

# Rapport sur l'Application de Quiz

## Introduction :

Ce projet est une application de quiz développée en Java Swing, proposant deux types de questions : "CS" pour l'informatique et "funny" pour des questions amusantes. L'application permet aux utilisateurs de répondre à des questions sous forme de QCM avec un timer pour chaque question. À la fin du quiz, le score est affiché sur une page de victoire ou de défaite. Ce projet vise à offrir une expérience interactive et éducative, en testant les connaissances techniques des utilisateurs en informatique et en intégrant des éléments de divertissement.

## Team Members:

- *Tahraoui Bochra* : Responsable du développement du code et de la recherche des questions.
- *Zemmour Litissia* : Responsable du développement du code et de la conception des fonctionnalités.

## Fonctionnalités :

Chaque fonctionnalité est décrite ci-dessous avec ses objectifs, son implémentation et son fonctionnement :

### • Page d'Accueil :

- **Description** : Une page d'accueil accueillante avec un bouton de démarrage bien visible pour lancer le quiz.
- **Implémentation** : Conçue en utilisant les panneaux Java Swing.
- **Fonctionnalité** : Lorsque l'utilisateur clique sur le bouton de démarrage, il est redirigé vers le choix des catégories du quiz

## Page d'Accueil de l'Application de Quiz :



*Home page.*

### • **ChoosingPage :**

#### • **Description :**

La ChoosingPage permet aux utilisateurs de naviguer entre différentes catégories de quiz telles que les questions de codage, les quiz amusants, et des informations sur l'application (About). Chaque catégorie est accessible via un bouton spécifique sur l'interface.

#### • **Implémentation :**

Les boutons sont créés avec Java Swing, utilisés pour rediriger l'utilisateur vers différentes pages de l'application. Les couleurs et les polices sont définies via des propriétés de style pour améliorer l'expérience utilisateur. Les ActionListeners sont intégrés à chaque bouton pour gérer les interactions et rediriger l'utilisateur vers les pages correspondantes (AboutPage, Funnyquiz, QuestionPage).

#### • **Fonctionnalité :**

Permet de sélectionner une catégorie de quiz, suit les réponses de l'utilisateur pour générer un score final, et offre un accès facile aux informations sur l'application. Les boutons incluent des actions qui déplacent l'utilisateur vers les pages appropriées, offrant une navigation fluide et intuitive dans l'application.

### **Page des choix de l'Application de Quiz :**



*Choosingpage*

- **Page de Résultats (Victoire/Défaite)**

- **Description :**

Affiche le résultat basé sur le score. Si l'utilisateur obtient plus de 5 points, une page de succès s'affiche, sinon une page de consolation apparaît. Des options permettent de recommencer ou de quitter.

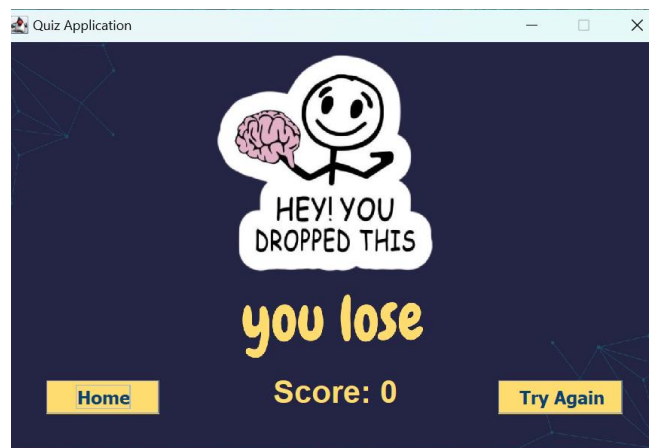
- **Implémentation :**

Utilise `CardLayout` pour afficher dynamiquement la page de succès ou de défaite. Le score est calculé et affiché pour déterminer l'écran approprié. Des boutons interactifs permettent de relancer le quiz ou de fermer l'application.

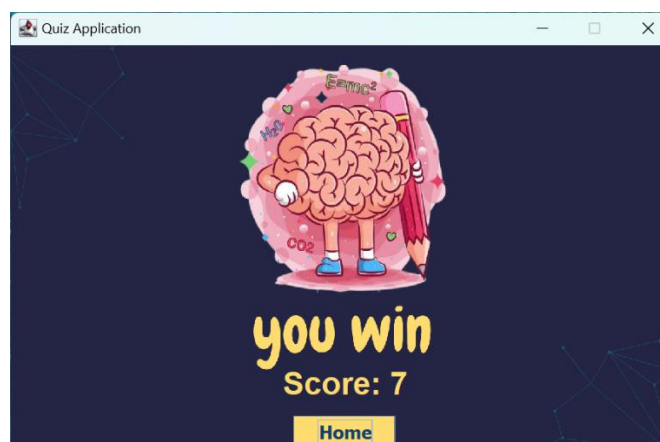
- **Fonctionnalité :**

Vérifie si le score dépasse 5 pour afficher la victoire. Propose une navigation simple pour recommencer ou quitter.

