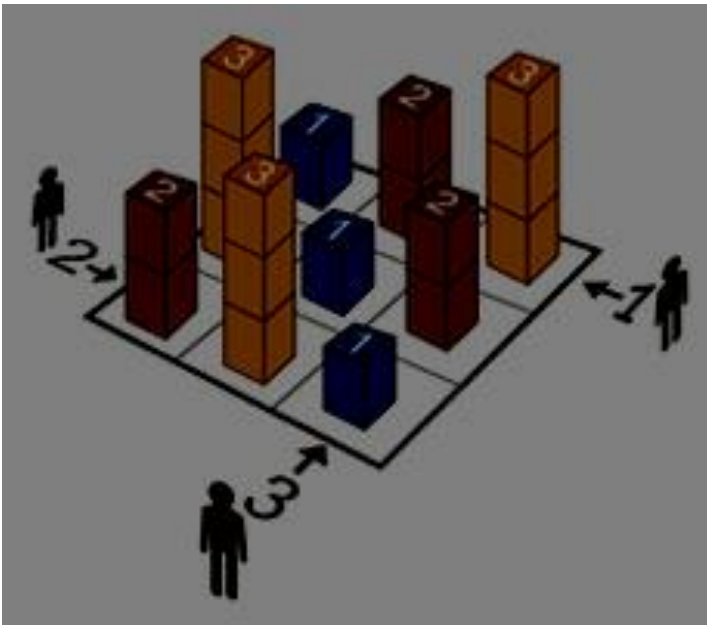


# Skyscraper Puzzle

## Rapport de spécification



**GOLDMAN Violetta  
VANRYCKE Loann  
MOHAMMEDI Vanessa  
EL HAOUAJI Hiba  
ROZBULLY Faraan**

# Introduction :

Dans ce projet, nous avons décidé de nous atteler à la réalisation d'un algorithme codé en langage C# et JavaScript avec l'aide de l'application Unity, afin de reproduire le casse-tête du nom de Skyscraper Puzzle.

Il s'agit d'un jeu de logique basé sur une grille carrée de dimension 4x4 pouvant évoluer vers des niveaux de difficulté atteignant une dimension 6x6. Le but du jeu est de remplir une grille de taille NxN avec des chiffres allant de 1 à N. Suivant la difficulté de la grille, de rares ou nombreux indices placés à l'extérieur de la grille seront donnés aux joueurs, précisant le nombre de gratte-ciel visibles au point de vue de l'indice. Il faut savoir qu'un immeuble est le seul de sa hauteur sur la ligne et la colonne qu'il occupe, ainsi nous ne retrouverons pas deux fois le même chiffre sur une ligne ou une colonne, ce dernier indiquant la hauteur du gratte-ciel. Sachant qu'un gratte-ciel cache derrière lui tous les immeubles plus petits, on ne pourra pas voir un immeuble de 1 ou 2 étages situé derrière un immeuble de 3 étages d'un certain point de vue. Le but du jeu sera donc de trouver la disposition des immeubles dans la grille à partir des indices situés à l'extérieur de la grille.

## Description d'écrans:

Menu principale: nom, choix de niveau (3 boutons - facile, moyen, difficile), préférence , l'aide (explication de règles).

Préférence: bouton de retour au menu principale, choix de langues (français/ anglais), effets sonores (activé/désactivé), musique (activé/désactivé), mode (dark/ light)

L'écran du jeux: grille de jeu, menu, préférences, aide.

## Diagramme de spécification:

Voici un schéma qui explique le déroulement de la réalisation de notre projet. Ce schéma comprend la partie algorithme ainsi que la création de l'interface du jeu :

[illegible]

# Étapes de la réalisation du projet :

## Gestion du projet :

L'élaboration de notre projet nous a poussé à dresser un schéma de travail constitué de :

- Une répartition de tâche équitable entre les membres du groupe ;
- Une liste détaillée des différentes étapes du projet ;
- Un diagramme de Gantt permettant le suivi du développement du projet ;
- Une création de maquette pour visualiser l'aspect global de notre travail ;
- Une liste de problèmes à résoudre ;

## Operating Interface:

- Equipe: Vanessa MOHAMMEDI, Hiba EL HAOUAJI
- Durée: 15 semaines

### Description de l'interface :

L'interface utilisateur est un dispositif matériel ou logiciel permettant à un usager d'interagir avec un produit informatique. C'est une interface informatique qui coordonne les interactions homme-machine, en permettant à l'utilisateur humain de contrôler le produit et d'échanger des informations avec le produit. Il s'agit donc de la partie visible de notre projet, nous nous devons donc de la rendre la plus simple d'utilisation et la plus esthétique possible.

Nous souhaitons modéliser une interface assez sobre mais tout de même complète afin de proposer à l'utilisateur la meilleure expérience de jeu. Nous intégrerons un maximum de fonctionnalités au jeu afin d'apporter le plus grand confort de jeu à ce dernier.

