# Introduction:

Dans le cadre de notre deuxième année au sein de l’école d’ingénieur IMAC, nous devons mettre nos connaissances en jeu dans la réalisation d’un projet logiciel programmé en Java et utilisant Swing comme gestionnaire d’interface.

Le projet qui nous a été confié consistait à développer jeu se basant sur le mondialement connu Tetris, mais en ajoutant une nouvelle dimension : une recherche de mots à partir des lettres contenues dans les cases des formes du Tetris, afin de réaliser le meilleur score possible.

Ce rapport présente donc la conception du jeu ainsi que les principaux axes du développement de ce jeu.

Contents

[Introduction: 1](#_Toc388283075)

[Présentation 3](#_Toc388283076)

[Règles du jeu 3](#_Toc388283077)

[Thème 4](#_Toc388283078)

[Visuel 4](#_Toc388283079)

[Le logo du jeu 4](#_Toc388283080)

[Les boutons 5](#_Toc388283081)

[Le visuel d’input 5](#_Toc388283082)

[Le Portrait 6](#_Toc388283083)

[La grid 7](#_Toc388283084)

[Les briques 8](#_Toc388283085)

[Contrôles 8](#_Toc388283086)

[Conception 8](#_Toc388283087)

[La création des briques de jeu 8](#_Toc388283088)

[Recherche dans le dictionnaire 8](#_Toc388283089)

[Le passage des données à l’interface 8](#_Toc388283090)

[Problèmes et solutions 9](#_Toc388283091)

[Gestions des inputs 9](#_Toc388283092)

[La Gravité 9](#_Toc388283093)

[La gestion des collisions des briques 9](#_Toc388283094)

[Conclusion 9](#_Toc388283095)

# Présentation

## Règles du jeu

Il s’agit de réaliser un score maximal avant de perdre. Les règles de bases sont similaires à celles de Tetris. Cependant, afin de supprimer une ligne ou plusieurs cases précédemment tombées, il faut réaliser un mot via les modes Worddle ou Anagram.

Le mode Worddle est initié via la touche ‘w’. Il permet de choisir une lettre dans la grille, en partant de cette première sélection, on choisit une autre case adjacente contenant elle aussi une lettre. De proche en proche on réalise un mot que l’on valide via la touche entrée. Si le mot est validé, seules les cases sélectionnées sont supprimées.

Le mode Anagram est initié via la touche ‘a’. Il permet des qu’une ligne est de choisir des lettres d’une seule ligne complète afin de former un mot. On soumet le mot à la validation du dictionnaire via la touche entrée. Si le mot est validé la ligne entière est supprimée.

Au fur et à mesure du déroulement du jeu, des bonus apparaissent sur le terrain. Ces bonus ont plusieurs effets :

- Points bonus

- Effet miroir (le jeu est inversé)

# Thème

## Visuel

Une cohérence visuelle a été pensée lors de la conception des différents éléments graphiques composant le jeu. Ainsi, chaque élément a été réalisé afin de correspondre avec le thème choisi, une ambiance sombre, souterraine. Une police personnalisée a également été choisie afin d’être en corrélation avec le thème choisi. L’image de fond représente ainsi un ensemble de structures que l’on peut retrouver au fond d’une grotte: stalactites et stalagmites. Ces éléments sont représentés selon un niveau de profondeur pour donner une plus grande immersion au sein du jeu.

*Image de fond*



Tous les autres éléments viennent se superposer sur cette image de fond (hormis les briques de Tetris) utilisent un fond avec une texture rappelant une roche de couleur marron.

## Le logo du jeu

Un logo pour le jeu a été réalisé, reprenant brièvement les différents composants graphiques afin de de rendre un résultat simple mais efficace, toujours en adéquation avec le thème.



