

# TRAVAIL PRATIQUE #2

Cours 420-3D4-MO – Interface utilisateur

## CONSIGNES

- Date de remise du travail : **Vendredi 13 décembre 2024 avant 23h00**
- Remettre le travail sur Moodle dans un fichier compressé (ZIP) nommé **TP2votreNom.ZIP** qui comprend tout le dossier du projet et les prototypes d'écran (voir grille d'auto-évaluation).

## NIVEAU

Niveau	Événements/ Validations dans l'interface	Prototype d'interface
Niveau 3	À déterminer	À déterminer

## UTILISATION DE GITHUB CLASSROOM:

Connectez-vous dans votre compte GitHub. Ensuite, acceptez l'assignation en cliquant sur le lien suivant:

Assignation	Description	Lien
Travailpratique2	Dépôt pour le travail pratique 2	<a href="https://classroom.github.com/a/CJR3isaE">https://classroom.github.com/a/CJR3isaE</a>

## COMMITTS DANS GITHUB CLASSROOM:

Vous devez faire des commits pour les étapes suivantes de votre projet. Vous pouvez faire plus d'un commit pour chacune des étapes:

Étapes	Description de votre commit	
Étape 1	Création du projet	
Étape 2	Création des prototypes d'écrans	
Étape [3, 4, 5]	Création de l'interface <nom de l'écran>	Faire pour chaque écran (3 au total)
Étape [noÉcran-1,...noÉcran-n-1]	Interaction <contrôle> <nom de l'interaction> pour l'écran <nom de l'écran>	On se base sur les événements de chaque écran.
Étape n	Fin du projet	

## COMMITTS OBLIGATOIRES en laboratoire:

<b>Semaine 10</b>	GR 01 : Lundi 4 novembre GR 02 : Lundi 4 novembre GR 03 : jeudi 31 octobre et mercredi 6 novembre	<b>Prototype et interfaces d'écrans :</b> <ul style="list-style-type: none"><li>• Fenêtre 1 : accueil et démarrage du jeu</li><li>• Fenêtre 2 : propriétés des joueurs</li><li>• Fenêtre 3 : fenêtre du jeu</li></ul>
<b>Semaine 11</b>	GR 01 : Mercredi 11 novembre GR 02 : Mercredi 11 novembre GR 03 : Jeudi 7 novembre	<b>Prototype et interfaces d'écrans :</b> <ul style="list-style-type: none"><li>• Fenêtre 1 : accueil et démarrage du jeu</li><li>• Fenêtre 2 : propriétés des joueurs</li><li>• Fenêtre 3 : fenêtre du jeu</li></ul>

## VALIDATIONS POUR LE PROFESSEUR (OBLIGATOIRE):

Vous aurez plusieurs validations pour votre travail pratique. Ces validations sont obligatoires pour la continuation du travail. **Les validations peuvent se faire par Teams, en classe ou sur rendez-vous.**

### PROTOTYPE D'ÉCRANS

Trois écrans sont demandés (Écran d'accueil, Écran propriétés des joueurs et écran de jeu). Vous devez faire un prototype d'écran, pour chaque fenêtre, **AVANT de programmer** votre application. Vous devez aussi respecter les spécifications demandées. **Les trois prototypes d'écrans (format image ou autre) devront être validés par le professeur par Teams ou en classe.**

### FONCTIONNALITÉ

Vous devez me présenter **une fonctionnalité de votre travail pratique pour validation**. La validation peut se faire par Teams (sur rendez-vous), en classe ou par une capsule vidéo avec narration ou avec des explications en fichier Word.

<b>Fonctionnalité à valider:</b>	Création d'un joueur pour une partie et sauvegarde des données dans un fichier.
	Choix des joueurs et débiter une partie.

