



DIRECTIVES ET INFORMATIONS RELATIVES AU TRAVAIL

- Ce travail compte pour 15% de la note finale.
- Ce travail se fait en équipe.
- La remise du travail pratique complet (format zip) se fait sur Léa avant la date butoir spécifiée sur Léa.
- Il est interdit de partager du code avec d'autres équipes. Attention au plagiat.

LE JEU QUESTIONNAIRE

Monsieur Tartempion est une application interactive toute simple où l'utilisateur doit répondre à des questions tirées aléatoirement d'une banque de questions. Il y a toutefois une tournure particulière à ce jeu questionnaire : il faut répondre de la mauvaise façon! En effet, Monsieur Tartempion a toujours tort.



Figure 1 – Exemple d'une question

Le joueur a 60 secondes pour réussir toutes les questions en autant de tentatives qu'il le souhaite. Si le délai de 60 secondes expire, un écran d'échec lui est affiché pendant trois (3) secondes, puis l'application revient dans un état de lancement d'une nouvelle partie. Cet écran est accompagné d'un effet sonore de déception.



Figure 2 - Image d'échec

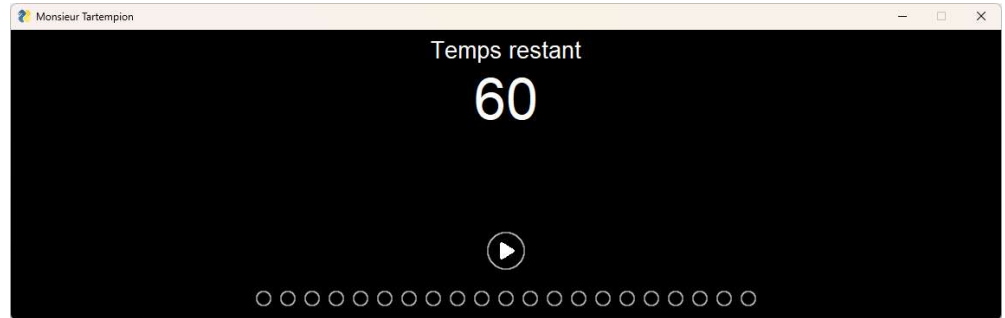


Figure 3 – Lancement d'une partie

Si le joueur manque une question en cours de partie, l'application émet un son d'erreur puis pause momentanément la partie. Le joueur peut effectuer une nouvelle tentative en appuyant sur la flèche.



Figure 4 – Partie momentanément sur pause.

Le joueur doit alors reprendre toutes les questions depuis le début et tenter de réussir la partie dans le temps restant.

En cours de tentative, de petits cercles indiquent l'état des questions :

- Cercle vide (○) Une question qui n'a pas été posée dans cette partie.
- Cercle vert (●) Une question réussie lors de la tentative en cours.
- Cercle jaune (●) Une question réussie lors d'une tentative précédente dans cette partie.
- Cercle rouge (●) La question échouée lors de la meilleure tentative dans cette partie.

L'indicateur rouge indique la question échouée lors de la meilleure performance (la meilleure tentative) dans cette partie.



Figure 5 – Indicateurs en cours de tentative

La figure 5 illustre une tentative en cours où le joueur répond à la quatrième question. Il a réussi les trois premières questions de la tentative, mais a échoué à la sixième question lors d'une tentative précédente. Il lui reste 31 secondes pour réussir toutes les questions.

Si l'utilisateur réussit toutes les questions, il remporte la partie. Un écran de réussite lui est affiché pendant trois (3) secondes, puis l'application revient dans un état de lancement d'une nouvelle partie. Cet écran est accompagné d'un effet sonore d'accomplissement personnel des plus gratifiants (mais pas d'un chèque, malheureusement).



Figure 6 - Image de victoire

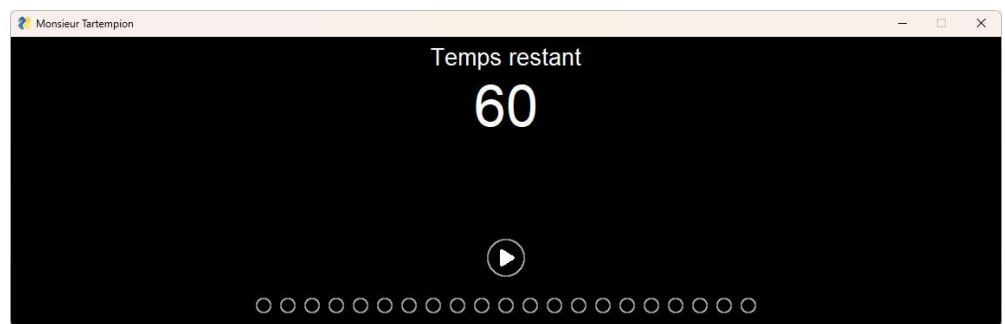


Figure 7 – Lancement d'une partie

Chaque nouvelle partie comporte une nouvelle sélection aléatoire de 21 questions à partir d'une banque de plus de 250 questions.

