



Analyse et programmation orientée objet

Type de projet 1 et 2

Compétition de « Ranch sorting »

Biemar Nicolas

Loforte Mateo

29/05/2023

Description de l'application informatique

Notre application a pour but d'organiser des compétitions de « Ranch sorting ».

Le « Ranch sorting » est une discipline d'équitation western qui consiste à trier des vaches par équipe de deux cavaliers. L'épreuve se déroule dans un petit enclos en forme de huit où les boucles peuvent communiquer par un couloir au centre du huit. Un troupeau de vaches numérotées de 0 à 9 (+ une vache sans numéro) se trouve dans la partie haute de l'enclos.

Le but est de faire passer, dans l'ordre, les vaches de la partie haute vers la partie basse en le moins de temps possible.



Pour ce faire, un numéro est tiré au sort. Les cavaliers doivent ensuite démarrer de la vache ayant le numéro tiré puis, dans l'ordre, faire passer toutes les autres vaches.

Attention, la vache sans numéro ne peut pas passer la ligne et les vaches situées dans la partie basse ne peuvent plus retraverser le huit, dans les cas contraires la partie s'arrête et l'équipe obtient un score nul équivalent à « No Time ».

La partie se termine à la fin du chrono ou bien lorsque toutes les vaches sont passées.

Chaque équipe a un ou deux rounds selon l'épreuve, on cumule les résultats et l'équipe qui a le plus de vaches validées en un minimum de temps gagne l'épreuve.

En ce qui concerne l'application informatique, elle sera tout d'abord utile aux organisateurs de tournois puisqu'elle permettra de créer des équipes, des lieux et des épreuves puis d'inscrire les équipes aux épreuves. Une fois les inscriptions terminées, le tournoi peut commencer.

L'application informatique est ensuite destinée à l'arbitre qui va pouvoir lancer le chrono, annoncer le numéro aléatoire (choisi par l'ordinateur) à l'équipe en jeu et compter le nombre de vaches valides ainsi que le temps de chaque vache. Il pourra aussi indiquer lorsqu'une vache non valide passe à travers le huit. Cette action mettra immédiatement fin au chrono et l'équipe obtiendra alors un score nul. Ces résultats seront stockés dans une base de données Access afin d'établir un classement.

Mode d'emploi

Notre projet utilise une base de données Access : format .mdb 2002-2003 et notre version microsoft office est la version 32 bit.

Fenêtre d'accueil

Lorsque vous lancez notre application graphique, vous arrivez sur la fenêtre d'accueil qui, comme vous pouvez le voir ci-dessous, est divisée en 2 parties distinctes : la partie gauche est destinée à l'ajout et la suppression d'équipe dans la base de données et la partie droite est dédiée à la création d'épreuves.

NomEquipe	NomCavalier1	NomCavalier2	NomCheval1	NomCheval2
Bloute	Mateo	Lea	Smarty	Hope
Dream Team	Tom	Joe	Eclaire	Foudre
Famille 1	Leo	Mateo	Sugar	Smoky
Team Cowboy	Davide	Bill	Croquette	TheKing

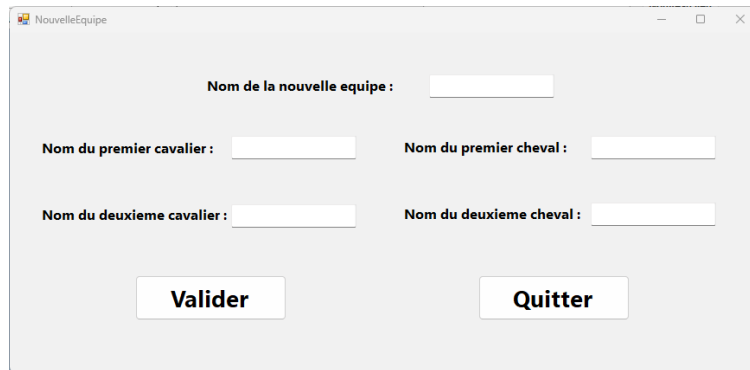
Visualisation d'une épreuve existante

Lorsque vous appuyez sur « Visualiser une épreuve existante », vous avez accès à la liste des épreuves existantes et vous pouvez également obtenir la liste des inscrits en suivant les instructions.

