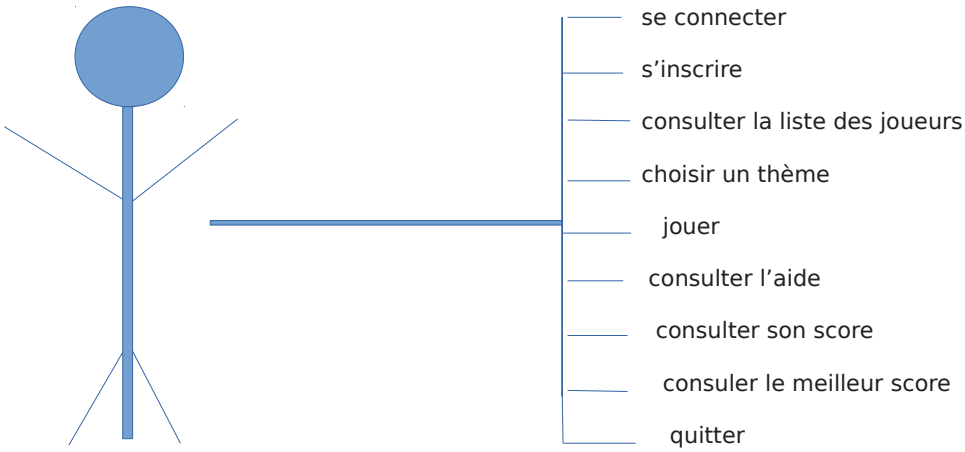
1. Introduction :

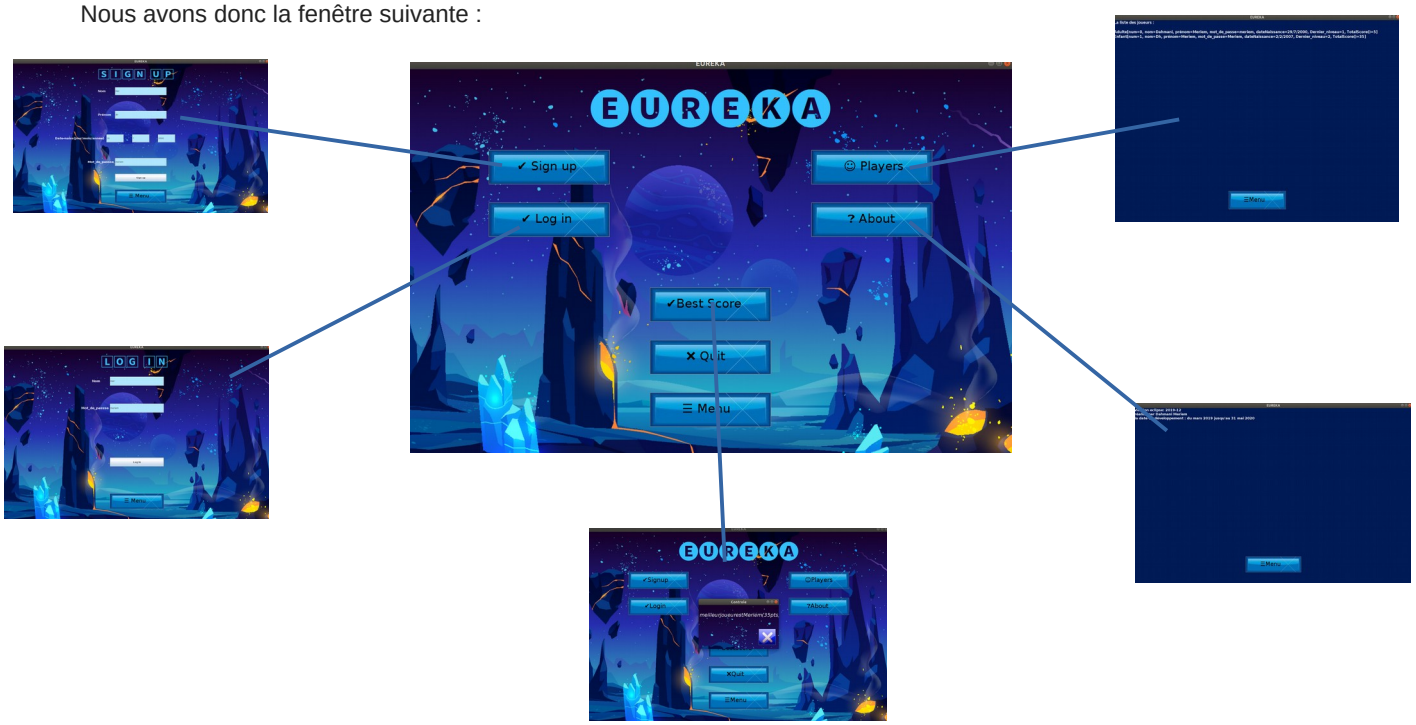
L'objectif principal de notre projet est de réaliser un jeu de pendu qui est ludique et amusant en même temps .Il consiste à poser des questions à l'utilisateur, puis, à lui de trouver la réponse en devinant quelles sont les lettres qui la composent.Si le joueur arrive à trouver la réponse , il passe au nivau suivant,sinon il reste dans le même niveau.

