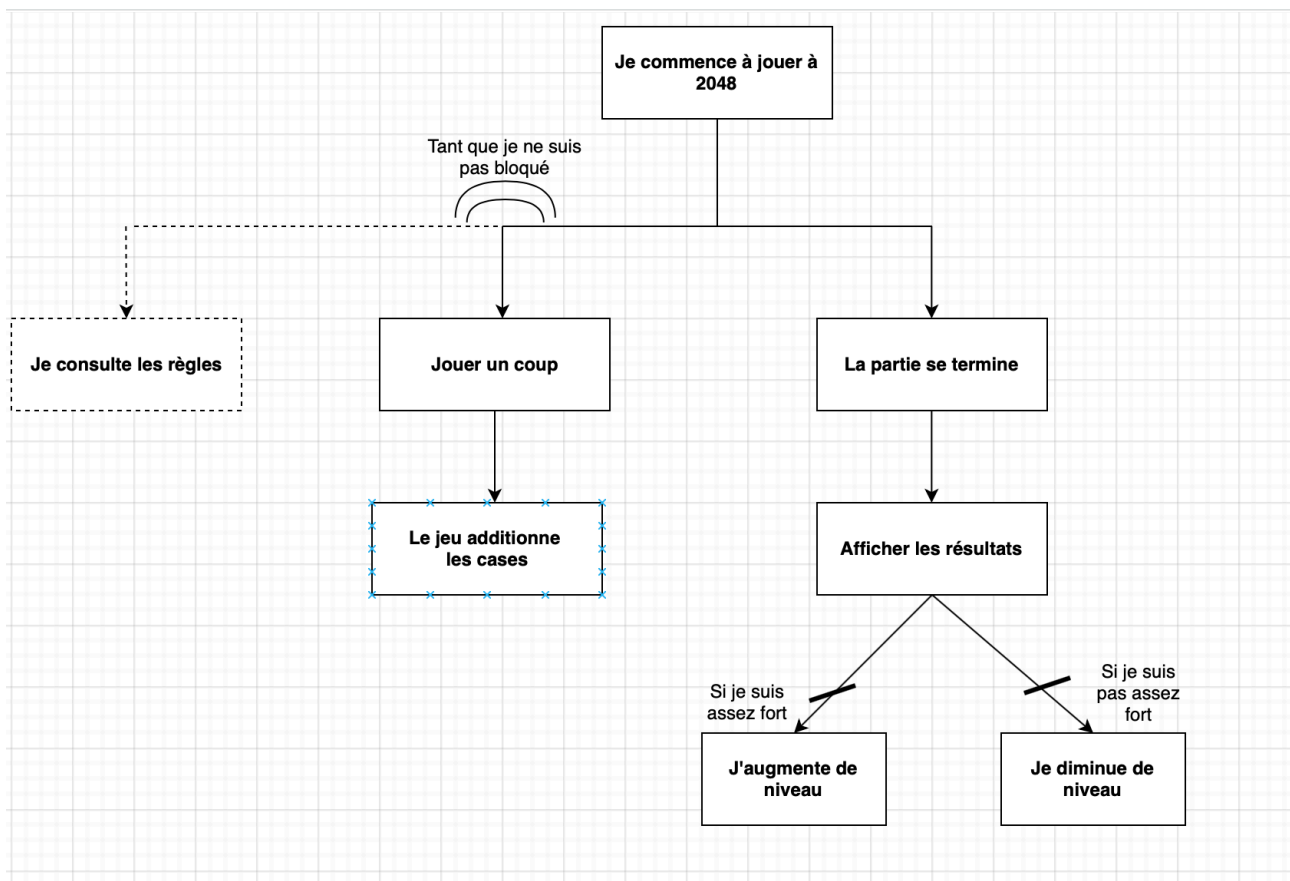
Modèle des tâches

Modèle Général



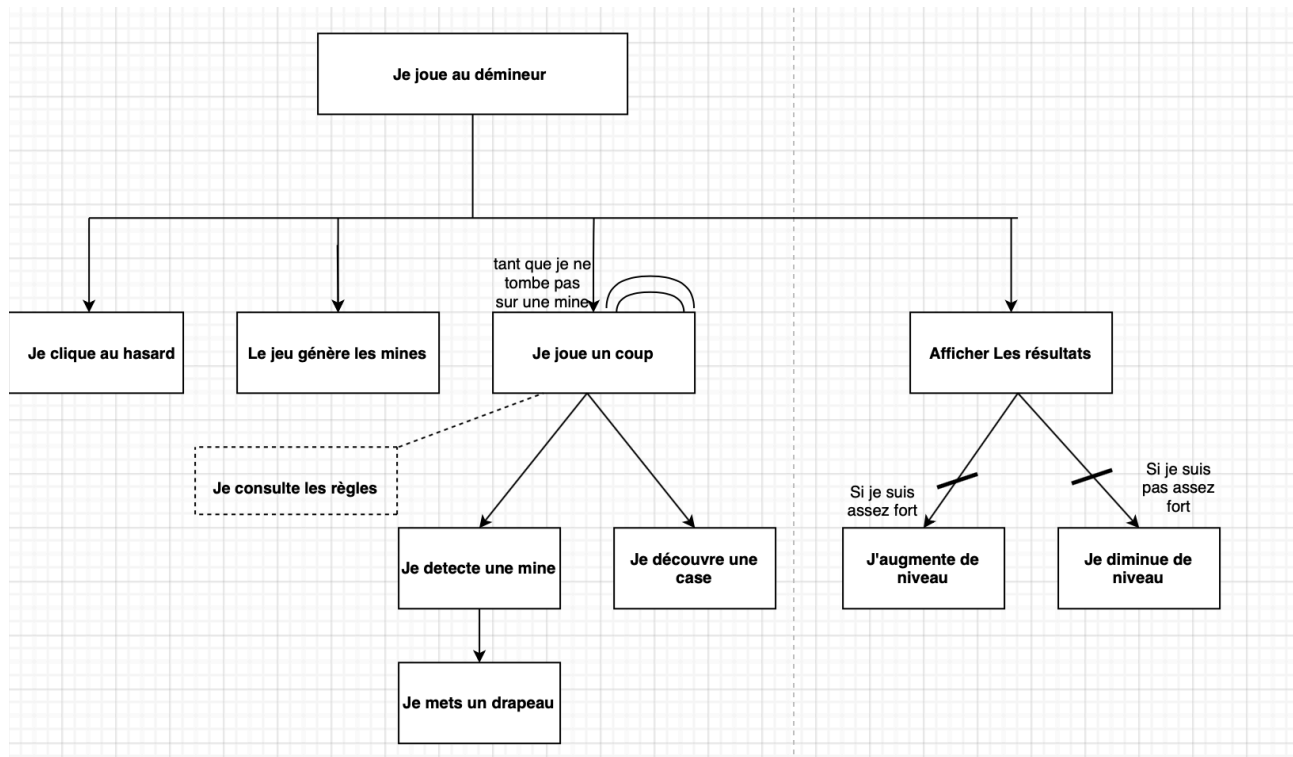
L'utilisateur doit se connecter avec un pseudo, il peut consulter son profil s'il a déjà joué ou choisir un jeu. Il joue tant qu'il en a envie et peut passer du 2048 au Démineur.

Modèle 2048



Je commence une partie, je peux me déplacer avec mes flèches seulement si le coup est autorisé, des tuiles s'additionnent et la partie se termine uniquement lorsque je suis bloqué. La victoire est déclenchée lorsque j'obtiens une tuile 2048.

