

# JIN Interactions Humains-Systèmes

## Projet : Contrôleur d'interactions d'un jeu

v3.0

### Objectifs des compétences

- Analyser une application interactive existante
- Synthétiser une recherche d'informations
- Concevoir et programmer des interactions paramétrables pour un prototype de jeu spécifique
- Ajouter des feedbacks et des éléments de « game feel »
- Conduire des tests utilisateurs

**Pré-requis :** Cours IHM, Connaissances Unity

**Conditions :** En binôme, 6 séances de 3h30 encadrées + travail non-encadré

## 1. Projet, méthode et planning

Le but du projet est de programmer le contrôle d'une entité et de la caméra dans un jeu puis de le rendre « intéressant » à contrôler.

### Etapes :

1. Cours d'introduction sur les interfaces et fonctionnalités associées
2. [TP de prise en main](#) d'une manette sous Unity (individuel, noté)
3. [Choix](#) d'un sujet de projet / type de jeu
4. [Analyse](#) de 2 jeux (binôme pour toute la suite, noté)
5. [Recherche](#) d'informations sur le développement des fonctionnalités de ces jeux (commun)
6. [Développement](#) du contrôleur sous Unity
7. [Recherche](#) d'informations et visionnage de vidéos sur le « game feel »
8. Cours UX et évaluations
9. Conception et ajout de feedbacks au contrôleur
10. Tests utilisateurs et finalisation du projet
11. [Rendu projet, rapport et soutenance](#) (noté)

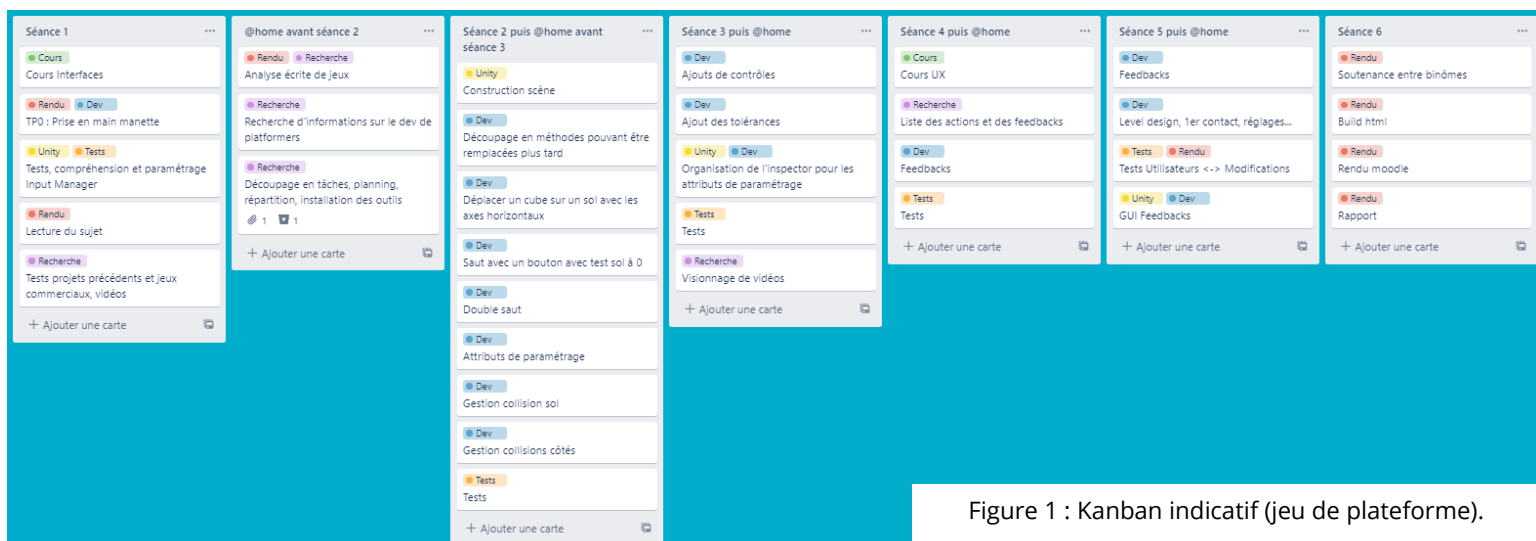
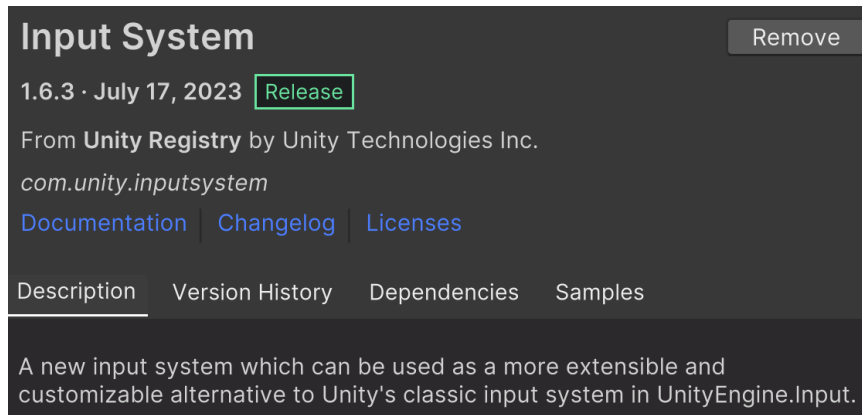