2. Architecture de la solution :

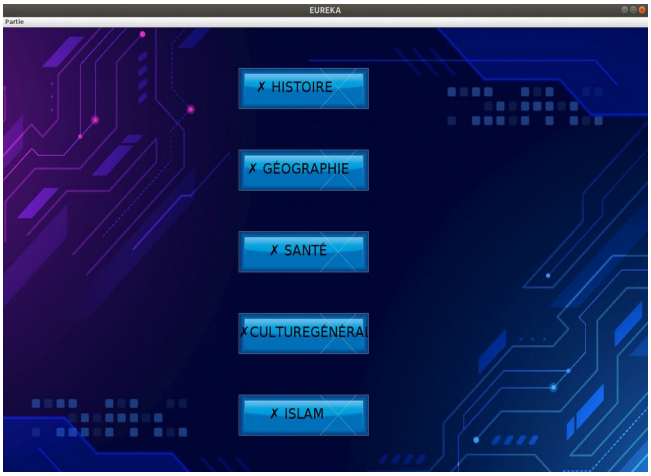
Diagramme des cas d'utilisations :



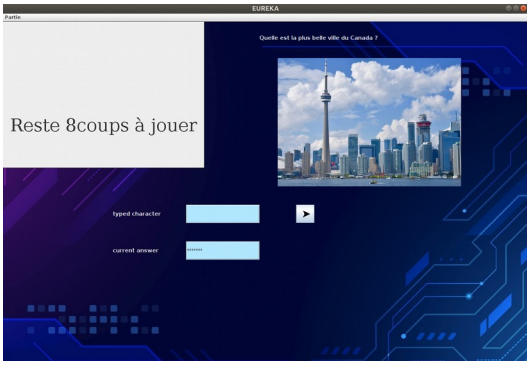
Lorsque le jeu débute nous sommes sur la page principale du jeu qui contient 6 boutons principaux ,elle permet au joueur de s'inscrire(Sign up) , ou de se connecter (Log in) s'il est déjà inscrit.Lors de l'inscription , tous les champs doivent être remplis, et la date doit être valide , sinon une fenêtre d'alerte s'affiche.le joueur peut aussi consulter la liste des joueurs ,ou les informations à propos , ou bien retourner au menu principal.
Nous avons donc la fenêtre suivante :



Une fois le joueur est connecté, il peut choisir un thème pour commecer une nouvelle partie , celle-ci a le dernier niveau atteint si le joueur a déjà réalisé d'autres parties, sinon on entame le premier niveau.



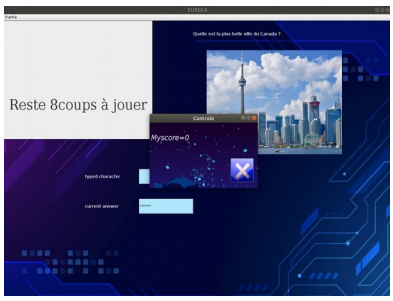
Quand on lance un nouvelle partie, une question et une image seront affichées à l'utilisateur, ainsi que deux zones de saisie ,la première zone est utilisée pour la saisie des caractères devinés par le joueur, et la seconde zone est initialisée avec des étoiles pour contenir les caractères trouvés au fur et à mesure qu'ils sont tapés durant le jeu. Une dernière zone est utilisée pour afficher une partie du dessin du pendu et sa potence.

