MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

BUT Informatique
1ère année
Groupe 1B

SAE 1.01
Implémentation d'un besoin client

MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

Sommaire:

MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

Introduction

Dans le cadre de notre SAE, nous avons eu l'opportunité de concevoir et de programmer une application de course de chevaux, répondant ainsi à l'implémentation d'un besoin client.

Ce projet nous a permis de mettre en pratique nos compétences en programmation tout en créant une application interactive.

Notre objectif était de développer une application où l'utilisateur serait un arbitre de courses de chevaux, souhaitant à la fin de cette dernière obtenir facilement le classement et les résultats. Au cours de ce projet, nous avons eu l'occasion de mettre en œuvre des concepts de programmation tels que la saisie utilisateur, la manipulation de données, la logique de jeu et l'affichage des résultats. Notre application de course de chevaux a été conçue pour simuler deux manches de compétition, où l'utilisateur doit fournir des informations sur les compétiteurs, telles que la longueur de la course, le nombre d'obstacles, les chutes éventuelles et les temps des joueurs. À la fin des deux manches, le programme retourne un podium final, révélant les meilleurs compétiteurs de la course.

La présentation de notre travail commencera par l'introduction de l'application en commençant par ces fonctionnalités et limites puis en expliquant le découpage fonctionnel effectué du problème. Ensuite nous montrerons comment nous avons testé l'application, en introduisant nos différents jeux d'essai.

MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

Présentation de l’Application

Fonctionnalités et Limites

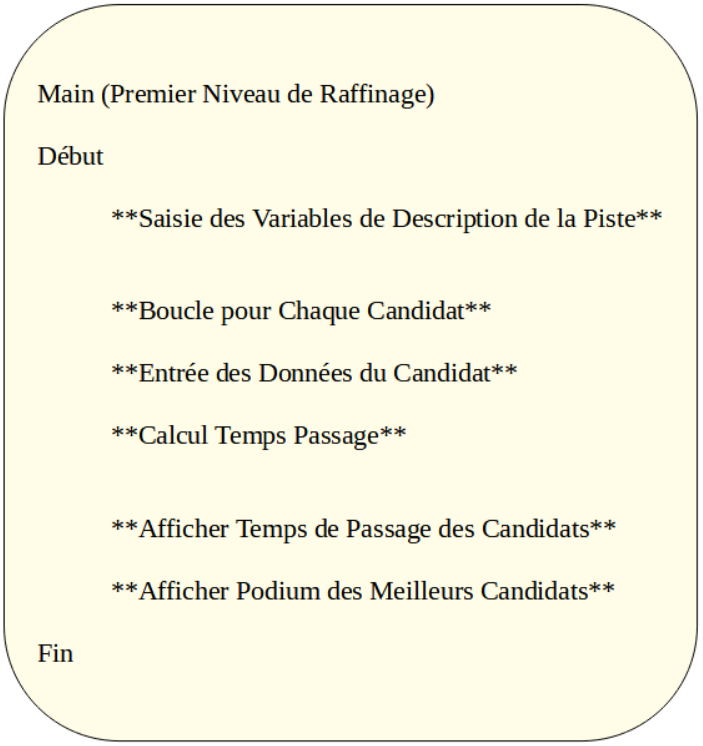
Notre application de course de chevaux permet à l'utilisateur de configurer des courses en entrant des données. Elle simule deux manches, génère un classement basé sur le temps total, incluant les pénalités dûes aux fautes du compétiteur, et affiche le podium, composé des différents gagnants de la course. Elle gère les situations spéciales, y compris les égalités et l'élimination des joueurs en cas de chutes répétées. L'application offre une personnalisation totale à l'utilisateur, lui permettant de choisir librement le nombre d'obstacles et de barres. Cependant, pour maintenir la cohérence du jeu, toutes les saisies sont contrôlées pour éviter les configurations incompatibles. De plus, elle convertit et affiche le temps en minutes, secondes et millisecondes.

Cependant, certaines faiblesses méritent d'être notées. Tout d'abord, l'application ne permet pas de gérer des courses avec plus de 50 compétiteurs, ce qui peut limiter son utilité dans certaines situations. De plus, le processus de saisie de toutes les données peut être assez long, en particulier avec un grand nombre de compétiteurs. L’application peut être assez floue au début car pas de texte affiché lors du lancement de l’application.L'application présente ne permet pas aux utilisateurs de personnaliser des éléments clés, tels que le type de terrain, les conditions météorologiques, ou la difficulté de la course. Enfin, l'application est entièrement textuelle, sans aucune image ou élément visuel, ce qui peut réduire son aspect esthétique pour certains utilisateurs.

MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

Découpage du Problème

La première étape de résolution d'un problème en groupe est, dans n'importe quel projet, de répartir les différentes tâches au sein de ce même groupe. Afin d'y parvenir nous avons dans un premier temps fait un premier raffinage de notre fonction principale et donc des différents problèmes qu'il faudrait résoudre indépendamment. On a donc rapidement obtenu cette ébauche de l'application



notre application se découpera donc en 5 sous programmes:

- *calculTemps* visant à évaluer le temps avec les pénalités de chaque compétiteurs.
- *afficherPodium* visant à trouver puis afficher les 3 premiers(et plus si égalité).
- *conversion* visant à convertir le temps en minutes, secondes et millisecondes. Avant de l'afficher.
- *saisieIntMin* visant à effectuer une saisie contrôlée d'un entier pour qu'il soit supérieur à un minimum.