## CALENDRIER MISE EN ŒUVRE

Date / semaine	Calendrier
30 et 31 octobre (semaine 10)	Présentation de l'énoncé
31 octobre et 4 novembre : en classe, sur Teams et sur Git	Validation des prototypes d'écrans (débiter)
Jusqu'au 14 novembre 23h00 (semaine 11) : en classe, sur Teams et sur Git	Validation des prototypes d'écrans Création des interfaces graphiques (doivent être terminés)
Semaine 12	Atelier 2 : ajout d'une fonctionnalité pour l'impression de données. Évaluer à l'extérieur du travail pratique.
En classe : jusqu'au 5 décembre avant 13h00 Par Teams par Rendez-vous : jusqu'au 5 décembre avant 13h00  Par capsule vidéo avec narration ou explications Word : jusqu'au 5 décembre avant 23h00 (Remise du fichier vidéo sur Teams)	Validation d'une fonctionnalité
13 décembre avant 23h00 (semaine 15)	Remise du travail

## MISE EN SITUATION :

### CRÉATION D'UN JEU DE DÉS AVEC LES FILS D'EXÉCUTION

Vous devez développer une application de jeu de dés élaborés à **deux ou plusieurs joueurs**. Celle-ci doit utiliser les fils d'exécution (threads) pour la gestion des éléments du jeu (dés). **Peut-être accepter : un joueur et un joueur ordinateur.**

### APPLICATION DE JEU:

L'application doit comporter les éléments suivants :

- Avoir au minimum deux classes obligatoires
  - Classe **partie** : qui permet de créer un objet partie avec au minimum les éléments suivants :
    - Nombre de joueurs
    - Éléments de votre partie (différents dés)
    - Objets joueurs dans une structure de données (tableau, liste ou autre)
    - Tour en cours
  - Classe **joueur** : qui permet de créer un objet joueur avec au minimum les éléments suivants :
    - Nom du joueur
    - Éléments qui permet de différencier les joueurs (p.ex. couleur différente par joueur).
    - Pointage du joueur pour la partie en cours
    - Total de parties jouées : à incrémenter après chaque partie
    - Pointage des parties jouées : à incrémenter après chaque partie
- Vous devez utiliser une structure de données (tableau, liste, dictionnaire ou autre) pour intégrer vos classes.
- Avoir au **minimum trois fenêtres obligatoire (voir plus bas dans l'énoncé)**
  - Vous devez gérer toutes les validations possibles dans votre application (p.ex. : saisie d'un nom de joueur, gestion des exceptions, etc.)
- Utiliser les fils d'exécution (threads) pour la gestion des éléments du jeu (p.ex. brassage des dés).
  - Pour chaque brassage, assurez-vous de mettre un temps de quelques secondes avant d'afficher.
  - On peut brasser tous les éléments simultanément.
  - Lors du brassage, l'application ne doit pas être « figée » ou « gelée ».
- Utiliser des sons sur certaines actions de votre jeu.
- **Gérer les validations des champs selon leurs propriétés et leurs types possibles.**
- **Utiliser des images libres de droits (common creative). Plusieurs images de dés sont disponibles sur Moodle.**

### FENÊTRE ACCUEIL DU JEU (PREMIÈRE FENÊTRE):

La fenêtre d'accueil permet de choisir un nombre de joueurs et de sélectionner les joueurs. Vous ne pouvez pas commencer une partie sans que tous les joueurs soient sélectionnés. Celle-ci doit contenir les éléments suivants :

- Titre du jeu avec image ou titre de présentation de votre jeu stylisé
- Bouton Créer d'un joueur
  - Ce bouton affiche la fenêtre des propriétés du joueur pour la création.
- Boîte de liste ou liste déroulante pour sélectionner un joueur.
- Bouton pour confirmer la sélection d'un joueur pour une partie.
- Boîte de liste pour déterminer les joueurs sélectionnés pour la partie.
- Bouton Démarrer une partie.
  - Ce bouton affiche la fenêtre de jeu et débiter une partie selon le nombre de joueurs sélectionnés.
- Bouton Quitter l'application.
  - Ce bouton quitte l'application avec un message avant la fermeture

Une barre de menu qui doit comprendre les éléments suivants :

