

LOG 725 A2025 - Travail Pratique 2

Dans le TP2, vous allez procéder à la refactorisation d'un jeu de votre choix pour ajouter une implémentation du patron ECS (Entity-Component System).

De plus, vous devrez implémenter d'autres patrons que nous avons abordés pendant le cours, tels que Command, Observer, State, etc.

Vous êtes libre de choisir le jeu et la manière d'implémenter les patrons, la seule exigence étant d'implémenter un ECS minimal, avec des entités, des composants et des systèmes bien définis et documentés.

Étape 1 - Choisir un jeu

Notre suggestion : Vous pouvez modifier un des exemples disponibles dans le package [Unity Playground](#) téléchargé lors de la première séance de laboratoire, utiliser les jeux Pygame du livre [Code the Classics](#), ou vous pouvez utiliser n'importe quel autre jeu Open Source, peu importe le langage ou l'engin (assurez-vous de contrôler la complexité du jeu choisi). Utilisez le système de scripting du game engine en question, si tel est le cas.

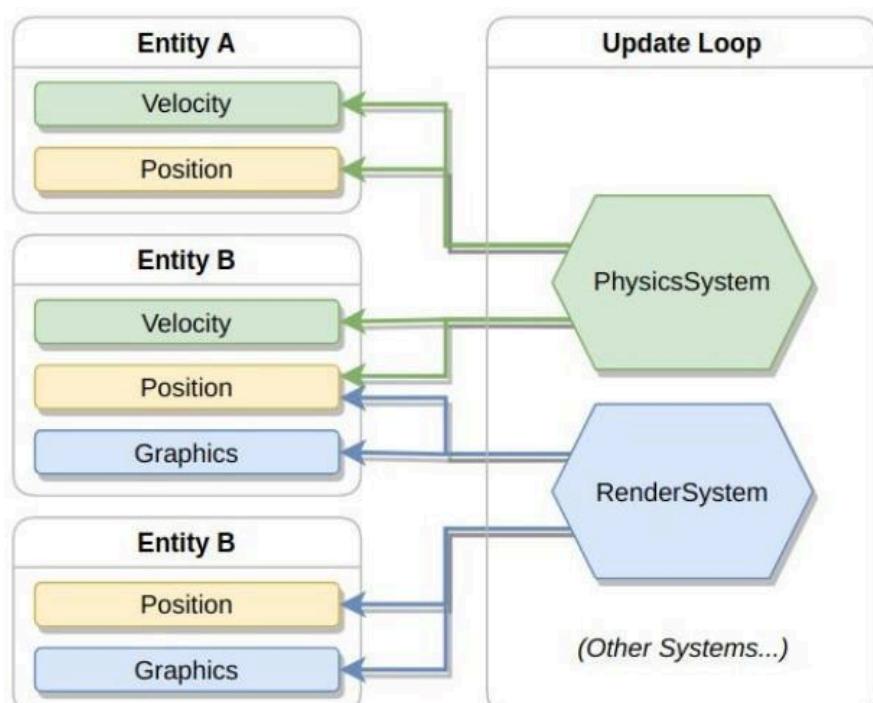
N.B : Il est préférable de choisir un jeu simple, dont l'implémentation peut facilement être modifiée et améliorée.

Étape 2 - Intégrer un ECS et des patrons de conception

Après avoir choisi un jeu, vous devez le refactorer pour y ajouter les éléments suivant :

- Au moins un système géré selon le patron d'architecture *Entity Component System*
- Au moins 2 patrons de conception (Commande, Observateur, Stratégie, etc.)

Vous pouvez utiliser les diagrammes et les documents que nous avons vus pendant les cours et le laboratoire comme référence pour créer les classes, méthodes et structure de projet.



Notes importantes

Pour ce qui est de l'implémentation du patron ECS, vous aurez à faire une implémentation simplifiée et théorique du patron. C'est-à-dire que votre implémentation doit représenter la philosophie et l'architecture ECS, mais celui-ci ne bénéficiera peut être pas du gain de performance que peut amener une implémentation complète du patron. Veuillez prendre en note les points suivants :

- **L'entité** doit être un identifiant ou un conteneur permettant d'y attacher des composants.
- Les **composants** doivent contenir les données nécessaires pour caractériser une entité dans le contexte d'un système. Aucune logique ne doit être contenue dans le composant ; seulement des données.
- Le **système** doit utiliser les données stockées dans les composants des différentes entités pour gérer l'état. C'est ici que toute la logique se trouvera.

Après votre implémentation, le jeu doit fonctionner de la même manière qu'il fonctionnait sans modifications. Votre code ne doit donc pas interférer avec ce fonctionnement, à moins que ce soit à but amélioratif.

Ce que vous devez soumettre :

- Un rapport expliquant la raison derrière l'implémentation de chaque patron de conception que vous avez utilisé. Le rapport doit contenir les sections suivantes :
 - Page ou en-tête de présentation contenant les informations [requises pour les projets de l'ETS](#)
 - **Diagrammes** représentant le fonctionnement du jeu **avant** l'implémentation des patrons **vs. après** l'implémentation des patrons
 - Vous pouvez inclure le type de diagramme que vous voulez, mais je recommande des diagrammes de classe ou de séquence.
 - Mettez bien l'emphase sur les classes modifiées pendant la refactorisation du code.
 - Explication des **relations des patrons** avec le **contexte du jeu** (quelles mécaniques sont touchées par la refactorisation)
 - Explication des **avantages** et des **inconvénients** de l'implémentation des patrons
 - Référence vers le code original du jeu, ainsi que les librairies utilisées, le cas échéant.
- Votre **implémentation** complète de la re-factorisation ECS et des autres patrons choisis.
 - Suivez les instructions de remise sur Moodle.

Grille d'évaluation

Critère	Pondération
Code : Implémentation du patron ECS	/4
Code : Implémentation des patrons de conception	/4
Code : Qualité	/3
Rapport : Contenu	/3
Rapport : Format	/3
Rapport : Diagrammes	/3
Total	/20