

Numériques et Sciences Informatiques

Projet de groupe n°1

Augustin B. - Victor H. - Matéis R.

Octobre - Novembre 2022

La recherche informatique peut se résumer en plusieurs grands axes, mais un des principale est de trouver des solutions à des problèmes. Avec le problème du coffre-fort, il faut raisonner en programmation orientée *objet*. Ce grand principe de l'informatique moderne consiste à définir et à interagir avec des briques logicielles appeler objets. Cependant, c'est au développeur d'implémenter ses propres objets en fonction de ses besoins. Il possède une logique un fonctionnement parfois inconnu pour l'utilisateur, ce qui permet de définir ce paradigme de programmation comme abstrait. Il y a bien des avantages à programmer avec des objets, mais c'est une manière totalement différente de penser la programmation du problème.

Part I

Initialisation du projet

Dans notre cas, il faut résoudre un problème de coffre fort avec des combinaisons. Ce jeu se joue à plusieurs joueurs avec autant de cadenas que de participants. Le but de ce jeu est de trouver la combinaison de ce coffre avant les autres joueurs, celui qui trouve la combinaison en premier obtient une plus grosse part du trésor, cette part diminue en fonction du nombre de joueur qui ont trouver le code avant. Dans le cas où il n'y a qu'un seul joueur, ce participant est limiter en nombre d'essais.

Quand on lance le jeu, après avoir indiquer le nombre de joueurs, des combinaisons de 5 caractère se génère aléatoirement a raison de une combinaison par joueurs. Lorsque l'on essaie de rentrer une combinaison, le programme nous indique s'il y a des valeurs de l'essai présente dans la combinaison.

Part II

Le travail au sein du groupe

Nous avons utiliser l'outil de collaboration en ligne Github, qui nous permet de centraliser le code en un seul endroit avec des fonctions d'édition du code collaboratives. Nous avons travaillé 8h sur le projet. À raison de 8h chacun soit 24h, les 4 premières heures étaient en groupe et étaient dédiées à la compréhension du sujet ainsi qu'à l'écriture des différents algorithmes. Par la suite nous nous sommes chacun occupé d'une partie de la traduction en langage python ainsi que la rédaction de ce document, comprenant de l'entraide entre nous. Les tâches étaient réparties de telle que Augustin étaient assigné à l'écriture de la classe "ListeCarCombinaison" ainsi que le fichier "main", Victor à l'écriture de la classe "Cadenas" et une partie du document pdf, Matéis à l'écriture du document pdf, à l'écriture des algorithmes et à l'écriture d'une partie du fichier "main".

Part III

Algorithme générale

Pour l'algorithme de ce jeu, le programme se découpe en 3 parties: la classe *Cadenas*, la classe *ListeCarCombinaison* et le fichier *main*.

Dans le fichier *main*, on commence par entrer le nombre de joueurs, ensuite une combinaison à 4 caractère est généré pour chaque joueurs s'ils sont supérieurs ou égale à 2. Si le joueur est seul alors il à 10 essais pour trouver la combinaison. Ensuite, on demande aux joueurs de rentrer la combinaison qu'il pensent correcte et on procède aux vérifications. Le cas pour 1, 2, et plus de 2 joueurs.