Options du menu	Description	Raccourcis
<b>Créer un joueur...</b>	Option qui permet la création d'un joueur en ouvrant la fenêtre des propriétés des joueurs	Oui
<b>Démarrer une partie...</b>	Option qui permet de démarrer une partie du jeu. On doit s'assurer que les joueurs soient sélectionnés dans la fenêtre.	
<b>Quitter</b>	Option qui permet de quitter l'application	oui

## FENÊTRE DES PROPRIÉTÉS DU JOUEUR (DEUXIÈME FENÊTRE)

La fenêtre des propriétés du joueur permet de créer un objet joueur avec ses informations. Assurez-vous de faire toutes les validations possibles pour les zones de textes. Au début, les zones de textes doivent être inactives. On utilise le bouton nouveau joueur pour activer les zones de textes. Une option va aussi permettre de sauvegarder les informations du joueur dans un fichier texte. L'écran doit comprendre les éléments suivants :

- Zone de texte pour l'identifiant du joueur.
  - Inactif au début
- Choix d'un avatar ou d'une couleur pour votre joueur
  - Inactif au début
- Autres éléments à votre choix selon le jeu choisi (non obligatoire)
  - Inactif au début
- Zone de liste avec les joueurs créés
  - Selon le joueur sélectionné dans la liste, on affiche ses éléments dans les zones de textes.
- Bouton Nouveau joueur
  - Option qui permet d'effacer les données des zones de textes. On rend actives les zones de texte pour créer un nouveau joueur.
- Bouton Sauvegarder
  - Option qui permet de sauvegarder un joueur et de le mettre dans la zone de liste des joueurs créés. Les zones de texte deviennent inactives
- Bouton Modifier
  - Option qui permet de modifier le joueur sélectionné dans la zone de liste. On active les zones de texte pour être capable de les modifier.
- Bouton Supprimer
  - Option qui permet de supprimer le joueur sélectionné dans la zone de liste. On demande un message de confirmation dans une boîte de message avant la suppression.
- Un menu avec les éléments suivants :

Options du menu	Description	Raccourcis
<b>Exporter les données du joueur</b>	Option qui permet d'exporter les propriétés du joueur (identifiant, avatar, nombres de parties jouées, pointages des parties jouées) dans un fichier JSON ( <b>Atelier 2 : énoncé durant la semaine 12</b> ).	
<b>Imprimer les données du joueur</b>	Option qui permet d'imprimer les propriétés du joueur (identifiant, avatar, nombres de parties jouées, pointages des parties jouées) dans un fichier PDF ( <b>Atelier 2 : énoncé durant la semaine 12</b> ).	
<b>Retour à l'accueil</b>	Option qui permet de retourner à la fenêtre d'accueil	oui

## FENÊTRE DU JEU (TROISIÈME FENÊTRE)

La fenêtre de jeu permet de gérer le déroulement de la partie du jeu de dés. Vous devez pouvoir lancer et arrêter le brassage des dés. Une plate-forme de jeu ou une carte de pointage par joueur doivent être affichées à l'écran. À chaque tour du jeu, on doit afficher visuellement quel joueur doit jouer. Après le tour, on doit afficher le pointage du joueur ou des joueurs qui ont joué et l'état du jeu sur la plate-forme si c'est le cas.

Celle-ci doit comprendre les éléments suivants :

- Un menu avec les éléments suivants :

Options du menu	Description	Raccourcis
<b>Nouvelle partie</b>	Option qui permet de recommencer une partie selon le nombre de joueurs sélectionnés auparavant	Oui
<b>Propriétés des joueurs</b>	Option qui permet de modifier les propriétés des joueurs : noms, éléments pour différencier le joueur	
<b>Retour à l'accueil</b>	Option qui permet de retourner à la fenêtre d'accueil	oui

- Les dés à brasser (images).
- Boutons pour lancer les dés et arrêter le brassage.
- Une plate-forme de jeu (selon votre jeu) ou une carte de pointage (selon votre jeu) par joueur. Ça peut être les deux aussi.
  - La plate-forme de jeu, c'est ce que le joueur va sélectionner (dés ou autre) pour son tour de jeu.
  - La carte de pointage, c'est où l'on inscrit le nom du joueur et son pointage dans le jeu pour les éléments du jeu.
- Affichage du tour du joueur visuellement.
- Affichage de son pointage (carte de pointage) à chaque tour OU affichage de l'état du jeu sur la plate-forme après chaque tour.