NomEpreuve	DateEpreuve	NomLieu	NumRound
Championnats de Belgique	27-5-2023	Ranch Leo	2
Championnats du Monde	27-5-2023	Ranch Leo	2
Belgium GrandPrix	28-5-2023	Ranch Leo	2
Toumois simple	28-5-2023	Ranch Leo	1

Création ou suppression d'une équipe

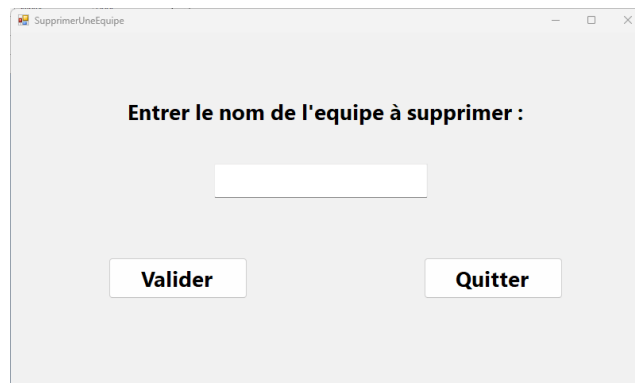
Si vous voulez ajouter une équipe à la base de données, il vous suffit d'appuyer sur le bouton prévu à cet effet dans la fenêtre d'accueil. Vous arrivez alors sur une nouvelle fenêtre dans laquelle, vous devez compléter les différents champs nécessaires à la création d'une nouvelle équipe. Vous pouvez ensuite appuyer sur « Valider » et la nouvelle équipe est inscrite dans la base de données.



The screenshot shows a window titled "NouvelleEquipe". It contains the following elements:

- A label "Nom de la nouvelle equipe :" followed by a text input field.
- Two labels in a row: "Nom du premier cavalier :" and "Nom du premier cheval :", each followed by a text input field.
- Two labels in a row: "Nom du deuxieme cavalier :" and "Nom du deuxieme cheval :", each followed by a text input field.
- Two buttons at the bottom: "Valider" and "Quitter".

Si vous avez commis une erreur lors de l'encodage de l'équipe, pas de soucis, vous pouvez la supprimer très simplement en appuyant sur le bouton « Supprimer une équipe » et en entrant le nom de l'équipe que vous voulez supprimer.

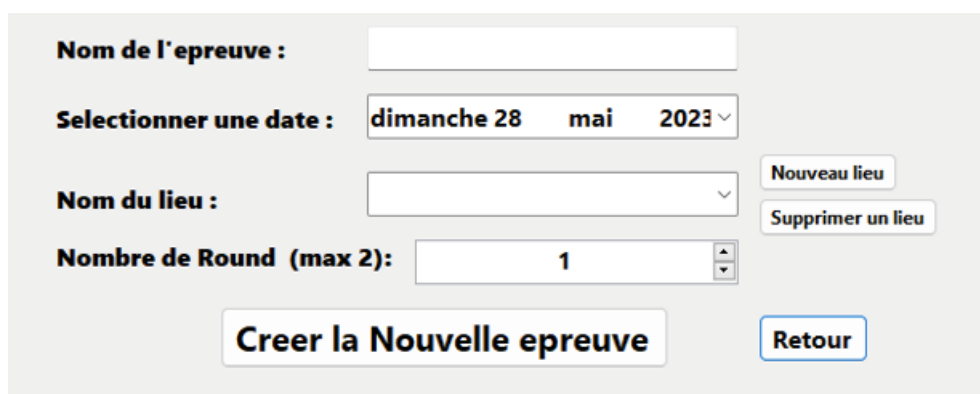


The screenshot shows a window titled "SupprimerUneEquipe". It contains the following elements:

- A label "Entrer le nom de l'equipe à supprimer :" followed by a text input field.
- Two buttons at the bottom: "Valider" and "Quitter".