## Les boutons

Chaque bouton possède deux états, actif et passif, respectivement représenté par une image adaptée afin de signaler directement à l’utilisateur s’il peut ou non interagir avec ces boutons. Ces état de boutons ont été pensé afin d’améliorer significativement l’expérience utilisateur en terme d’IHM.

*Bouton actif Bouton inactif*



## Le visuel d’input

Chaque mot entré par l’utilisateur en mode Worddle ou Anagram est affiché dans l’interface par le biais de l’élément graphique input. Un témoin de validation est présent sur cet élément afin de signaler en temps réel au joueur si le mot qu’il vient de rentrer est valide ou non.

Tout comme les boutons, l’élément d’input possède un état actif (lorsque le joueur est en mode Worddle ou Anagram) et un état passif (lorsqu’aucun des deux modes n’est activé).

*Input actif*

https://lh3.googleusercontent.com/aCaYyS5CYZb95uj8RG-hbUovz6Y98fet9nJN5VHnXLbYm8r5cHWBrDb4LoYh5qFKWjaH2oJ89XDb5yHMm8-1vrLypqhBxm0amdTT5PuFrzATSLfThcSA2s7O0sEhh70aiA  


*Input inactif*

https://lh3.googleusercontent.com/YxkEu283rRcdVznuvqd1hKfzp-cLKabFQyHQk6jHyDFyb4ObQCTE898TrDSXQRC0F288oA0lR1SLnp7CNWzitOkQ0LMh03OsMh6ktq9LY-EpkEb9qnCcP7Ej7Bj9LrkzQQhttps://lh3.googleusercontent.com/VW4bKCX-fkSDPg3oiXWm9jVE3e5uGUCFdCW_zxlquSmanJReif1mi5vb0mFr07O0jbpa-aKJHTHOWah1WREkL9Iebz0SaNvnLEDdsE0eSwT9ROCmEOF-jqYNCj_x4Pg9Lghttps://lh3.googleusercontent.com/m5Cbr9OkE_ONiPwRLCnKlX1ctvdY5JDQ1D_HeURbQwv3jfcPNlxOMW94ah-4eSGZHzxFqyP5twFtlgiuxS9u8Xtl2f935eVrUVrtRODZSKiFkBhtOHx_4Q88M4tFRknLVg  

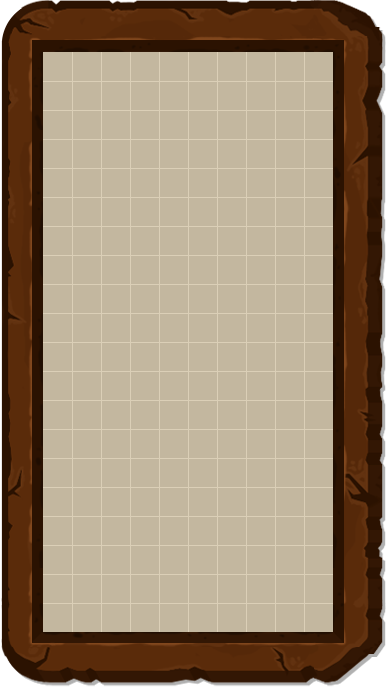

Témoin valide Témoins invalide

## Le Portrait

Le portrait a été conçu en réutilisant le même aspect roche utilisé pour les boutons, l’input et le logo. Un emplacement pour ajouter une image personnalisée a été pensé, bien que cette fonctionnalité ne soit pas actuellement proposée. Sous l’image est donc présent un parchemin sur lequel est affiché le pseudo du joueur actuel. A droite de ces deux éléments on retrouve un emplacement réservé pour afficher le score du joueur, représenté par un ensemble de cristaux pour rester en cohérence avec le thème du jeu.



## La grid

Tout comme le portrait, la grid a été conçue en utilisant l’aspect roche. Un encadré central plus clair ainsi qu’une grille ont  été réalisé afin de faciliter la visibilité du jeu, notamment lors de la chute de pièce. L’aspect roche est donc utilisée ici uniquement en tant que bordure.