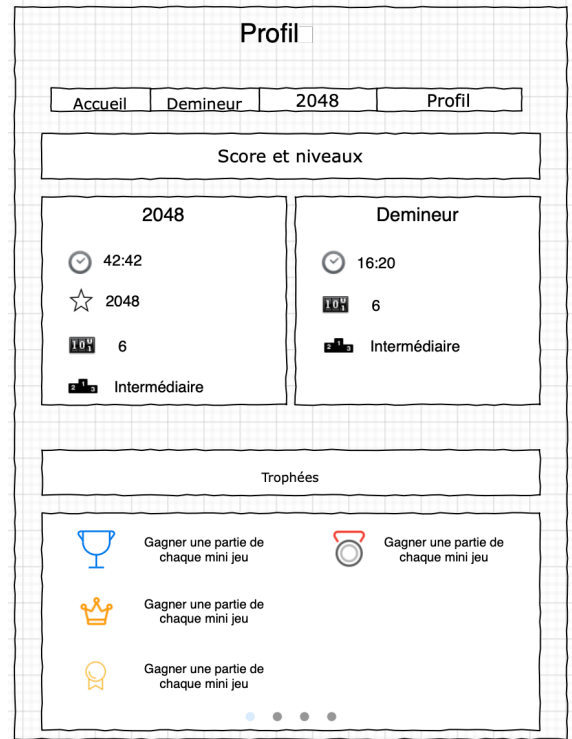
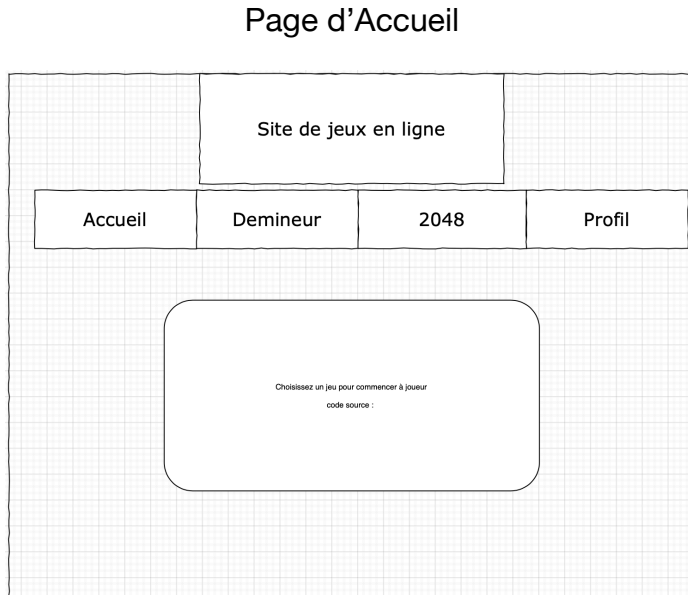
Modèle Démineur



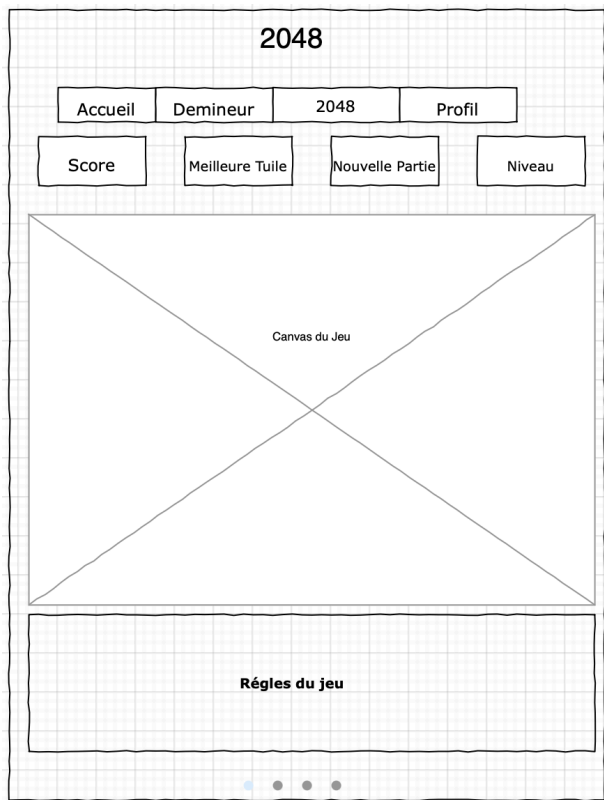
Je commence une partie de démineur en cliquant sur l'une des cases, les mines sont alors placées aléatoirement après ce premier clic afin de ne pas faire perdre l'utilisateur. Je peux dévoiler des cases en cliquant dessus, tant que celles-ci ne sont pas des mines. Je peux positionner un drapeau sur une mine. La partie se termine lorsque j'ai découvert toutes les cases qui ne sont pas des mines. Je peux alors consulter mon profil.

