Cahier des charges

Jeremiah Sergius, Charlotte Pochez et Oumayma Torkhani

1. Les grands principes du projet

Le Tusmo, aussi connu sous le nom de Wordle, est un jeu dans lequel l'objectif est de deviner un mot. On commence par choisir la difficulté qui détermine le nombre de tentatives possibles, puis on choisit la longueur du mot à deviner. Le jeu, fonctionnant par tour, commence alors. Au début du premier tour, on rentre un premier mot. Le programme compare le mot entré par le joueur au mot choisi au hasard par l’ordinateur. L’état de chaque lettre change : si la lettre devient grise cela signifie qu’elle n’est pas présente dans le mot à deviner, si la lettre devient jaune c’est qu’elle est présente dans le mot à deviner mais qu’elle est mal placée et si la lettre est rouge c’est qu’elle est présente et bien placée. Au tour suivant le joueur réessaye de deviner le mot, puis le programme compare ce mot à celui à deviner et ainsi de suite jusqu’à ce que le nombre d’essais sélectionné au début par le joueur soit atteint. Si pendant un tour le joueur rentre le même mot que celui sélectionné par le programme alors il gagne et le jeu s’arrête. Si le nombre d’essais est dépassé et que l’utilisateur n’a pas trouvé le mot, il perd. La solution est alors affichée à l’écran

1. Les fonctionnalités de base choisies

* Pouvoir faire tourner le jeu avec un seul niveau de difficulté (nombre d’essais)
* Avoir un affichage du jeu mais jouer dans la console Idle
* Utiliser pygame

1. Les fonctionnalités avancées

* L’intégration des plusieurs niveaux de difficulté
* Pouvoir interagir avec l’affichage graphique pour jouer (ajout d’un clavier dans la fenêtre de jeu et donc changement de l’aspect de ce clavier au cours du jeu: quand une lettre a été définie comme n’appartenant pas au mot à trouver elle n’est plus utilisable)

1. Documentation des fonctions

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| nom | lettre\_correcte | mot\_correct | intro | difficultes |
| But | vérifier si la lettre | comparer le mot entré par l’utilisateur à la solution  (appelle lettre\_correcte) | présenter les règles du jeu à l’utilisateur s’il le souhaite | présenter les différents niveaux de difficulté et demander à l’utilisateur de choisir |
| Attribut | mot : le mot entré par l’utilisateur  place : le rang de la lettre dans le mot  solution : le mot-solution | mot : le mot entré par l’utilisateur  solution : la solution de l’ordinateur |  |  |
| Renvoie | le statut de la lettre :  0 si absente  1 si mal placée  2 si bien placée | True si l’utilisateur a trouvé le bon mot  False sinon | affiche les règles à l’écran | affiche les niveaux de difficultés à l’écran |

1. L’algorigramme du programme

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

1. La répartition des tâches

Charlotte : listes de mots aléatoires de différentes tailles (entre 4 et 8 lettres), cahier des charges, affichage graphique basique

Jeremiah : présentation des règles et des niveaux de difficulté pour l’utilisateur, algorigramme, fonction de jeu

Oumayma : fonctions qui déterminent si l’utilisateur a trouvé le bon mot, cahier des charges, différents niveaux de difficulté