## Les briques

8 briques au total ont été réalisées (en comptant celle utilisée pour représenter le bonus et la brique inactive)

Chacune de ces briques a été conçue sur le même Template graphique, seule la (ou les) couleur(s) change(nt). Chaque design de brique a été pensé de manière à ressembler à une sorte de gemme.

https://lh4.googleusercontent.com/tIhS87zTORZjniIFJXHXFyDFHooYbrv2_d4q_7Uh1jx8jZw9tZthJul3FtW06e22RP8dmNNb1l_n4l4s6aZ8mVfCFb9JLophQXgtqLZCMY3Pi9ocCHaOy_0HlWxhQeydlQhttps://lh6.googleusercontent.com/95JV_yXZnrhR9_3C_FMDGwJIKlQZhr9g-1wiBlIbmImpIVWD6JhvImF2Bx6CMuavs4Bs6yqCByHJQKKwLaKbzwYQ4yv98Dgl0smi6qVSXktKygftm3SbuAhxUYbLhd2UQwhttps://lh3.googleusercontent.com/ZPc0bsHMw1P9nWB0Argl_ZBmAlfNe4pbKmmdtdDBHK3mY_W6MIPmA8W2NeiFfBHhSb6CU4v4aYMTHg8zszIvxzOFr5-EP-17UGHXs9-X69CPSRk7ymtyQVdXs2yhLnVJkwhttps://lh5.googleusercontent.com/lbXgW-smDw_2C4oUUDhKsEIV4WHbe9oSgF-quOh-uYH8JP2SRwTNuc5teSVkA2XJ5gKLKf7sGH2xEI-3I4ppUx_yGNqdjMwFAJpcPJSSdmRzq2fzJeeec7a3kecReqAvnwhttps://lh6.googleusercontent.com/up8BPmcqEMco7KgFYh96LFgWmxDHKa0iIJ4qMsZYAbC3JlavlVRyauKxiqa9BJXXbqmA6OvBvD9mhD_YWchVhMxUaJIEhYkQuZvIGy33WATU4XJ9NoSuSJjCtO7zSyzbXQhttps://lh3.googleusercontent.com/BKO-drCm1ZTDDL1OrosIUxipbbOPG9POFR6yekHSKaKxe44SrLjCMb5fAUjzk97aAjOz4reXEpikwgqHrwn4OxPBzvZXGodeKhf3Mk6dAX7Fqr9fOpG1oFQagKIkOlrBIAhttps://lh5.googleusercontent.com/-0RS_h6FLmXXsSMbzwVUAipVnr3ogIXxHswRbE-4p3HgPzTu9oGI0T-HFAfoReIGEFwz2FVuir_seSDgrsSRzRzz1uuCZ93aapxOBH-5pweXiXojhaCpZ52HiK_aqYDrAghttps://lh3.googleusercontent.com/yQ5dlxjjGqotJCicczANIttlpIXEZQcwE_QGPnQNEcV0lArOq8AieIK9odugivpJsB6HbTg7gTrhF_8coLFptkZC__xV2kmRPqDsA-EUCD7rjX2in2As_wGmRiXxOvDZqA

## Contrôles

Flèche gauche - Flèche droite : Déplacer la pièce vers la gauche ou vers la droite

Flèche bas : Descendre la pièce plus rapidement

Flèche haut : changer l’orientation de la pièce

W : Mode Worddle

A : Mode Anagramme

Entrée : Valider un mot (en mode Anagramme ou Worddle)

# Conception

## La création des briques de jeu

Chaque pièce qui tombe est composée de plusieurs briques. Chaque pièce a une position qui elle change et chaque briques qui la compose à des coordonnées dans un repère local à la pièce. Cette disposition permet une rotation des pièces faciles et efficaces et à fortiori d’ajouter des pièces facilement à la liste des pièces disponibles.