Maquettes

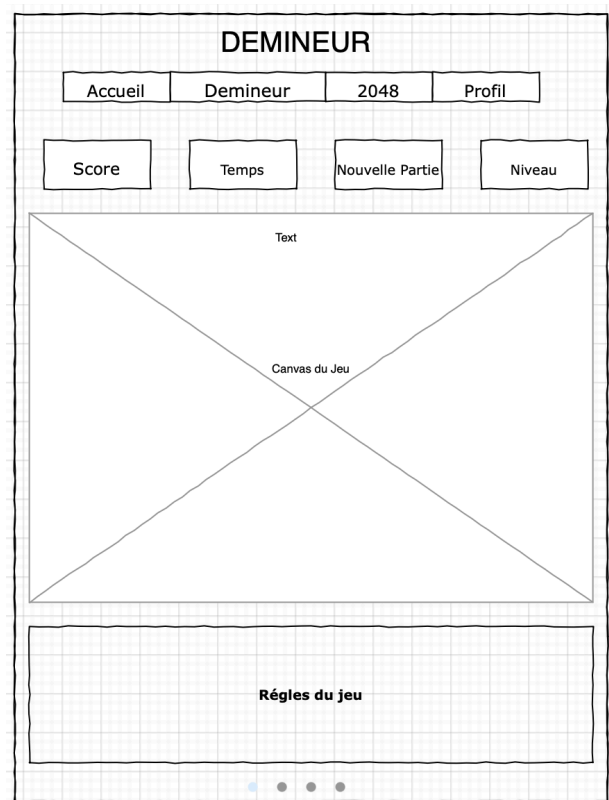
Page Profil



Page 2048



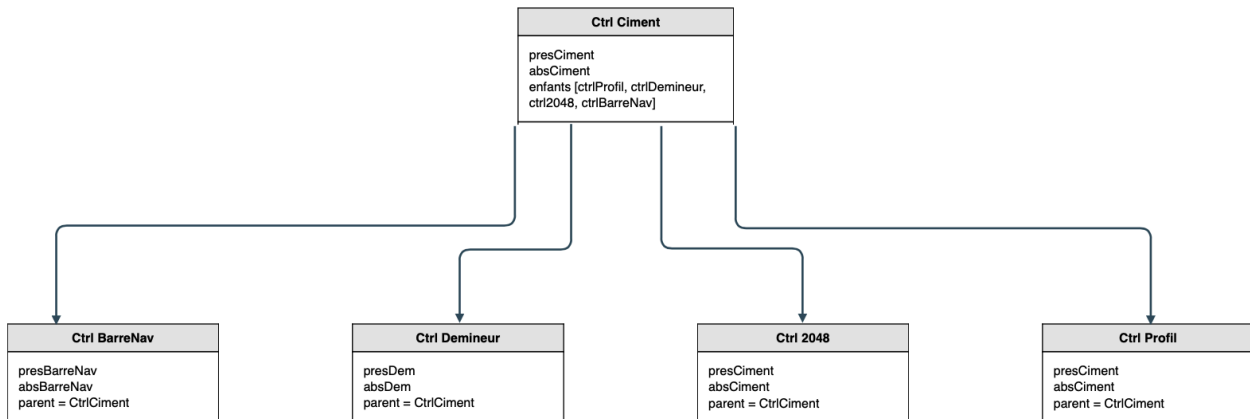
Page Démineur



Architecture logicielle

Pour l'architecture nous avons choisi d'utiliser les principes PAC afin de séparer la logique et la presentation.

Diagramme de classes



Ce diagramme permet de comprendre le raisonnement derrière la découpe des classes. Le contrôleur de Ciment qui contient sa présentation et son abstraction contient également des **enfants qui sont les contrôleurs des autres classes**. Cette découpe permet de **réglementer la communication** entre classes enfants qui doit obligatoirement passer par la class parent.

Barre de navigation

Celle ci, créée lors du changement de page, permet de naviguer entre les pages du site. C'est elle qui s'occupe de **vider le container** où la page a été générée dynamiquement. De plus, elle envoie le **message d'initialisation** en passant par ciment au contrôleur de la page à afficher. Elle s'occupe d'enlever les events listener.

Gestion d'un clic sur le canvas du Demeur

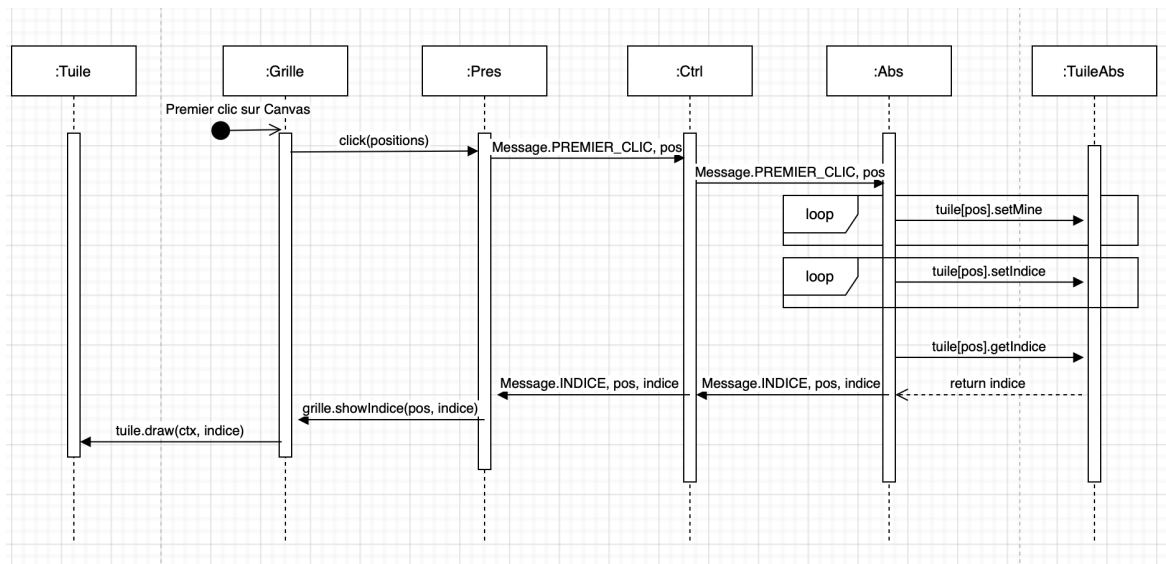
Appelé lors de la réception du message "init" envoyé par le parent (Ciment), la présentation génère le HTML et donne le choix à l'utilisateur entre trois niveaux. Une fois ce choix fait la présentation dessine la grille et ajoute les listener sur le canvas.