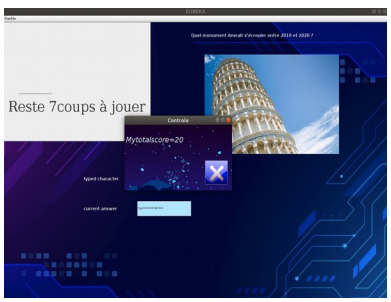


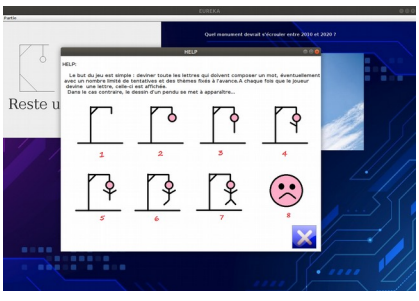
Lorsque le joueur tape un caractère sur la première zone, il est affiché à sa position dans la seconde zone s'il appartient au mot de la réponse. Sinon, une partie du dessin du pendu et sa potence sont affichés dans la zone à gauche, et le nombre de tentatives restant diminue.

Si le nombre de tentatives restant arrive à 0, le joueur est considéré comme perdant de la partie ,sinon il trouve la réponse et il peut passer au nivau suivant.

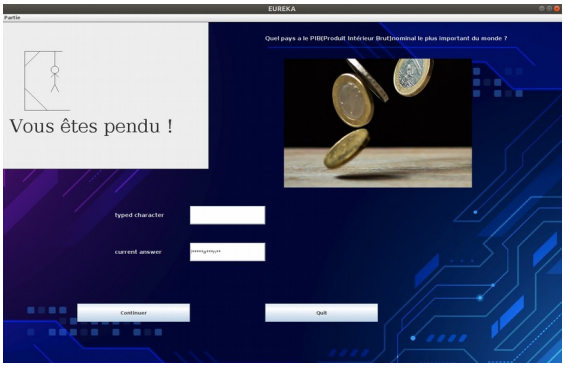
Pendant la partie ,le joueur peut afficher son scoreCourant (score), qui représnete le nombre de points cumulé depuis la dernière connexion , son Score totale (total score), qui représente le nombre de point totale cumulé depuis la première partie du joueur.Comme il peut aussi afficher l'aide .

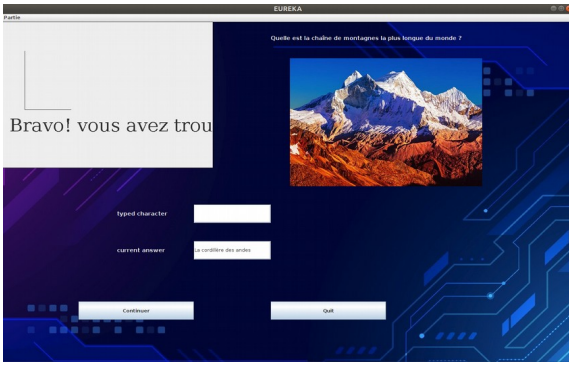




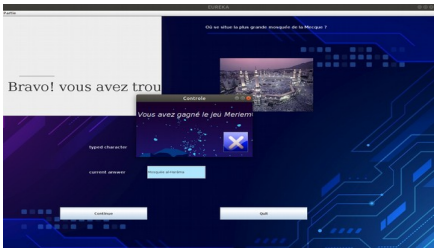


Une fois la partie est terminée , le joueur peut continuer (passer au niveau suivant ou rester dans le meme niveau) et choisir un autre thème pour la partie suivante, ou bien quitter le jeu .



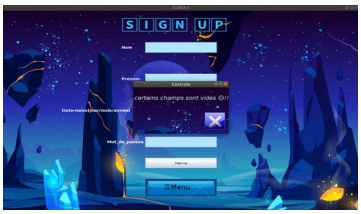


Si le joueur gagne les 5 niveaux , une fenetre sera affichée.