### **Page des (Victoire/Défaite) de l'Application de Quiz :**



*LosePage.*



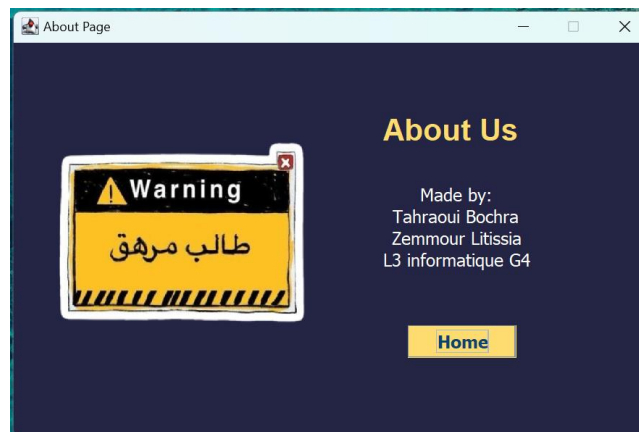
*WinPage.*

## Page About :

**Description :** Fournit des informations sur les développeurs.

**Implémentation :** Créée avec Java Swing, cette page affiche du texte structuré et stylisé pour une lecture claire. Des boutons permettent de revenir à la page d'accueil ou d'explorer d'autres fonctionnalités.

## Page About de l'Application de Quiz :



*AboutPage.*

## Question Page

### Description :

Cette page permet aux utilisateurs de répondre à une série de questions chronométrées. Les questions couvrent différents sujets informatiques avec des options multiples (QCM). Un chronomètre pour chaque question et une fonctionnalité d'évaluation en temps réel enrichissent l'expérience interactive.

### Implémentation :

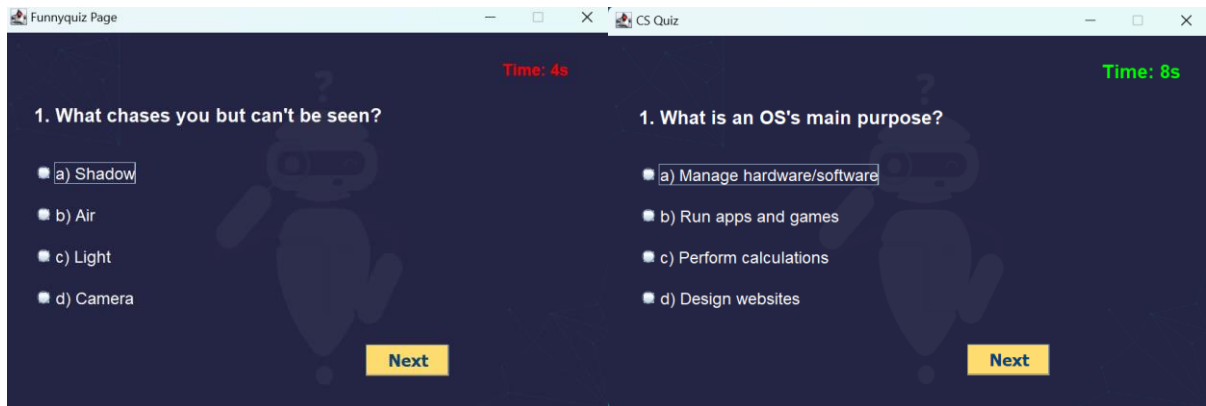
- **Gestion des questions :** Les questions et options sont stockées dans des tableaux, facilitant leur gestion et modification. Chaque question est liée à une réponse correcte vérifiée dynamiquement lors de la sélection.
- **Chronomètre :** Un Timer Swing décompte 10 secondes pour chaque question. Lorsque le temps s'écoule, l'application passe automatiquement à la question suivante. Le changement de couleur du texte (vert à rouge) alerte visuellement les utilisateurs à l'approche de la fin du temps.
- **Navigation :** Un bouton "Next" permet de soumettre la réponse et de passer à la question suivante. Une fois toutes les questions terminées, la page affiche un écran de victoire ou de défaite en fonction du score obtenu.

### Fonctionnalité :

- Évalue la réponse sélectionnée par l'utilisateur et suit le score en comparant avec les réponses correctes.
- Fournit un score final basé sur les bonnes réponses. Si le score atteint 5 ou plus, une page de victoire s'affiche ; sinon, une page de défaite est montrée.

- Offre une expérience utilisateur immersive grâce à un arrière-plan personnalisé et un chronomètre intégré.