Le diagramme en dessous montre les séquences qui ont lieu après le premier clic. La classe Grille qui fait partie de la présentation appelle une méthode de la présentation qui gère les clics, celle-ci envoie un message au contrôleur qui le transmet à l'abstraction. Dans l'abstraction un tableau 2D contenant des TuileAbs représente l'équivalent de la grille dans la présentation.

Lors de la réception de ce message l'**abstraction génère au hasard le nombre de mines en fonction du niveau** et génère ensuite les indices des autres tuiles et les stocke dans TuileAbs.

Une fois les mines et les indices générés, l'abstraction envoie un message au contrôleur avec l'indice à dévoiler, celui-ci le transmet à la présentation qui la fait apparaître sur le canvas.

Premier Clic sur le canvas

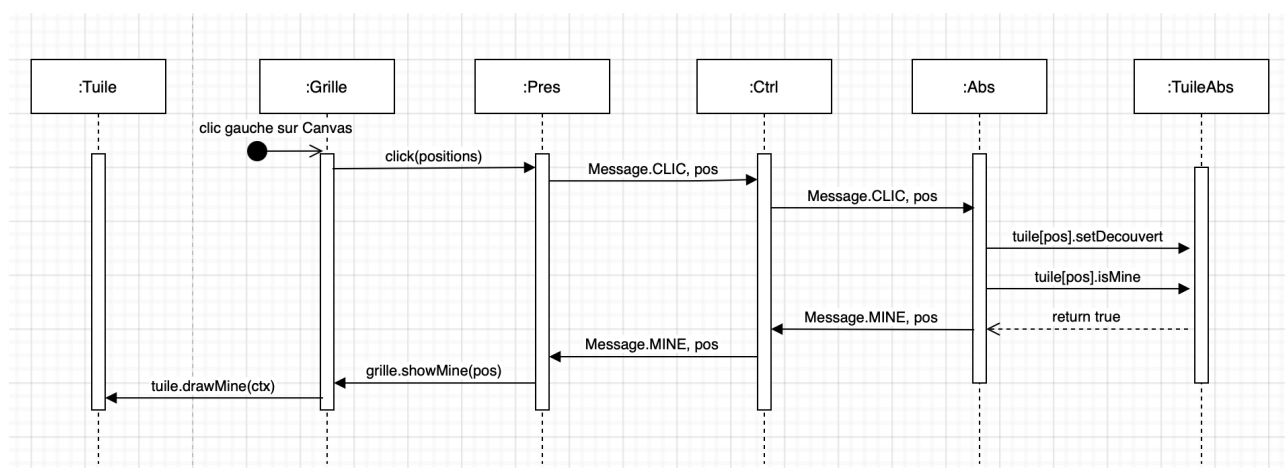


Quand l'abstraction reçoit le message CLIC ou PREMIER_CLIC, la méthode tuileClicked(pos) est appelée, elle trouve la TuileAbs dans le tableau2D et détermine ce qu'il faut faire en fonction de la tuile.