Pour arrêter le jeu questionnaire, l'utilisateur doit fermer la fenêtre de l'application. À l'heure actuelle et au grand dam des actionnaires ayant investi des sommes astronomiques dans le développement de ce produit, il n'est pas possible d'arrêter une partie en cours sans arrêter l'application.

Lorsqu'on démarre l'application, deux écrans sont affichés avant l'interface de partie. Ces écrans représentent l'image de marque de l'application : le logo des développeurs et l'écran titre de Monsieur Tartempion. Le logo des développeurs s'affiche pendant 1.5 seconde et l'écran titre s'affiche pendant 2 secondes.

MANDAT ET LIVRABLES

Une équipe de codeurs : vous. Une application, du code atroce, plein de bogues. Vos claviers et votre courage en main, saurez-vous décortiquer ce code, le corriger et l'améliorer?

Étape 1	Tests boîte noire	
<p>L'équipe élabore une liste de 5 tests manuels qui permettront de valider le bon fonctionnement du jeu. Chaque test doit avoir un titre, une description et une condition de succès. L'équipe exécute les tests boîte noire et consigne les résultats. <i>Exigence : au moins deux tests doivent échouer (il faut trouver au moins deux bogues en exécutant ces tests boîte noire).</i></p>		
<p>Livable : plan de test et résultats des tests <i>2 pts par test (pénalité de 3 pts par échec manquant)</i></p>		10 pts
Étape 2	Revue de code	
<p>Chaque membre de l'équipe effectue sa préparation et produit une liste d'observations. <i>Exigence : un minimum de 10 observations par préparation individuelle</i> L'équipe tient une revue de code et note les observations dans une fiche de revue de code. <i>Exigence : un minimum de 20 observations notées à la fiche de revue de code</i></p>		
<p>Livable : toutes les préparations individuelles <i>Pénalité de 2 pts par préparation individuelle manquante ou ne rencontrant pas le minimum de 10 observations valides</i></p>		15 pts
<p>Livable : fiche de revue de code dûment complétée <i>Pénalité de 5 pts si la fiche compte moins de 20 observations valides</i></p>		
Étape 3a	Correction et réusinage	
<p>L'équipe procède à la correction (erreurs et défauts trouvés lors des tests et de la revue de code) et au réusinage du code (structure, découpage, algorithmes, validations, gestion des exceptions, etc.). <i>Exigence : respect des bonnes pratiques et du guide de style PEP 8</i> L'équipe réfléchit aux tests unitaires (voir plus bas) et les développe en parallèle.</p>		
<p>Livable : une copie du code corrigé et réusiné</p>		40 pts
Étape 3b	Tests unitaires	
<p>L'équipe conçoit et réalise des tests unitaires avec le module <code>unittest</code> pour vérifier le code réusiné. <i>Exigence : minimum 10 tests, grande couverture de code et vérification des cas d'exception</i> <i>Exigence : tous les tests doivent passer</i></p>		
<p>Livable : une copie de la classe de test (fichier complet) <i>Pénalité de 2 pts par test manquant</i></p>		15 pts
Étape 4	Ajout de fonctionnalités et modifications	
<p>Remplacer le logo d'équipe par le vôtre (base64 exigée). Déplacer les images vers un fichier cornichon (<i>pickle</i>). Vérifier l'intégrité de la base de données et du fichier cornichon avant de les charger. Maintenir le même ordre pour les réponses d'une tentative à l'autre. Proposer votre propre fonctionnalité (l'enseignant doit accepter) et développez-la.</p>		
<p>Livable : une copie du code amélioré <i>Chaque membre de l'équipe doit implémenter au moins une fonctionnalité</i> <i>4 pts par amélioration implémentée (2 pts pour le fonctionnement et si fonctionnement confirmé, 2 pts pour la qualité du code)</i></p>		20 pts