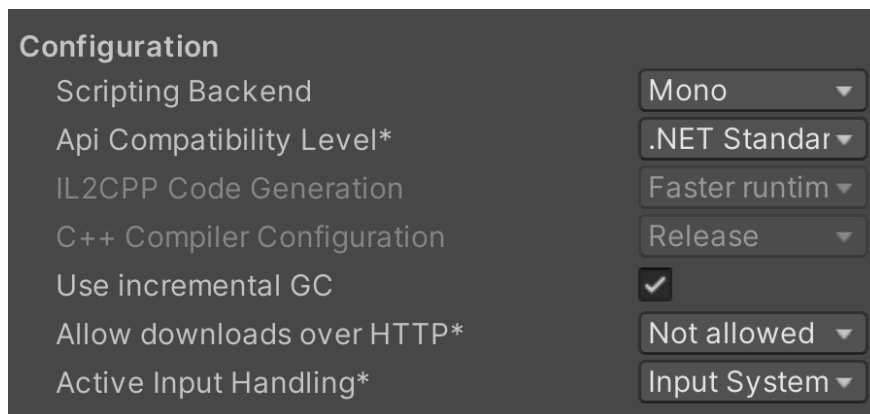
Figure 1 : Kanban indicatif (jeu de plateforme).

## 2. Prise en main manette (noté, individuel)

1. Connecter le gamepad à votre disposition.
2. Installer le package Input System (si besoin) et parcourez sa [documentation](#) ainsi que celle du [gamepad](#)



3. Activer ce nouvel Input System dans les Player Settings du projet à la place de l'ancien Input Manager (si besoin)



4. Créer une scène (2D ou 3D) contenant une sphère blanche au centre de l'écran
  5. Créer un script simple qui :
    - a. Colore la sphère selon l'un des 4 BOUTONS en cours d'appui (blanc quand relâché, ne pas traiter les appuis simultanés, couleurs de la manette Xbox).
    - b. La déplace en absolu dans le plan de l'écran quand on bouge le STICK gauche (elle revient donc au centre quand on lâche le stick). Faire de même avec la CROIX.
    - c. Quitte l'application quand on appuie sur le bouton START pendant une seconde.
- Vous n'aurez pas besoin ici de configurer des Actions, un "[workflow direct](#)" suffit.
6. Quand le script est fonctionnel, générer un BUILD pour WINDOWS puis tester ce build.
  7. Archiver tous les fichiers ainsi générés dans un fichier .zip à votre nom et le déposer sur Moodle (heure de dépôt prise en compte dans la notation)

Vous pourrez ensuite vous servir de cette scène pour tester les effets de différents paramètres d'Input proposés par Unity, ou ajouter des fonctionnalités (vibrations...). Parcourez la doc ou des [tutoriaux](#) pour une configuration plus avancée.

### 3. Choix du sujet

L'objectif principal du projet est de concevoir et développer le contrôleur des actions de l'acteur principal et de la caméra d'un jeu à choisir parmi les propositions de types suivantes :

1. Jeu de plateforme en vue de côté (contrôle des mouvements)
2. Jeu de course en vue de dessus (contrôle des déplacements)
3. Jeu de pêche en vue de dessus ou de côté (contrôle de la mécanique de pêche)

L'interface visée principalement sera la manette mais le projet devra être jouable au clavier.

### 4. Analyse de jeux existants et conception

#### 4.1. Analyse (noté, en binôme)

- Analyser en profondeur les mécaniques principales et les contrôles de 2 jeux existants du type choisi, en les testant si possible, ou en regardant des vidéos.
- Sur chaque jeu, écrire 2-3 pages récapitulatives de tous les contrôles, actions, mouvements caméra, effets esthétiques, feedbacks... que vous avez identifiés (fournir liens vers vidéos d'illustration).
- A déposer sur moodle avant la 2<sup>ème</sup> séance

#### 4.2. Conception

- Faire un document synthétique contenant la liste
  - o des fonctionnalités à développer
    - en terme de mouvements, d'actions et de caméra
    - avec la liste de leurs paramètres possibles
    - et les commandes manette/clavier associées
  - o des autres fonctionnalités minimales de la scène principale du jeu (environnement, simulation...)