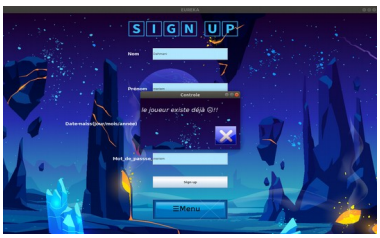
Quand le joueur quitte la partie , il revient au menu principal, et son objets sera modifié.On peut le vérifier on consultant la liste des joueur .
On peut quitter l'app , ou bien s'inscrire ou se connecter….etc.

Les controles:
*Durant l'inscription ou la connexion , si certains champs ne sont pas remplis.

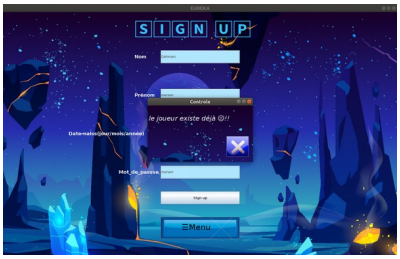


*De plus , si la date de naissance n'est pas valide.

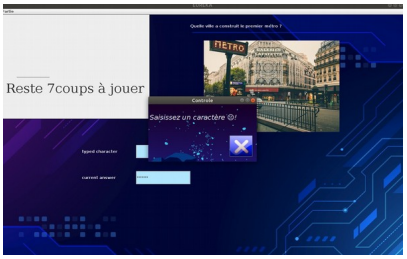
*si un joueur avec le meme nom et mot de passe existe déjà.



*si le joueur qui essaye de se connecter n’existe pas.



*Si le joueur ne saisie aucun caractère, ou il saisie plusieurs caractères.



les données (fichiers) en entrée à l'application:

*Le fichier joueurs.txt qui contient des objets de type joueur, on l’utilise pour charger la structure HashMap.
*Le fichier questions.txt qui contient des objets de type Question , on l’utilise pour charger la structure ArrayList.

les structures de données globales utilisées:

La classe EUREKA:c’est une une classe exécutable qui commence par charger les sructures hashmap(pour les joueurs) et hashset(pour les thèmes) à partir des fichiers,de plus,elle ouvre la fenêtre principale, et afficher le menu ,en utilisant la méthode menu(), pour permettre au joueur de s’inscrire(Sign up) en créant un nouvel objet Joueur (enfant ou adulte,selon la date de naissance),et l’ajouter au hashmap , ou de se connecter (Log in) s’il est déjà inscrit.On peut aussi afficher la liste des joueurs déjà inscrits en cliquant sur l’un des botons principaux(players).Elle permet aussi d’afficher le joueur qui a le meuilleur score(best score)

la classe Play:cette classe propose 5 thèmes principaux , en affichant 5 boutons , ce qui permet au joueur de choisir un thème et lancer une nouvelle partie(la méthode play),en affichant une question et une image , ainsi que deux zones de saisie ,la première zone est utilisée pour la saisie des caractères devinés par le joueur, et la seconde zone est initialisée avec des étoiles pour contenir les caractères trouvés au fur et à mesure qu’ils sont tapés durant le jeu. Une dernière zone est utilisée pour afficher une partie du dessin du pendu et sa potence.

la classe ThèmeJeu:cette classe énum,elle permet d’instancier les 5 thèmes principaux.

la classe PartieJeu: La classe PariteJeu crée un objet de type partie,de plus, elle vérifie si un caratère donnée appartient à la réponse de la la question courante.

la classe joueur:cette classe permet de créer un objet de type joueur, c’est une classe abstraite ,avec 2 autres classes filles, elle permet de lancer un nouvelle partie après le choix des questions selon leurs thème et leurs type.

la classe Adulte:C’est une classe fille de la classe abstrait joueur,elle permet d’inisialiser un objet de type joueur adulte,et de choissir les questions d’un adulte correspondante à un thème choisi , à partir du set.

la classe Enfant:C’est la classe fille de la classe abstrait joueur,elle permet d’inisialiser un objet de type joueur enfant,et de choissir les questions d’un enfant correspondante à un thème choisi, à partir du set.

la classe Question: cette classe permet de crée un objet de type question , avec un niveau et un thème précis.

la classe QuestionAdulte:C'est une classe fille de la classe Question, cette dernière permet d'instancier des questions pour un joueur adulte.

la classe QuestionEnfant:C'est une classe fille de la classe Question, cette dernière permet d'instancier des questions pour un joueur enfant.