Chaque brique voit sa forme générer aléatoirement entre toutes les formes et couleurs possibles de manière équiprobable. De même chaque case composant la brique voit son contenu (une lettre) générée de manière équiprobable.

## Recherche dans le dictionnaire

Le dictionnaire est un arbre contenant tous les mots, pour chercher un mot il suffit donc de faire un parcours d’arbre, nous avons retenu cette solution car c’est la meilleur que nous avons trouvé afin de stocker et chercher des mots anglophones.

En pratique, on parcourt le mot à rechercher lettre à lettre en regardant si la lettre précédente possède dans ses fils la lettre suivante

## Le passage des données à l’interface

Chaque classe est spécialiste et munie de fonction permettant d’obtenir et définir ses champs de manière pertinente. Dans sa globalité le code est séparé en deux parties distinctes : la gestion des mécaniques du jeu et l’affichage des différentes données pour l’utilisateur.

Cette séparation permet d’obtenir un moteur d’affichage efficace car il obtient toutes les données nécessaires à l’affichage via une seule classe qui est le contrôleur des mécaniques du jeu.

# Problèmes et solutions

## Gestions des inputs

Elle concerne 2 choses : la gestion des touches clavier dans un premier temps, puis la gestion de la souris.

La gestion du clavier a connu 2 versions principales : une version qui héritait de la classe Scanner pour gérer les touches clavier. Nous nous sommes rapidement rendu compte de plusieurs problèmes associés à l’utilisation d’une telle classe : les entrées clavier étaient limitées et nécessitaient un temps entre les saisies pour fonctionner efficacement. De plus, la communication avec l’interface réalisée sous Swing n’était pas aisée. Au final nous avons préféré utiliser une classe héritant de KeyListener afin d’utiliser facilement le clavier (et les méthodes associées aux touches) avec Swing et avec la matrice de jeu.

L’autre problème était lié à la gestion de la souris : nous utilisons MouseListener afin de manipuler simplement le clic de la souris. Nous avons cependant eu un problème avec la communication avec l’interface de jeu par rapport à la position de la souris (et de la position du clic pour être précis) afin que les différents éléments (en particulier les différents blocs de couleur) puissent être sélectionnés. Au final, nous avons décidé d’utiliser les coordonnées en pixel afin de gagner en temps de réponse au sein de Layout puisque dans les pixels sont présents dans toute la fenêtre de jeu.

## La Gravité

La gravité, après moult essais, a été séparée en deux forces distinctes :

Une est destinée uniquement à la pièce tombant et une aux différentes cases posées en bas de la matrice. Cette gestion différenciée de la gravité permet une chute différenciée des pièces et des cases et permet de mettre en pause facilement et de manière efficace la chute d’un élément.

## La gestion des collisions des briques

Avant chaque application de la gravité, chaque case de la pièce teste si elle peut chuter dans la case du dessous si une seule d’entre elle ne peut pas, chaque case de la pièce tombante est ajouter à la matrice des cases tombées, la pièce présente dans le buffer se met à chuter à la place de l’ancienne, et le buffer reçoit une nouvelle pièce.

La collision gauche et droite avec les bords de la matrice et les autres cases présentes sont gérées de manière similaire.

# Conclusion

L’objectif principal du projet était de développer un Tetris original, avec cette dimension de lettres avec l’optique d’avoir un jeu et le dictionnaire en anglais

Si le temps nous l’avait permis, nous aurions aimé développer un mode multijoueur amusant, ainsi qu’une tripotée de bonus et de malus différents, comme plusieurs modes de difficultés (Lettres X et Z plus fréquentes, difficultés du meilleur anagramme augmentée, malus plus présents que les bonus)

Ce projet nous a permis de mettre en application nos connaissances et de programmer un jeu complet uniquement avec Java (et Swing), et de nous rendre compte des possibilités offertes par le langage Java en terme d’optimisation et d’architecture de code.