









## 1 INFORMATIONS GENERALES

Candidat :	<b>KING</b>	<b>ALEXANDRE SAMUEL</b>																				
	 <a href="mailto:alexandre.king@eduvaud.ch">alexandre.king@eduvaud.ch</a>	 : 078 860 72 24																				
Lieu de travail :	<input type="checkbox"/> ETML, Rue de Sébeillon 12, 1004 Lausanne <input checked="" type="checkbox"/> CFPV (COFOP-ETML), Avenue de Valmont 28b, 1010 Lausanne																					
Orientation :	<input checked="" type="checkbox"/> 88601 Développement d'application <input type="checkbox"/> 88602 Informatique d'entreprise <input type="checkbox"/> 88603 Technique des systèmes																					
Chef de projet :	Curchod	Aurélie																				
	 : <a href="mailto:aurelie.curchod@eduvaud.ch">aurelie.curchod@eduvaud.ch</a>	 : 021 338 22 40																				
Expert 1 :	Nom : Glassey	Prénom : Nicolas																				
	 : <a href="mailto:nicolas.glassey@eduvaud">nicolas.glassey@eduvaud</a>	 : 024 557 60 70																				
Expert 2 :	Nom : Borboën	Prénom : Nicolas																				
	 : <a href="mailto:nicolas.borboen@epfl.ch">nicolas.borboen@epfl.ch</a>	 : +41 76 615 25 80																				
Période de réalisation :	Du <b>lundi 29 avril 2024 à 8h</b> au <b>mercredi 29 mai 2024 à 16h35</b>																					
Horaire de travail :	<table border="0"> <tr> <td>Lundi</td> <td>08h00-12h15</td> <td>13h10-15h45</td> <td><i>Pentecôte 20 mai</i></td> </tr> <tr> <td>Mardi</td> <td>-</td> <td></td> <td></td> </tr> <tr> <td>Mercredi</td> <td>08h00-11h25</td> <td>13h10-16h35</td> <td></td> </tr> <tr> <td>Jeudi</td> <td>-</td> <td>13h10-16h35</td> <td><i>Ascension 9 mai</i></td> </tr> <tr> <td>Vendredi</td> <td>08h00-12h15</td> <td>13h10-16h35</td> <td><i>Pont de l'Ascension 10 mai</i></td> </tr> </table> <p><i>Toutes les demi-journées ont une pause obligatoire de 15 minutes.</i></p>		Lundi	08h00-12h15	13h10-15h45	<i>Pentecôte 20 mai</i>	Mardi	-			Mercredi	08h00-11h25	13h10-16h35		Jeudi	-	13h10-16h35	<i>Ascension 9 mai</i>	Vendredi	08h00-12h15	13h10-16h35	<i>Pont de l'Ascension 10 mai</i>
Lundi	08h00-12h15	13h10-15h45	<i>Pentecôte 20 mai</i>																			
Mardi	-																					
Mercredi	08h00-11h25	13h10-16h35																				
Jeudi	-	13h10-16h35	<i>Ascension 9 mai</i>																			
Vendredi	08h00-12h15	13h10-16h35	<i>Pont de l'Ascension 10 mai</i>																			
Nombre d'heures :	88 heures																					
Planning (en H ou %)	Analyse 20%, Implémentation 45%, Tests 10%, Documentation 25%																					
Présentation :	Dates retenues : 6 ou 7 juin 2024																					

## 2 PROCÉDURE

Le candidat réalise un travail personnel sur la base d'un cahier des charges reçu le 1<sup>er</sup> jour.

Le cahier des charges est approuvé par les deux experts. Il est en outre présenté, commenté et discuté avec le candidat. Par sa signature, le candidat accepte le travail proposé.

Le candidat a connaissance de la feuille d'appréciation avant de débiter le travail.

Le candidat est entièrement responsable de la sécurité de ses données.

En cas de problèmes graves, le candidat avertit au plus vite les deux experts et son CdP.

Le candidat a la possibilité d'obtenir de l'aide, mais doit le mentionner dans son dossier.

A la fin du délai imparti pour la réalisation du TPI, le candidat doit transmettre par courrier électronique le dossier de projet aux deux experts et au chef de projet. En parallèle, une copie papier du rapport doit être fournie sans délai en trois exemplaires (L'un des deux experts peut demander à ne recevoir que la version électronique du dossier). Cette dernière doit être en tout point identique à la version électronique.