Création d'une épreuve

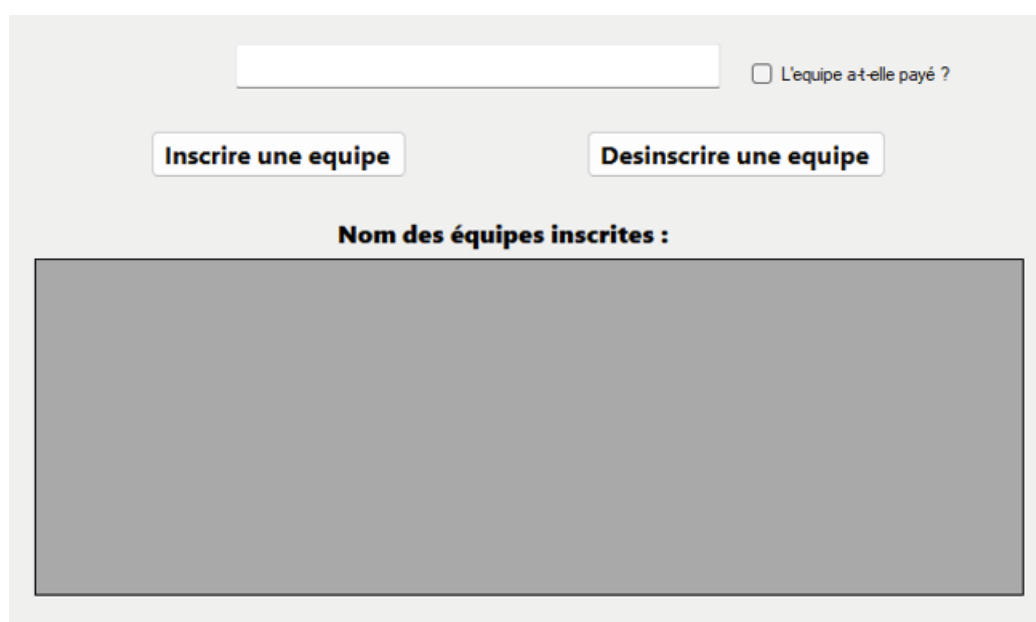
Pour créer une épreuve, ce n'est pas beaucoup plus compliqué, appuyez sur le bouton « Créer une nouvelle épreuve » et complétez les champs qui vous sont demandés. Pour le lieu de l'épreuve, vous pouvez choisir un lieu déjà existant dans la liste déroulante ou alors, vous pouvez ajouter un nouveau lieu grâce au bouton « Nouveau lieu » qui vous renverra sur une nouvelle fenêtre. Le principe est exactement le même que pour « Ajouter une équipe », vous devez remplir les champs demandés. Une fois toutes ces étapes terminées, appuyez sur le bouton « Créer la nouvelle épreuve ».



The form for creating a new race event contains the following elements:

- Nom de l'épreuve :** A text input field.
- Selectionner une date :** A date picker showing "dimanche 28 mai 2023" with a dropdown arrow.
- Nom du lieu :** A dropdown menu.
- Nombre de Round (max 2):** A numeric input field with the value "1" and a spinner.
- Buttons:** "Nouveau lieu" and "Supprimer un lieu" are located to the right of the location dropdown. "Créer la Nouvelle epreuve" and "Retour" are at the bottom.

Une fois l'épreuve créée, vous pouvez y inscrire des équipes en écrivant le nom de l'équipe dans la case blanche et indiquer si l'équipe a payé au moment de l'inscription ou non en cochant la case. Vous pouvez également désinscrire une équipe. Si vous voulez poursuivre les inscriptions ultérieurement, vous pourrez le faire via le bouton « Visualiser une épreuve existante ».



The form for managing race event teams contains the following elements:

- Search:** A text input field at the top.
- Payment Status:** A checkbox labeled "L'équipe a-t-elle payé ?".
- Buttons:** "Inscrire une equipe" and "Desinscrire une equipe" are located below the search field.
- Table:** A large table with the header "Nom des équipes inscrites :" and a greyed-out body.

Déroulement de l'épreuve

Pour lancer une épreuve, appuyez sur le bouton « Lancer une épreuve ». Vous remarquerez que ce bouton est toujours accessible donc vous pouvez très bien créer plusieurs épreuves et les lancer ultérieurement. Lorsque vous arrivez sur la fenêtre dédiée à l'arbitre, vous pouvez sélectionner une épreuve et le round. Le nom de l'équipe en jeu et celui de l'équipe suivante apparaitront une fois que vous aurez appuyé sur « Lancer ». Lorsque l'équipe est en place, vous pouvez appuyer sur « Go », un numéro aléatoire sera attribué et le jeu peut commencer. Chaque fois qu'une vache valide passe appuyez sur « Bonne vache ». Le jeu s'arrêtera soit à la fin du temps, soit lorsque les 10 vaches sont passées dans le bon ordre ou soit lorsqu'on appuie sur « Mauvaise vache ». Les résultats de l'équipe en cours sont affichés à gauche et le classement en bas de la fenêtre. Une fois le passage de l'équipe terminé, validez les résultats et vous pouvez passer à l'équipe suivante.