- *saisieIntMinMax* visant à effectuer une saisie contrôlée d'un entier pour qu'il soit supérieur à un minimum et inférieur à un maximum.

L'application commencera donc par effectuer la saisie contrôlée des différentes données de la piste avec *saisieIntMin* et *saisieIntMinMax*. Puis elle fera directement un calcul du temps de chacun des candidats en récupérant chacune de leur données et effectuant le calcul grâce à *calculTemps.* Elle utilisera ensuite *afficherPodium* pour trouver les premiers, puis affichera leur numéro et leur temps par l'intermédiaire de *conversion*.

Jeux d'essais

Voici les jeux d'essais pour la fonction main :
Tout ce fait dans une saisie contrôlée ainsi ces cas ne sont théoriquement pas possible.

nbComp	lenPiste	nbObstacle	nbBarres	nbBarresTombees	nbRefus	chute	tempsReel	Résultat
-2								erreur nbComp <0 ou nbComp>50
5	-2							erreur lenPiste <0
5	650	700						erreur nbObstacle <1 ou >lenPiste
5	650	300	33					erreur nbBarres <nbObsctacle*2 ou nbObstacle*4
5	650	300	600	650				erreur nbBarresTombees > nbBarres
5	650	300	600	2	-2			erreur nbRefus <0
5	650	300	600	2	2	oui		erreur chute !='true'/'false'
5	650	300	600	2	2	true	-6	erreur tempsReel <=0
5	650	300	600	2	2	true	55555	Fonctionne

Voici les jeux d'essais pour la fonction calculTemps :

pfTemps	pfNbBarres	pfNbRefus	pfChute	pfLenPiste	Résultat
80000	0	0	false	650	80000
80000	2	0	false	650	96000
80000	2	6	false	650	-1
80000	2	0	true	650	-1
120001	2	0	false	600	-1
180001	2	0	false	700	-1

MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

Voici les jeux d'essais pour la fonction conversion :
Cette fonction est appelée uniquement pour certaines valeurs dans la fonction podium. De plus "pfTpsms" ne peut pas être égal à 0 car il vient de la saisie contrôlée du main.

pfTpsms	Résultat
180 000	3min 0s 0ms
125 005	2min 5s 5ms

Voici un jeu d'essai pour la fonction afficherPodium

pfscore[]	pfnbComp	Résultat
80000 / 96000 / 120000	3	La premiere place revient au joueur n°: 1 avec un temps de : 80000 La deuxieme place revient au joueur n°: 2 avec un temps de : 96000 La troisieme place revient au joueur n°: 3 avec un temps de : 120000

Il existe plusieurs scénarios possible, et nous en avons trouvé 8:
Le jeu d'essai en haut correspond au septième. C'est le cas où le podium fonctionne de façon classique, c'est le cas qui sera le plus souvent possible. Cependant il en existe 7 autres et les voicis :

- Premier cas particulier

3 premiers donc pas de deuxième ou de troisième

- Deuxième cas particulier

1 premiers et plus de 2 deuxièmes
Il faut donc afficher toutes les personnes qui ont fait un temps leur permettant de se qualifier deuxième.

- Troisième cas particulier

2 premiers et tous les autres joueurs éliminés
Il n'y a pas de troisième place sur le podium à afficher.

- Quatrième cas particulier

2 premiers et beaucoup de troisièmes
Cas similaire au deuxième cas. Il faut afficher tous les troisièmes et les 2 personnes qui sont premières.

- Cinquième cas particulier

1 premier 1 deuxième et beaucoup de troisième
C'est presque le même problème que le quatrième cas sauf qu'il a un deuxième cette fois-ci.

- Sixième cas particulier

1 premier et tous les autres éliminés
Il faut afficher uniquement un compétiteur

- Dernier cas particulier

Tous les compétiteurs ont été éliminés
Nous afficherons dans ce cas ceci : Tous les joueurs ont été éliminés

MAGAZ Yahya
LOVIN Alex
DELAPLAGNE Titouan
RAINGEVAL Sacha

Conclusion

Dans le cadre de notre SAÉ d'informatique, nous avons créé un programme qui permet de calculer les temps et le podium d'une course de chevaux. Notre application simule deux manches de compétition, où l'utilisateur doit fournir des informations sur les compétiteurs, telles que la longueur de la course, le nombre d'obstacles, les chutes éventuelles et les temps des joueurs. À la fin des deux manches, le programme retourne un podium final, révélant les meilleurs compétiteurs de la course.

Notre programme a été conçu pour répondre à l'implémentation d'un besoin client. Au cours de ce projet, nous avons eu l'occasion de mettre en œuvre des concepts de programmation tels que la saisie utilisateur, la manipulation de données, la logique de jeu et l'affichage des résultats. Ce projet nous a permis de mettre en pratique nos compétences en programmation tout en créant une application interactive. Ce projet nous a servi de première expérience professionnelle dans laquelle nous avons dû répondre à un besoin client. Notre groupe a dû communiquer de façon efficace afin de réussir. Nous avons également réparti les tâches entre les différents membres du groupe comme l'aurait fait une entreprise.

Notre programme est une solution efficace pour les arbitres de courses de chevaux qui souhaitent obtenir facilement le classement et les résultats à la fin d'une compétition. Nous sommes fiers du travail accompli et nous espérons que notre programme répondra à vos attentes !