  
  
Projet de Jeu : Calcul Mental.

Préparé par  : DRAOUI Oumayma et BARUA Puspita

Lycée galilée, terminal s, option isn.

sommaire

INTRODUCTION

ANALYSE DU BESOIN-RECHERCHE D’IDÉES

RÉPARTITION DES TÂCHES ET DÉMARCHE COLLABORATIVE

Lycée galilée, terminal s, option isn.

RÉALISATION

INTÉGRATION ET VALIDATION

BILAN ET PERSPECTIVES

DIFFUSION DE PROJET

INTRODUCTION

Le calcul mental est une activité qui consiste à effectuer des calculs sans autres supports que le réflexion et la mémoire. Il s'appuie sur un certains nombre de technique et d'astuces, ainsi que sur la connaissance de certaines règles de calcul, comme les tables de multiplications. Le calcul mental est utilisé quand très peu de moyen pour calculer sont à dispositions, ou encore pour donner rapidement un ordre de grandeur d'un résultat trouvé pour valider sa valeur.

Dans le cadre de ce projet, nous avons décidé de réaliser un programme permettant de fournir à un utilisateur des calculs fournis de manière entièrement aléatoire par l'ordinateur. Ainsi, tous les nombres, les types de calculs sont fournies aléatoirement. Il doit également proposer à l’utilisateur de recommencer, en cas d'erreur. Le programme devait s'articuler de la manière suivante:

* + - L'ordinateur doit fournir une série de calcul à l’utilisateur;
    - A chaque question, le score total de l’utilisateur augmente d’un nombre à chaque bonne réponse;
    - Le programme doit fournir différents types de calculs de difficulté variable;
    - Le jeu commence au niveau 1 automatiquement et passe au niveau suivant après 5 bonnes réponses.

Le jeu réalisé est un jeu destiné à un public relativement jeune, ce qui explique la simplicité du programme ainsi que celle de l’interface graphique.

ANALYSE DU BESOIN-RECHERCHE D’IDÉES

Cahier de charge :

On souhaite créer un programme qui propose des questions de calculs mentaux (addition, soustraction, multiplication et division euclidienne) à l’utilisateur en vérifiant les réponses.

* Les questions posées vont être aléatoires selon le niveau ( Niveau 1, 2 ou 3)
* Une bonne réponse compte 1 point ; Les points vont être additionnés pour donner le score final de l’utilisateur.
* A un seuil de score donné, l’utilisateur passera automatiquement au niveau suivant.
* L’interface du programme va être simple à comprendre et colorée, car elle doit attirer un public à priori très jeune.
* Un bouton quitter doit être proposé à l’utilisateur qui lui permet de quitter le programme plus facilement.
* Un bouton recommencer permettra à l’utilisateur de recommencer le jeu à la fin.
* le jeu s’arrête après avoir passé les 3 niveaux de difficulté.
* L’utilisateur aura un temps limité de 5min pour répondre au maximum de questions.

Le language informatique utilisé est le Python.  
L’interface graphique est créée grâce au module Tkinter. La majorité de nos sources d’information qui permettent de réaliser ce projet proviennent de la recherche sur Internet surtout pour l’interface graphique car nous n’avons jamais travaillé avec Tkinter.  
 Cependant, notre logique et notre propre raisonnement restent indispensables pour le bon usage de ces informations.

RÉPARTITION DES TÂCHES ET DÉMARCHE COLLABORATIVE

**Répartition des rôles :**Oumayma : Chef de projet.  
Puspita : Répartition des tâches.  
  
**La répartition des tâches** a été faite en fonction des phases de programmation:

Phases de programmation :

**Phase n°1** : Un programme qui vérifie la réponse d’une question donnée.  
 L’utilisateur entre la réponse dans la console. (Oumayma)

**Phase n°2** : Le programme doit afficher une question créée aléatoirement. (Puspita)  
  
 **Phase n°3** : Le programme doit calculer la bonne réponse de la question posée. (Puspita)  
  
 **Phase n°4** : Le programme doit dire au joueur si la réponse entrée par ce dernier est bonne, sinon il doit lui proposer de recommencer jusqu’à ce qu’il trouve  
 la bonne réponse. (Oumayma)  
  
 **Phase n°5**: Le programme affiche plusieurs questions de manière consécutive. Une deuxième question s’affiche suite à la bonne réponse de la première question. (Puspita)

**Phase n°6**: Le Score – une bonne réponse = 1 pt ; une mauvaise réponse = 0 pt. (Oumayma)

**Phase n°7** : Définir 3 niveau du jeu (Easy ; Normal ; Hard). (Oumayma)  
  
 **Phase n°8** :La difficulté de la question dépend sur le niveau du jeu. (Puspita)

DÉMARCHE COLLABORATIVE :

Au début, la répartition des tâches a été faite de manière très précise. Cependant au fur à mesure de l’avancement de notre projet cette répartition est devenue de plus en plus floue car nous avons décidé de s’entraider et de construire le projet ensemble, nous avons plus ou moins travaillé sur certaines étapes ensemble en réfléchissant ensemble de façon très collaborative grâce à la plate-forme DROPBOX ainsi que les autres moyens de communication qui nous ont permises de rester en contact afin de pouvoir communiquer nos difficultés et nos avancements.   
  
 Personnellement, je trouve que le travail collaborative a été plus efficace car la répartition des tâches a été faite aléatoirement, donc parfois nous avons rencontré des difficultés lorsqu’il s’agissait d’une tâche qui nous correspondait pas personnellement; Alors que le fait de travailler ensemble nous a permises de choisir d’avancer sur les aspects qui correspondent à notre niveau de compréhension et de logique par rapport au projet. Parfois on a été amenés à finir la tâche de l’autre ou la commencer, ce qui a été, à mon avis, très collaborative et donc efficace.

RÉALISATION

Afin de réaliser les tâches de programmation, j’ai utilisé plusieurs fonctions qui permettent de réaliser chaque phase de programmation, vous trouverez les listings du programme en entier présentés dans l’annexe.  
 Je tiens à préciser que je vais seulement présenté ici les fonctions sur lesquelles j’ai travaillé personnellement.  
  
 Tout d’abord afin de calculer la réponse de la question posée aléatoirement, j’ai fait une fonction qui est sous forme de boucle If … return . Elle consiste à prendre les valeurs de l’opération qui sont déjà déterminée dans la fonction Question tout en précisant l’opération (Addition, soustraction, multiplication ou division), les valeurs sont calculées et le return permet de les stocker à nouveau sous leur nouvelle valeurs. Ensuite pour vérifier la réponse de l’utilisateur, j’ai crée une fonction Jeu qui demande à l’utilisateur d’entrer une valeur puis la compare avec la réponse déjà calculée par le programme, cette fonction permet aussi d’afficher un texte qui soit félicite l’utilisateur lorsqu’il s’agit d’une bonne réponse : « BRAVO! Bonne réponse! » ou lui propose de recommencer lorsqu’il se trompe : « FAUX! Essayez encore! » .