Round

Resultat de l'equipe :

Nom de l'equipe en cours :

Equipe en cours

Nom de l'equipe suivante :

Equipe suivante

00:00

Sélectionner l'épreuve à lancer

Sélectionner le numero du round

Lancer

Numero de la premiere vache : *Random vache*

GO

Bonne vache

Mauvaise vache

Valider les résultats

Equipe suivante

Retour à l'accueil

Classement :

Énumération des liens que nous avons utilisé durant nos recherches

Durant la réalisation du projet, nous avons utilisé les liens suivants. Pour plus de détails, veuillez consulter la partie « Explications complémentaires ».

Pour les requêtes : <https://www.w3schools.com/sql/default.asp>

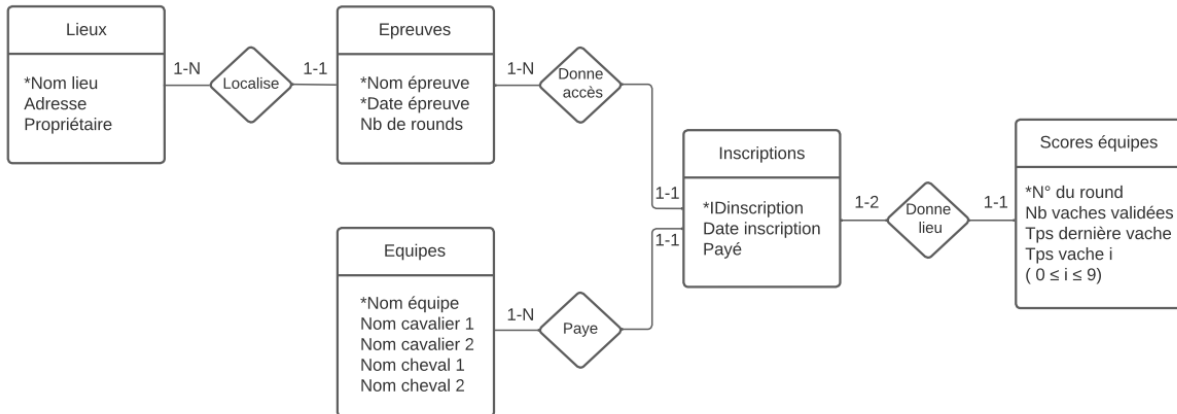
Pour toutes les nouvelles Windows Forms : <https://learn.microsoft.com/en-us/visualstudio/ide/create-csharp-winform-visual-studio?view=vs-2022>

Pour pouvoir travailler en collaboration : https://github.com/Mateo-lab/Projet_POO

Pour résoudre certains problèmes ou obtenir des conseils sur des aspects spécifiques de notre projet : <https://openai.com/blog/chatgpt>

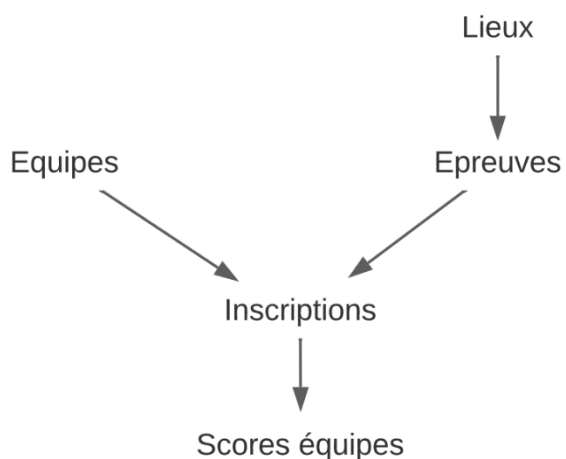
Schémas de modélisation orientés gestion de données

MCD : Modèle Conceptuel de Données ou modèle entités-associations



Ce modèle nous permet de bien comprendre comment les différentes tables sont liées entre elles. En effet, grâce à ce MCD, on peut voir qu'un lieu peut localiser une infinité d'épreuves mais qu'une épreuve ne peut être associée qu'à un seul lieu. Le raisonnement est le même pour les liens entre épreuves et inscriptions ainsi qu'entre équipes et inscriptions. Une inscription peut quant à elle donner lieu à 1 ou 2 scores équipes maximum puisque chaque équipe n'a droit qu'à 1 ou 2 rounds selon l'épreuve.