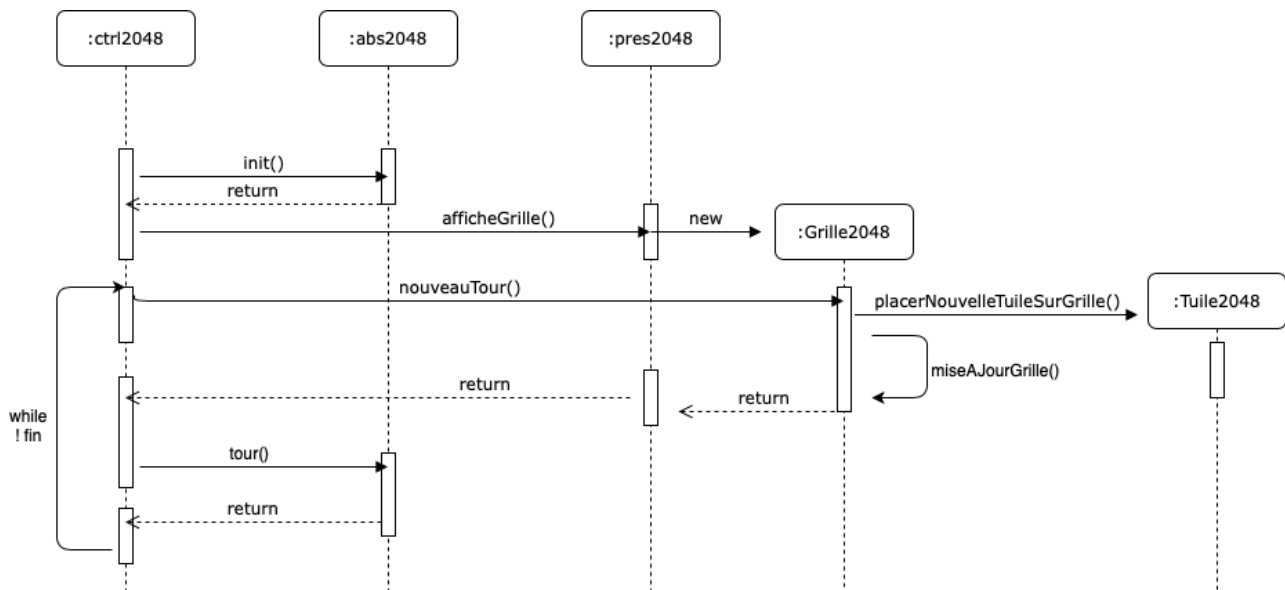
Un clic sur une tuile qui n'est pas une mine produit les mêmes séquences que PREMIER_CLIC sauf la génération des mines et des indices dans l'abstraction. Un clic sur une mine produit les séquences démontré dans le diagramme en dessous.

Mais ce qui n'est pas montré sur le diagramme est que lorsque TuileClicked () détecte que la tuile cliquée est une mine, une méthode s'appelant AfficheTout() est appelée. Celle-ci parcourt le tableau 2D de TuileAbs et appelle TuileClicked () sur chaque tuile, cette méthode utilise donc la récursion afin de dévoiler toute la grille.

Clic sur une mine



Gestion des tours dans 2048



Dans la partie haute de ce diagramme, on peut voir l'initialisation d'une partie de 2048 : le contrôleur 2048 reçoit un message d'initialisation qu'il envoie à son abstraction puis son abstraction communique avec la présentation en passant par le contrôleur. La présentation crée un objet **Grille2048**. Deux tuiles sont créées par la Grille2048, le HTML de la grille est **mis à jour**, alors la partie peut commencer.

Les handlers sur les flèches permettent de déclencher un tour, si celui-ci est autorisé. Le nouveau tour entraîne le **placement d'une nouvelle tuile** sur la grille ainsi que le **calcul des fusions** des tuiles. À la fin du tour, la présentation récupère les informations importantes du tour (score, tuile la plus grande, ...) et les envoie au contrôleur qui les transmet à la présentation. La présentation décide si une victoire est à générer ou non et s'occupe de communiquer les données de la partie au profil.

Envoi des données au profil, trophées et localStorage

À chaque tour de 2048 ou à chaque fin de partie de démineur, **les jeux communiquent avec le profil**. Pour se faire leur contrôleur respectif envoie un message avec une **pièce jointe** à leur parent, Ciment, qui transmet alors au contrôleur de Profil.

Donnée et Timer

Lorsqu'une partie est gagnée / perdue, pour le démineur, ou gagnée, pour le 2048, un message est envoyé à Profil. Ce message transit par Ciment qui gère le timer. Lorsque Ciment voit ce message de victoire/défaite passer dans son contrôleur, il **enrichit la pièce jointe du Timer** qui se trouve dans son présentation. Le tout est réceptionné par le profil qui compare l'ancienne "meilleure" valeur à celle reçue et si celle-ci est jugée "meilleure", alors elle remplace l'ancienne. Ex : si on a un score de 3000 au 2048 et que l'ancien meilleur score est 2500, 3000 devient le nouveau meilleur score et est entré en localStorage.