### 1) Fonctionnalités essentielles :

- Menu avec choix de la langue (Français et Anglais), la langue sera réglée automatiquement sur le Français
- Choix de la difficulté : création de 3 boutons pour chaque niveau de difficulté (facile, intermédiaire et difficile)
- Bouton « Nouvelle partie » permettant de générer une nouvelle grille de jeu
- Bouton « Règle du jeu » : fournit au joueur les règles du jeu
- Bouton « Aide » : fournit un indice au joueur lui révélant la position d'un chiffre dans la grille. L'utilisation de cette fonctionnalité sera limitée pour chaque partie

## **2) Fonctionnalités liées au confort de jeu :**

- Bouton « Tout effacer » ou « Recommencer » qui effacera uniquement les chiffres ayant été rentrés dans la grille par le joueur afin de recommencer la même partie sans génération d'une nouvelle grille de jeu. Cette fonctionnalité évitera à l'utilisateur de tout effacer manuellement
- Ajout d'une musique de fond. L'utilisateur aura la possibilité d'activer ou de désactiver cette musique. Ce bouton sera représenté par un logo. Lorsque l'on cliquera dessus, le logo changera en un logo « muet » et la musique sera coupée
- Ajout d'effet sonore pour chaque action du joueur dans la grille. Ce dernier aura la possibilité d'activer ou de désactiver ces effets sonores de la même manière que la musique de fond
- Ajout d'un bouton permettant de basculer la page sur un « thème foncé » offrant au joueur un environnement de jeu plus agréable, lors d'une partie nocturne

Nous nous chargeons de créer une interface agréable au jeu, pragmatique et opérationnelle. La réalisation de l'interface du plateau de jeu consiste à lier le visuel à l'algorithme, tandis que les autres fonctionnalités, mode, son, langue, règles du jeu ou autre, seront codés selon nos préférences. Notre objectif est de reproduire un environnement de jeu conforme aux attentes du joueur, lui facilitant la navigation sur le site (ou application) aisément.

Nous discutons encore la possibilité d'utiliser le framework Unity, celui-ci pourrait nous être d'une grande aide, comme il pourrait retarder l'aboutissement du projet, puisqu'il faut d'abord apprendre à le manier avant de l'utiliser.

L'équipe de programmation back-end portera son aide à celle du front-end et vice versa, pour assurer une coordination entre les deux surfaces: nous travaillerons ensemble sur la beauté et l'efficacité de notre projet.

## **Algorithme:**

- Equipe: Violetta GOLDMAN, Faraan ROZBULLY, Loann VANRYCKE
- Durée: 17 semaines

### **Description de l'algorithme:**

Nous avons décidé d'attaquer la partie algorithme par la génération de tableaux  $n \times n$  représentant la solution des plateaux de jeu, ainsi que ses bords. Le remplissage du tableau se fait avec la fonction backtracking.

L'idée est de stocker la solution des grilles, puis lancer un plateau vide lui correspondant.

La génération de ces bords se fait en boucle jusqu'à ce qu'on tombe sur ceux qui vérifient le tableau de jeu initial (selon les règles du jeu). Ainsi, il sera plus facile de tester la validité de la partie jouée, notamment les éventuels cas d'erreur.

Le jeu comptera plusieurs niveaux. Les dimensions de la grille croient en fonction de la difficulté de la partie. Le joueur aura la possibilité de choisir la difficulté du jeu à partir du menu. Nous mettrons aussi en place un système d'aide qui permet de donner la solution d'une case sélectionnée.

## **Version 0:**

- Génération du tableau HTML:

On génère un tableau HTML grâce aux balises <th> et <tr>. Les cellules qui font l'angle prennent l'id « empty » et sont de la même couleur que le background. Les cellules qui entourent le tableau (privé de celles ayant l'id « corner ») prennent l'id « arrow ». Avec le CSS, on utilise border-radius sur les cellules pour leur donner un aspect rond.

- Récupération des informations :

Lorsque l'on clique sur une cellule, on vérifie que son id n'est ni «arrow» ni «empty» et on lui associe l'événement «click». A toutes les cellules ayant l'événement «click» on associe l'événement «keydown» de manière à récupérer la touche appuyée par le joueur.

### Version 1:

- Génération d'un tableau HTML/CSS:

Génération du tableau HTML grâce à la fonction initInnerHTML(tab,size) qui prend en argument le tableau d'id «gamediv» créée dans le HTML et la taille souhaitée (le tableau étant carré). Les cellules qui font l'angle prennent l'id «corner» et sont de la même couleur que le background. Les cellules qui entourent le tableau (privé de celles ayant l'id «corner») prennent l'id «arrow». Le reste des cellules prennent l'id «cell». Les cellules avec l'id «active» sont colorées en rose. Avec le CSS, on utilise border-radius sur les cellules pour leur donner un aspect rond.

- Récupération des informations :

Lorsque l'on clique sur une cellule du tableau, cette dernière est activée grâce à l'événement «click» et la fonction makeActive (qui donne l'id «active» aux cellules cliquées). Ensuite grâce à l'événement «keydown» donné aux cellules «actives», on récupère la touche appuyée par l'utilisateur et on vérifie si elle appartient bien à l'ensemble (1,2,3,4,0) , (0 servant à effacer un chiffre déjà écrit). On écrit ensuite ce chiffre dans le texte HTML de la cellule.

- Code JavaScript :

```
function Init(size) {
  /*int -> HTMLElement[]*/
  var tab = document.querySelector("#gamediv"); // récupérer la zone de création du tableau
  tab = initInnerHTML(tab,size);                // créer le tableau aléatoire
  initClickListeners(tab);                      // rendre les cellules du tableau clickables
  initKeyListener(size);                        // écouter les touches appuyées
  return tab;
}
```

# Conclusion:

Pour conclure, nous développerons un jeu plutôt sobre mais très complet, avec de nombreuses fonctionnalités permettant au joueur de profiter pleinement de sa partie. Le menu du site sera très intuitif facilitant la prise en main du jeu, avec diverse niveau de difficulté et un choix de langue entre Français et Anglais, rendant le jeu accessible à un très large panel d'utilisateurs.