#### 4.3. Recherche

- Rechercher des informations sur le développement des fonctionnalités que vous avez identifiées
- Contribuer au document partagé en ligne (lien sur moodle)

### 5. 1<sup>ère</sup> partie du développement

#### 5.1. Objets et Scène

A partir de vos analyses, conceptions, et recherches, l'objectif de cette 1<sup>ère</sup> partie est de programmer le contrôle d'un game object basique dans un environnement représentatif du jeu.

L'environnement visuel sera volontairement simple. L'accent sera mis sur les fonctionnalités de contrôle/mouvement et les réglages disponibles pour un game designer dans l'inspecteur unity.

Vous ne devez pas utiliser les packages unity déjà tout prêts (ex. pas de character controller) mais vous pouvez utiliser les outils génériques existants (ex. physique).

Pour vous donner une idée de cette 1<sup>ère</sup> phase (ou voir les problèmes potentiels) vous pouvez tester les projets des [promos précédentes](#) (jeu de plateforme uniquement).

Les entités principales et la scène sont constituées de solides simples (carré, rectangle, cercle...). La scène peut être 2D ou 3D. Les mouvements sont en 2D.

La caméra simule a priori une vue 2D et peut rester fixe dans un 1<sup>er</sup> temps.

Le level design doit rester simple (ex. figures 3 et 4) et mettre en valeur vos algorithmes et vos réglages de contrôle, pas vos compétences de level designer.

Il n'y aura pas de son ni d'effets visuels dans cette 1<sup>ère</sup> partie (objets rigides, pas d'animation, matériaux simples, pas de lumière particulière...).

Figure 3 : exemple de 1<sup>ère</sup> scène de travail (plateforme)

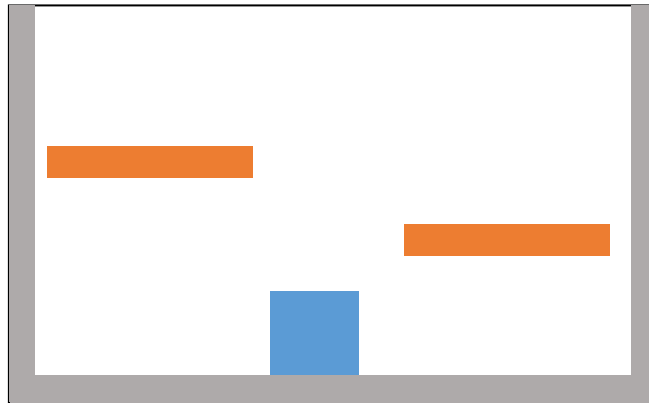


Figure 4 : exemple de 1<sup>ère</sup> scène de travail (course)



## 5.2. Réglages

- Faire apparaître dans l'**inspecteur** les variables nécessaires à un game designer pour paramétrer vos contrôles.
- En bonus, donnez accès à ces réglages via un **GUI** qui sera donc utilisable dans la version html finale.

## 6. 2<sup>ème</sup> partie du développement

### 6.1. Etude préliminaire sur le « Game Feel »

Regarder entre 2 et 4 vidéos parmi les suivantes :

- Best Practices for fast game design in Unity, Renaud Forestié  
<https://youtu.be/NU29QKag8a0?t=1738>
- Jan Willem Nijman - Vlambeer - "The art of screenshake"  
<https://youtu.be/AjdEqssNZ-U>
- Secrets of Game Feel and Juice | Game Maker's Toolkit  
[https://youtu.be/216\\_5nu4aVQ](https://youtu.be/216_5nu4aVQ)
- Juice it or lose it - a talk by Martin Jonasson & Petri Purho  
<https://youtu.be/Fy0aCDmgnxg>

### 6.2. Feedbacks

A partir de ces conférences et de vos analyses de jeux :

- Identifier des feedbacks pour améliorer l'expérience utilisateur (vous pouvez ajouter des actions/événements par rapport à la v1)
- En choisir et en développer quelques-uns (faire une 2<sup>ème</sup> version du jeu dans une scène séparée)
- Ajouter dans un GUI de quoi les activer/désactiver individuellement et tous ensemble

### 6.3. Tests et finalisation

- Améliorez le 1<sup>er</sup> contact avec votre scène, le paramétrage des contrôles, le flow des écrans, le level design et les feedbacks
- En vous inspirant des méthodes vues en cours, organisez des tests avec au moins 2 utilisateurs non-JIN.
  - o Ex. de questions à se poser :
    - Quel profil ont-ils ? Ont-ils l'habitude du type de jeu ?
    - Est-ce qu'ils comprennent rapidement l'environnement, le but (cf. affordances)
    - Est-ce qu'ils interagissent facilement ?
    - Proposent-ils des modifications ?
    - ...
  - o Rédigez un compte-rendu de ces tests
  - o Si besoin, modifiez votre projet en conséquence
- Attention à ne pas faire preuve de trop d'imagination/difficulté dans votre level design et votre gameplay, cela nuira aux tests (et à la correction)

## 7. Notation

### Soutenance (dernière séance)

L'application sera d'abord testée par un autre binôme qui suivra une liste d'éléments à vérifier et donnera son avis, sans que vous puissiez intervenir.