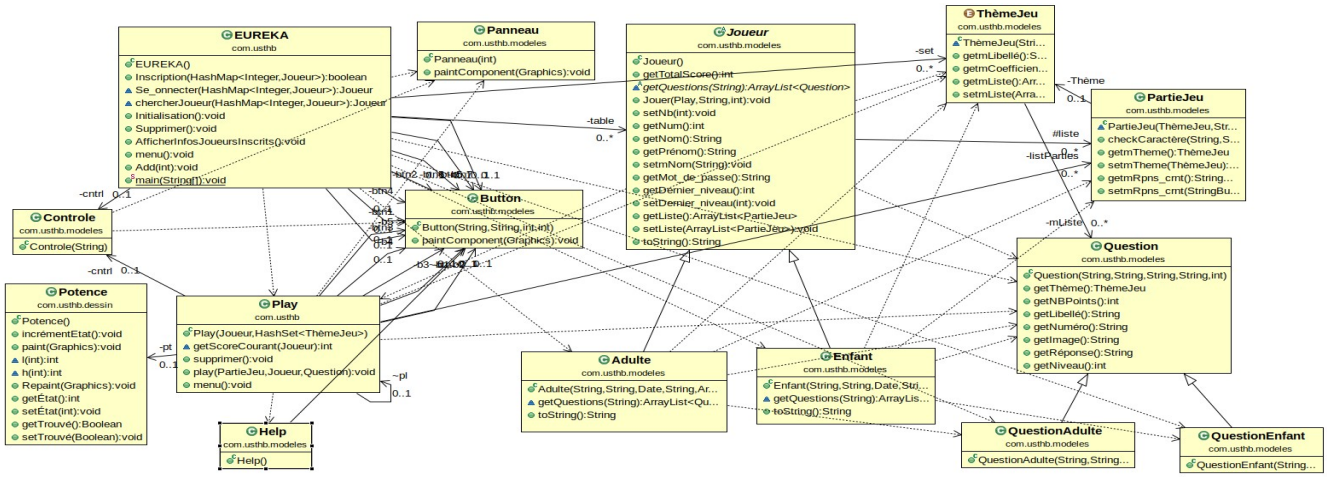
la classe Potence: elle permet d'afficher le dessin de la potence selon le nombre de chances restant .

la classe paneau: elle permet de personnaliser les fenêtres du jeu, en ajoutant une image en arrière plan.

la classe help: elle permet d'afficher l'aide en cliquant sur le bouton help, et de quitter cette fenetre en cliquant sur un bouton .

la classe controle: elle permet d'afficher une erreur de saisie lors de l'inscription ou de la connexion , ou bien la fin des 5 niveaux , on peut la quitter en cliquant sur un bouton .

la classe Button: elle permet de personnaliser les boutons du jeu , en ajoutant une image en arrière plan,et un texte personnalisé.



Les structures de données globales utilisées:

*Hashmap contenant les joueurs.
*Hashset contenant les thèmes.
*ArrayList contenant les questions d'un thème donné.
*ArrayList contenant les parties réalisées par le joueur.

Outils de développement utilisés:

visual studio code et eclipse.

Conclusion :

En ce qui concerne l'amélioration de l'application , je propose:
* Ajouter un chronomètre à la partie , et si le joueur ne trouve pas la réponse avant la fin du temps précis, il est considéré comme un perdant de la partie.
*trouver une lettre qui a appartient à la réponse courante, permet de recharger le chronomètre
*Choisir la prochaine question en prenant en considération le score du joueur , se qui définit ses capacités.
*Le score du joueur ne dépand pas uniquemet du thème,et du niveau de la questions , mais aussi du nombre d'erreurs.

A l'aide de ce projet j'ai pu comprendre et expérimenter les différentes étapes de la conception d'un projet.De plus la programmation m'a permis d'améliorer mes connaissances du langage java.