MLD : Modèle Logique de Données ou diagramme de Bachman



Le MLD permet de faire des liens entre les segments propriétaires et les segments membres. Ainsi, si on suit notre MLD, on peut dire qu'un lieu « possède » une ou des épreuve(s), une épreuve « possède » une ou des inscription(s), une équipe « possède » une ou des inscription(s) et une inscription possède un ou deux (voir MCD) score(s) d'une équipe.

MRD : Modèle Relationnel de Données

Lieux (**Nom lieu**, Adresse, Propriétaire)

PK – clé primaire

Epreuves (**Nom épreuve**, **Date épreuve**, Nombre de round, Nom lieu)

FK – clé étrangère

Equipes (**Nom équipe**, Nom cavalier 1, Nom cavalier 2, Nom cheval 1, Nom cheval 2)

Inscriptions (**IDinscription**, Nom épreuve, Date épreuve, Nom équipe, Date inscription, Payé)

Scores équipes (IDinscription, Nom équipe, **N° round**, Nb vaches validées, Tps dernière vache, Tps vache i ($0 \leq i \leq 9$))

Ce modèle nous permet de mettre en avant les clés primaires et les clés étrangères.

La clé primaire permet d'identifier de manière unique les différents enregistrements qui composent une table. Pour ce faire une clé primaire peut être composée d'un seul champ comme c'est le cas pour l'entité « Lieux » mais elle peut aussi être composée de plusieurs champs comme pour l'entité « Scores équipes ». Un lieu est rendu unique par le nom du lieu tandis qu'un score équipe est rendu unique par l'IDinscription et par le numéro du round.

On parle de clé étrangère lorsqu'une clé primaire d'une entité est présente dans une autre entité.

Explications complémentaires

Dans le cadre de notre projet de programmation "Ranch Sorting" en C#, nous avons acquis plusieurs compétences et connaissances importantes. Voici la liste des choses que nous avons apprises :

1. Utilisation de la Classe Timer : nous avons appris comment utiliser la classe Timer pour gérer le temps dans le programme. Cela nous a permis d'implémenter des fonctionnalités liées à la gestion du temps, comme la synchronisation des actions ou les délais entre les événements.
2. Gérer les erreurs : Bien que nous n'ayons pas eu suffisamment de temps pour nous occuper de toutes les erreurs potentielles, nous avons appris comment gérer les erreurs les plus courantes dans le projet. Cela inclut la gestion des exceptions et la mise en place de mécanismes de récupération en cas d'erreur.
3. Effectuer un bon débogage : Nous avons développé des compétences en matière de débogage pour trouver et résoudre les problèmes dans le code. Nous avons appris à utiliser les fonctionnalités de débogage de l'environnement de développement intégré (IDE) pour suivre l'exécution du programme, examiner les valeurs des variables et détecter les erreurs.
4. Créer des requêtes SQL : Nous avons acquis des connaissances sur la création de requêtes SQL pour interagir avec une base de données. Nous avons appris à écrire des requêtes pour récupérer, insérer, mettre à jour et supprimer des données dans une base de données relationnelle.
5. Gérer des événements MessageBox : Nous avons appris à gérer les événements MessageBox, qui sont des boîtes de dialogue modales affichant des messages à l'utilisateur. Nous avons pu implémenter des fonctionnalités telles que l'affichage de messages d'erreur, de notifications ou de confirmations dans notre programme.
6. Utilisation de ChatGPT : Pour résoudre certains problèmes ou obtenir des conseils sur des aspects spécifiques de notre projet, nous avons exploré l'utilisation de ChatGPT. Cette plateforme de conversation basée sur l'intelligence artificielle nous a permis d'obtenir des réponses rapides et d'approfondir mes connaissances en échangeant avec un modèle de langage avancé.
7. Utilisation de GitHub : Lors du développement de notre projet, nous avons découvert l'importance de l'utilisation de GitHub pour gérer efficacement le code source. Nous avons appris à créer un dépôt, à effectuer des commits pour enregistrer les modifications apportées à notre code, et à collaborer en utilisant les fonctionnalités de gestion de version offertes par GitHub.

Il est important de noter que dans ce projet, nous avons rencontré des difficultés pour déterminer où s'arrêter en termes de fonctionnalités à implémenter, car il y avait toujours

plus de choses à ajouter. Nous avons donc pris la décision de nous concentrer sur ce qui nous semblait le plus important et de prioriser les fonctionnalités essentielles pour atteindre les objectifs du projet.

Nous espérons que cette liste des compétences acquises dans le cadre de notre projet de programmation répond à vos attentes pour le rapport.