Trophée

Une fois dans le profil, les données sont analysées pour savoir si elle donne droit à l'obtention d'un trophée (si celui ci n'a pas déjà été distribué). Si c'est le cas, un **pop-up** s'affiche sur le jeu (pendant la partie, pour un 2048, ou en fin de partie pour un démineur) et le trophée est push dans une liste de trophée qui est elle même mise en localStorage.

Planning des réalisations des scénarios

Semaine 1

User story : Je peux naviguer dans le site, choisir mon niveau (**démineur**), me rendre sur le **Démineur** et le **2048**

- Faire squelettes des pages (4)
- Mise en place du template de PAC
- Faire la grille des deux jeux et les remplir
- Choix de la difficulté au **démineur**, génération de la grille + nombre de mines en conséquence

*

Semaine 2

User story : Je gagne le **démineur** après mon premier click et je peux bouger des cases avec mes flèches

- **Démineur** :
 - * Gérer le premier click sur la grille
 - * Générer l'apparition des mines en fonction de ce premier click
 - * Découvrir les cases après ce click
- **2048** :
 - * Afficher aléatoirement les premiers chiffres
 - * Déplacement des cases avec les flèches

*

Semaine 3

User story : Je perds une partie lorsque je clique sur une mine et je gagne **2048** lorsque je fusionne deux cases

- **Démineur** :
 - * Générer les indices
 - * Fonction pour gérer les clicks qui découvrent une case
- **2048** :
 - * Fonction pour additionner deux cases
 - * Fusion entre deux cases
 - * Ajout de nouvelles cases à chaque coups

*

Semaine 4

User story : Je peux mettre des drapeaux dans mon **démineur**, pouvoir consulter mon temps de jeu ainsi que mon score

- **Démineur** :
 - * Affichage de toutes les mines lorsque je perds
 - * Possibilité de mettre des drapeaux avec click droit
 - * Gérer la victoire
 - * Ajouter un écran de victoire
- **2048** :
 - * Stocker le score
 - * Gérer la victoire (Avoir une case **2048**) mais laisser la possibilité de continuer
 - * Pop up pour la victoire et augmentation de niveau
- Gérer les scores, temps de jeu, trophées avec une classe mère
- Créer ou récupérer des assets pour embellir le jeu

*

Semaine 5

User story : Je peux consulter mes hauts faits

- L'utilisateur peut gagner des trophées
- Gérer l'apparition des trophées pendant le jeu (affichage pop up dans le coin)
- Ajouter les trophées et statistiques à l'écran de victoire

Techniques utilisées

Langages

Le squelette est en HTML.

1.3 % de notre code se trouve être du HTML

6% du CSS

92.7% JavaScript

Le but été d'avoir une seule page HTML et de créer un **Single Page Application** afin de profiter d'une expérience utilisateur optimale, puisque sans latence et en cas de perte de connexion, l'user peut toujours naviguer entre les pages puisqu'il dispose des scripts.

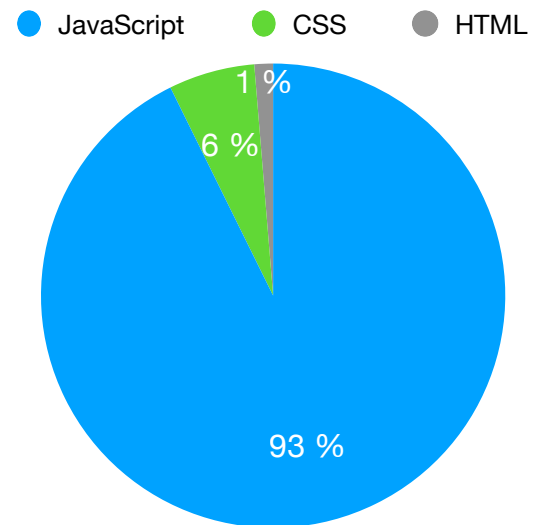
Le **JavaScript** nous permet alors de **générer dynamiquement** les éléments DOM dont nous avons besoin.

Nous avons utilisé une fonctionnalité intéressante de JS : **localStorage**, sorte de mémoire qui agit de la même façon que les cookies mais dont la capacité est supérieure. Nous l'avons donc utilisée afin de stocker des objets. Il a fallu convertir en JSON pour le stockage et la récupération demande une conversion supplémentaire puisque l'objet récupéré depuis le JSON doit être retransformer en l'objet qu'il était avant (Trophée).