## Page de Question de l'Application de Quiz :



*QuestionPage.*

## Défis et solutions :

- **Affichage du chronomètre :** La gestion du chronomètre a nécessité l'ajout d'un Timer pour chaque question. Le changement de couleur dynamique a été implémenté pour améliorer l'ergonomie.
- **Gestion des questions via tableaux :** Les tableaux ont simplifié le stockage et la récupération des questions/réponses, mais leur maintenance pourrait devenir complexe pour un plus grand volume de questions.
- **Transition entre les questions :** Un autre défi a été la gestion de la transition entre les questions, particulièrement lors du changement d'une question à une autre avec un délai chronométré. La mise en œuvre d'un ActionListener associé à un bouton "Next" permet de capturer la réponse sélectionnée, puis de vérifier la réponse et de charger dynamiquement la prochaine question tout en redémarrant le chronomètre. Cela garantit une expérience fluide pour l'utilisateur sans perte de temps.

## Quelque image de la Solution :

```

private void checkAnswer() {
    String selectedAnswer = "";
    for (int i = 0; i < 4; i++) {
        if (radioButton[i].isSelected()) {
            selectedAnswer = radioButton[i].
getText().substring(0, 1);
// get the first letter 'a', 'b', 'c', or 'd'
        }
    }
    if (selectedAnswer.equals(
correctAnswersArray[current])) {
        correctAnswers++;
    }
}

private void showNextQuestion(int direction) {
    current += direction;

    if (current < 0) {
        current = 0;
        return;
    } else if (current >= questions.length) {
        current = questions.length - 1;
        displayResult();
        return;
    }

    label.setText(questions[current]);
    for (int i = 0; i < 4; i++) {
        radioButton[i].setText(options[current
][i]);
    }
    bg.clearSelection();
    startTimer();
}

public int getCorrectAnswersCount() {
    return correctAnswers;
}

private void displayResult() {
    if (correctAnswers >= 5) {
        new Winpage(getCorrectAnswersCount
()).setVisible(true);
    } else {
        new LosePage(getCorrectAnswersCount
()).setVisible(true);
    }
    dispose();
}
}

```

```

private String[] questions = {
    "1. What is an OS's main purpose?",
    ,
    "2. What does 'HTML' stand for?",
    "3. Which structure uses LIFO?",
    "4. Time complexity for searching in BST?",
    "5. What is inheritance in OOP?",
    "6. Which is NOT a programming language?",
    "7. Algorithm for shortest path in a graph?",
    "8. What does SQL mean?",
    "9. Function of DNS?",
    "10. Example of a NoSQL database?"
};

private String[][] options = {
    {"a) Manage hardware/software",
    "b) Run apps and games",
    "c) Perform calculations",
    "d) Design websites"},
    {"a) Hyper Text Markup Language",
    "b) High-level Machine Language",
    "c) Hyperlink Manipulation Language",
    "d) Home Tool Markup Language"},
    {"a) Queue", "b) Stack",
    "c) Linked list", "d) Array"},
    {"a) O(1)", "b) O(log n)",
    "c) O(n)", "d) O(n log n)"},
    {"a) Create objects from others",
    "b) Restrict access to members",
    "c) Hide complexity of operations",
    "d) Inherit properties and methods"},
    {"a) Python", "b) JavaScript",
    "c) Java", "d) InternetScript"},
    {"a) QuickSort", "b) MergeSort",
    "c) Dijkstra's Algorithm", "d) Binary Search"
},
    {"a) Structured Query Language",
    "b) Simple Query Language",
    "c) Sequential Query Logic",
    "d) Simple Query Logic"},
    {"a) Assign IP addresses",
    "b) Convert domain names to IPs",
    "c) Send data packets",
    "d) Encrypt web traffic"},
    {"a) MySQL", "b) MongoDB",
    "c) Oracle Database", "d) PostgreSQL"}
};

private String[] correctAnswersArray =
{
    "a", "a", "b", "b", "d", "d", "c"
, "a", "b", "b"
};

```

```

private void startTimer() {
    timeLeft = 10;
    // Reset the time for each question
    timerLabel.setText("Time: " + timeLeft
+ "s");

    timer = new Timer(1000, new
ActionListener() {
        @Override
        public void actionPerformed(
ActionEvent e) {
            timeLeft--;
            timerLabel.setText("Time: " +
timeLeft + "s");

            if (timeLeft <= 0) {
                timer.stop();
                showNextQuestion(1);
            }
            if (timeLeft < 5) {
                timerLabel.setForeground(
Color.RED);
            } else {
                timerLabel.setForeground(
Color.GREEN);
            }
        }
    });
    timer.start();
}

```

*les solutionproposer*

## Conclusion

Le projet a réussi à créer une application de quiz interactive qui offre une expérience utilisateur engageante. Les fonctionnalités comme le suivi du chronomètre, la gestion dynamique des questions via des tableaux, et la présentation visuelle de "victoire/défaite" enrichissent l'expérience utilisateur. Les défis rencontrés, tels que l'affichage du chronomètre et la transition fluide entre les questions, ont été résolus par l'utilisation efficace de `Timer` et `CardLayout`. Les leçons apprises incluent l'importance de la gestion des données et de l'interaction utilisateur pour améliorer l'ergonomie et l'engagement.