## TESTS UNITAIRES

Vous devez créer 2 tests unitaires de votre choix pour votre projet.

## PROTOTYPE DES FENÊTRES

Vous devez créer votre propre interface de jeu (fenêtre d'accueil, fenêtre de propriétés des joueurs, fenêtre du jeu). Assurez-vous de respecter les contraintes de l'énoncé (composants demandés et propriétés des composants).

## CONSTANTES OBLIGATOIRES

Vous devez avoir des variables constantes dans votre programme. Vous pouvez faire une classe « constante » ou les déclarer au début de votre programme et bien identifiées.

## CLASSES OBLIGATOIRES ET STRUCTURE DE DONNÉES

Vous devez avoir deux classes obligatoires (joueur et partie) dans votre programme avec les attributs énumérés plus haut dans l'énoncé. À noter que vous pouvez ajouter d'autres attributs dans vos classes. Assurez-vous d'utiliser une structure de données pour intégrer vos objets.

## GRILLE D'ÉVALUATION - TRAVAIL PRATIQUE #2

Écran d'accueil	Note
(4) Interface avec les contrôles demandés (4) Programmation des événements et validations des contrôles à l'écran (4) Exécution de la fenêtre	<b>12</b>
Écran de propriétés des joueurs	
(4) Interface avec les contrôles demandés (4) Programmation des événements et validations des contrôles à l'écran (4) Exécution de la fenêtre	<b>12</b>
Écran de jeu	
(8) Interface avec les contrôles demandés (8) Programmation des événements et validations des contrôles à l'écran (4) Affichages des données à l'écran (4) Exécution de la fenêtre	<b>24</b>
Validations auprès du professeur	
(4) Prototype de l'écran d'accueil (4) Prototype de l'écran de propriétés des joueurs (4) Prototype de l'écran de jeu (4) Exécution de la fonctionnalité demandée	<b>16</b>
Git et structure du projet	
(4) Commits obligatoires : semaine 10 et 12 (4) Étapes Git (4) Tests unitaires de votre choix (4) Nomenclature, constantes, classes et structure de données	<b>16</b>
Note finale	<b>80</b>

## GRILLE D'AUTO-ÉVALUATION

Questions à se poser avant la remise	
Est-ce que j'ai accepté l'assignation dans GitHub Classroom?	
Est-ce que j'ai fait le commit pour la création du projet vide?	
Est-ce que j'ai créé mon prototype pour mes deux fenêtres ?	
Est-ce que j'ai fait valider mes prototypes (images) auprès du professeur ?	
Est-ce que j'ai créé mes interfaces avec les contrôles selon mes prototypes?	
Est-ce que j'ai fait le commit pour la création des interfaces?	
Est-ce que j'ai programmé tous les événements possibles selon les interfaces ?	
Est-ce que j'ai ajouté mes validations/exceptions possibles selon les interactions?	
Est-ce que j'ai programmé toutes les validations sur les contrôles?	
Est-ce que j'ai fait les étapes Git : commits selon les interactions?	
Est-ce que j'ai fait les 2 tests unitaires dans le programme?	
Est-ce que j'ai éliminé les bogues d'exécution dans le programme?	
Est-ce que j'ai testé les interactions programmées conformes aux interfaces créées?	
Est-ce que j'ai fait le commit pour signifier la fin du projet?	
Est-ce que j'ai remis tout le dossier du projet C# dans un fichier compressé?	
Est-ce que j'ai remis les prototypes d'écran dans mon dossier de projet ou par un lien partagé?	
Est-ce que j'ai fait valider une fonctionnalité auprès du professeur ?	
Est-ce que j'ai remis le fichier ZIP sur Moodle?	