En ce qui concerne le **CSS**, nous avons fait du mieux que nous pouvions. Nous sommes conscient du fait que notre CSS présente beaucoup de propriétés qui se chevauchent et donc beaucoup de lignes inutiles. Le « maquillage » étant la partie la plus importante pour l'utilisateur mais la moins intéressante à développer.

Nous avons tout de même réussi à créer des petites animations avec keyframe pour l'apparition des nouvelles tuiles du 2048, des sélecteurs évolués, ...

Deux techniques différentes ont été abordées pour la création des jeux : **DOM** pour le 2048 et **Canvas** pour le démineur.



Techniques organisationnelles

Nous avons procédé à un découpage en itération avec une ou plusieurs users stories et des issues reliées afin de se répartir le travail.

Chaque jeu a été développé par une personne. La structure générale PAC ainsi que le Profil, l'accueil ou la mécanique de trophée/statistique a été développée à deux.

Aperçu des issues d'une itération

Itération 4

Closed 14 days ago 100% complete

0 Open ✓ 18 Closed

Barre des stats lors d'une partie de 2048 ou Démineur task

#50 by aryamaan3 was closed 14 days ago

Je veux avoir des informations pendant et après ma partie story

#46 by aryamaan3 was closed 14 days ago

[Démineur] Enlever un drapeau ne devrait pas impacter Mines Restants bug

#63 by aryamaan3 was closed 14 days ago

[Démineur] Mines restant subtask

#55 by aryamaan3 was closed 14 days ago

[2048] Je veux connaître mon score, savoir si j'ai gagné mais avoir la possibilité de continuer story

#56 by aryamaan3 was closed 14 days ago

[2048] Gestion de la victoire task

#59 by aryamaan3 was closed 14 days ago

Envoyer les informations lors d'une fin de partie au Profil task

#61 by M-axel was closed 14 days ago

Mettre en place le stockage localStorage pour avoir des données persistantes subtask

#62 by M-axel was closed 14 days ago

Formater l'affichage du timer subtask

#60 by M-axel was closed 17 days ago

[2048] Correction des bugs bug

#58 by aryamaan3 was closed 19 days ago

Affichage du temps de jeu subtask

#54 by aryamaan3 was closed 19 days ago

[2048] Affichage du score actuel subtask

#57 by aryamaan3 was closed 19 days ago

Subtask

Formater l'affichage du timer #60

Closed M-axel opened this issue 19 days ago · 1 comment

M-axel commented 19 days ago

Task #50

M-axel added the subtask label 19 days ago

M-axel added this to the Itération 4 milestone 19 days ago

M-axel self-assigned this 19 days ago

M-axel commented 19 days ago

Author

Avoir quelque chose style "01:30".

M-axel added a commit that referenced this issue 17 days ago

Formatage du timer grâce à une fonction dans Ciment.js : formatTimer() ...

7e4e814

M-axel closed this 17 days ago

Comparaison

Page de connexion

Nous n'avons pas mis en place de page de connexion car nous avons préféré l'utilisation de local storage afin de stocker les données de l'utilisateur sur son navigateur. La page de connexion rajoute des étapes excessives sans tant d'apport.

Single Page Application

On a décidé de faire une single page application avec l'html qui se génère dynamiquement plutôt que 4 pages html pour les raisons évoqués dans la partie "techniques utilisés"

Niveaux 2048

On s'est rendu compte que des niveaux dans 2048 n'était pas pertinent car la seule façon de endurcir/faciliter le jeu est de changer la valeur de la tuile qui apparait après chaque mouvement.

Augmentation/baisse de niveau automatique

Nous avons voulu que ce soit le site qui choisi le niveau du joueur en fonction de ses parties antérieurs, mais nous l'avons pas implémentée car 2048 n'a pas de niveaux. De ce fait nous laissons le choix à l'utilisateur concernant le niveau de 2048.

Règles

Non pertinent car les jeux sont connu et assez simple à comprendre.

Suppression des titres

Cela embellit le site et la barre de navigation grise la case du page où l'on se situe.

Score en fonction du niveau de démineur

Les statistiques dans profil contenant démineur ne se différencie pas en fonction du niveau, donc forcément le meilleur temps sera celui du niveau débutant. Nous l'avons pas mis en place à cause d'une mauvaise gestion du temps lors de la dernière itération où l'on a rencontré des bugs dans les deux jeux.