---

### 3 TITRE

CodeTris : Le bloc de la programmation

---

### 4 MATÉRIEL ET LOGICIEL À DISPOSITION

- Un PC standard de l'ETML (Windows 10)
- Visual Studio Community 2022
- Suite Microsoft Office pour la documentation
- Un dépôt Git (GitHub, BitBucket ou autre)

---

### 5 PRÉREQUIS

- Connaissances en programmation (Modules ICT 403, 404, 226, 120, 326, 411, 133)

---

### 6 DESCRIPTIF DU PROJET

#### 6.1 Introduction

CodeTris est un jeu éducatif qui intègre des questions de programmation dans le jeu d'arcade Tetris.

L'objectif de ce projet est de créer une expérience ludique et éducative, encourageant les joueurs à renforcer leurs compétences en programmation tout en jouant à un jeu classique.

Ce jeu pourrait être utilisé par exemple comme accroche lors des modules de programmation de 1<sup>ère</sup> année.

#### 6.2 Fonctionnalités

##### 6.2.1 Jeu Tetris

La première fonctionnalité à implémenter est un jeu de Tetris classique à l'aide de C# en mode Console. Le jeu doit pouvoir être joué avec les flèches du clavier ou via les touches « WASD »<sup>1</sup>

Les pièces (formes standard) tombent les unes après les autres et le joueur peut :

- Les faire pivoter (flèche du haut)
- Les déplacer (flèches gauche / droite)
- Faire descendre la pièce plus rapidement (flèche bas)
- Faire tomber la pièce directement (touche espace)

L'objectif est de continuer à jouer jusqu'à ce qu'une pièce ne puisse plus être placée sans sortir de la zone de jeu (haut).

Lorsqu'une ligne est complétée, traditionnellement cela donne des points au joueur et augmente la vitesse de jeu.

Le jeu « CodeTris » déclenche l'interrogation d'une question de programmation au joueur à ce moment-là.

---

<sup>1</sup> A définir dans les options de jeu

### 6.2.2 Questions de programmation

Lorsqu'une ligne est complétée, le jeu est mis en pause et le joueur doit répondre à une question liée à la programmation (en C#), par exemple :

« Comment s'appelle le type de donnée qui permet de gérer les chaînes de caractères »

L'utilisateur doit alors écrire la réponse puis valider avec la touche enter.

Si la réponse est correcte, on l'indique au joueur puis un certain nombre de points (à définir) est ajouté au score.

Si celle-ci est erronée, on l'indique au joueur (avec la réponse attendue) une ligne supplémentaire « malus » est ajoutée en bas de l'écran de jeu. Cela complique la suite du jeu car elle ne peut pas être supprimée. Cette ligne devra être visuellement distincte des autres.

Ces questions seront stockées dans une base de données MySQL.

### 6.2.3 Fin de la partie

Lorsque l'écran est complètement rempli, et qu'une pièce sort de la zone de jeux, la partie se termine.

Le score est alors affiché au joueur et celui-ci est enregistré dans la DB (avec la date et un pseudo).

L'indication du nombre de réponses justes / fausses aux questions est également affichée.

### 6.2.4 Règles du jeu

L'application devra proposer une page d'aide à l'utilisateur lui permettant de comprendre comment jouer.

### 6.2.5 Options

Le jeu devra proposer un certain nombre d'options :

- Niveau des questions : trois niveaux de questions différents seront proposés au joueur
- Jouabilité : deux modes de jeux, flèches ou « WASD »

### 6.2.6 Scores

Les 10 meilleurs scores (par niveau de difficulté) stockés dans la DB seront affichés sur cette page, avec le pseudo du joueur ainsi que la date de réalisation.

### 6.2.7 Ergonomie des interfaces

Aucune contrainte n'est imposée sur l'ergonomie des interfaces.

Des maquettes d'écran devront être réalisées dans l'analyse.

Les critères UX suivants devront être respectés :

- **Clarté et lisibilité** : Utiliser une police appropriée et des couleurs de texte et d'arrière-plan contrastées.
- **Simplicité** : Eviter les fonctionnalités superflues et ne présenter à l'utilisateur que les options nécessaires.
- **Gestion des erreurs** : Fournir à l'utilisateur des messages d'erreur explicites.
- **Consistance** : Maintenir une cohérence dans la manière dont les informations sont présentées et dans les commandes utilisées.
- **Optimisation des performances** : L'application console doit répondre rapidement aux commandes de l'utilisateur.

### 6.3 Tests

Des tests fonctionnels devront être définis dans la conception de l'application et le résultat de leur exécution devra être documenté dans la partie réalisation.

### 6.4 Sauvegarde et gestion de versions

Le code source devra être versionné dans un dépôt Glit (GitHub, BitBucket ou autre). L'accès à ce dépôt devra être fourni au chef de projet ainsi qu'aux deux experts.

### 6.5 Technologies

L'application devra être réalisée en C# en mode console.

Deux éléments devront être stockés dans une base de données MySQL :

- Les questions / réponses de programmation (par niveau de difficulté)
- Les scores (date, pseudo du joueur et niveau de difficulté)

Un fichier de configuration permettra de configurer l'accès à la DB.

---

## 7 LIVRABLES

Le candidat est responsable de livrer à son chef de projet et aux deux experts :

- Une planification initiale
- Un rapport de projet contenant au minimum :
  - L'analyse du projet (un ou des schémas de principe sont souhaitables pour illustrer les réflexions rédigées).
  - L'analyse devra contenir des informations sur les objectifs, le contenu et le public cible avec les conséquences.
  - La réalisation répondant au cahier des charges (recherches de solutions, fonctionnalités, argumentation sur les choix effectués, obstacles rencontrés, mentions des aides extérieures apportées, etc.), ainsi qu'une charte graphique et une maquette du design. Les schémas de base de données (MCD, MLD, MPD) sont aussi attendus.
  - Les tests planifiés et effectués avec les résultats et analyses de ceux-ci.
  - Un résumé du projet (1 page) qui décrit de manière succincte les enjeux du TPI
  - La conclusion contenant, en outre, des considérations personnelles, des considérations techniques et des considérations de réalisation du projet (comparaison entre ce qui devait être fait et ce qui a été réellement fait, etc.).
- Le journal de travail avec mentions de ce qui a été accompli, les suites à donner chaque fois qu'elles sont envisagées, liens et références des informations collectées ou retranscrites.
- Le code source de l'application (liens vers le dépôt Git fourni au CDP et aux experts)
- Un manuel technique permettant l'installation de l'application (configuration)

## 8 POINTS TECHNIQUES ÉVALUÉS SPÉCIFIQUES AU PROJET

La grille d'évaluation définit les critères généraux selon lesquels le travail du candidat sera évalué (documentation, journal de travail, respect des normes, qualité, ...).

En plus de cela, le travail sera évalué sur les 7 points spécifiques suivants (Point A14 à A20) :

1. Le jeu est fonctionnel, les pièces peuvent pivoter, se déplacer et descendent à un rythme fluide.
2. Lorsqu'une ligne est complétée, une question de programmation est posée à l'utilisateur.  
Une réponse correcte ajoute des points, une réponse incorrecte ajoute un malus sous forme de ligne supplémentaire en bas de l'écran.
3. Le score est enregistré dans la base de données et les 10 meilleurs scores sont visibles dans la page dédiée, par niveau de difficultés
4. Une page d'aide est présente et elle permet à l'utilisateur de comprendre facilement le jeu.
5. Les options sont présentes et fonctionnelles : 2 types de touches pour jouer, niveau de difficulté à choix.
6. Le code est exempt de sections copiées/modifiées (principe DRY : Don't Repeat Yourself) et respectant le style de programmation des langages utilisés.
7. La qualité du repository Git : messages de commits explicites et lisibles, permettant de retracer l'évolution du code (plusieurs commits par jour, création de branches de fonctionnalités), fichier README.md présentant le projet et son déploiement

### Remarque :

Le recours à des outils en ligne d'intelligence artificielle (ex. : Chat GPT) doit être mentionné et ne peut servir que d'inspiration à la réalisation. En cas d'abus, l'évaluation du TPI en tiendra compte.

## 9 VALIDATION

	Lu et approuvé le :	Signature :
Candidat :		
Expert n°1 :		
Expert n° 2 :		
Chef de projet :		