Vous pourrez ensuite répondre aux questions et vous expliquerez ensuite la manière dont vous avez conçu et codé les contrôles et feedbacks, puis évalué votre projet. L'objectif est de montrer les compétences acquises sur les interactions, les problèmes principaux et les solutions apportées.

### 2 rendus à faire **\*avant\*** la dernière séance

Dépôt 1 sur moodle d'un dossier zippé Nom 1 – Nom 2 comportant :

- Un build webgl/html5 fonctionnel (testé avec une manette) qui sera mis directement en ligne sur <https://web4.ensie.fr/~bouyer/IHM/Gamepad2024/> et servira aux soutenances
  - NB : pour fonctionner sur le serveur web de l'école : Player settings -> Publisher -> enlever la compression
  - Si la manette n'est pas reconnue, une solution peut être de mettre un délai dans le code avant de l'initialiser
- Un build windows fonctionnel

Dépôt 2 sur moodle d'un dossier zippé Nom 1 – Nom 2 comportant :

- L'ensemble du projet Unity, nettoyé des ressources et dossiers inutiles (library, obj, temp, fichiers.pdb, assets inutiles...)
- Un document pdf contenant
  - un court récapitulatif des contrôles et fonctionnalités disponibles, des feedbacks choisis et des valeurs exactes de vos réglages (ex : vitesses, gravité, tolérances, inertie...)
  - le compte-rendu d'évaluation d'une version du projet sur au moins 2 personnes (points négatifs et positifs détectés, solutions choisies...)
  - les captures d'écran du niveau principal du jeu et des éventuelles autres scènes (menus, paramètres...)

La **notation** portera sur (voir [grille d'évaluation indicative](#) ci-dessous) :

- les aspects techniques :
  - les contrôles de base
  - les contrôles plus avancés
  - les feedbacks
  - la qualité des collisions
  - les réglages possibles pour un game designer
- les aspects **d'utilisabilité** :
  - le level design jouable
  - le paramétrage possible pour le testeur
  - les affordances
  - la 1<sup>ère</sup> impression

La notation ne portera pas sur l'architecture du code.

## 8. Autres exercices pour s'entraîner sur d'autres problématiques

Voir un ancien [TP manette](#).

## 9. Plus d'informations sur l'UX

Livres de base

- Méthodes de design UX: 30 méthodes fondamentales pour concevoir des expériences optimales, septembre 2018, Carine Lallemand, Guillaume Gronier
- Don't make me think! : Web & Mobile Usability, octobre 2014, Steve Krug
- The Design of Everyday Things, novembre 2013, Don Norman

Sites à suivre

- <https://uxdesign.cc/>
- <https://www.smashingmagazine.com/usability-and-user-experience/>
- <https://www.nngroup.com/articles/>
- <https://uxplanet.org>
- <https://medium.com/topic/ux>
- <http://www.allaboutux.org/>

## 10. Grille d'évaluation indicative

	0	0,25	0,5	0,75	1
<b>Rendus et Evaluations (7)</b>					
analyse					
rapport					
évaluations					
respect des consignes (webgl, exe, clavier...)					
<b>Mouvement et autres actions, events (7)</b>					
Mouvements de base (selon sujet)					
Actions et events supplémentaires (selon sujet)					
raffinements, tolérances...					
<b>Environnement, simulation, physique (5)</b>					
Contenu de base (selon sujet)					
Autres					
<b>Caméra (2)</b>					
Gestion caméra (selon sujet)					
<b>Utilisabilité (5)</b>					
écran d'accueil					
flow, retours, gestion erreurs					
menus					
aide/contrôles en jeu ou dans menu					
ui en dehors des menus					
1er contact : niveau et objectif compréhensible					
affordances en jeu					
level design utile/jouable					
<b>Feedbacks, game feel (5)</b>					
liste des retours suites aux actions					
liste des retours suites aux events					
<i>saut montant, parabole</i>					
<i>descente, collision, frottement</i>					
<i>course, dérapage</i>					
<i>succès</i>					
<i>défaite, mort, dégats...</i>					
<i>autre</i>					
<b>Paramétrage hors code (2)</b>					
GUI feedbacks					
réglages mouvements, actions, et caméra					



## 11. Indication prise en main ancien Input Manager

Lire la doc de l'Input Manager (<http://docs.unity3d.com/Manual/class-InputManager.html>). Grâce au mapping figure 2, y ajouter ou modifier 8 axes pour représenter les 4 boutons A, B, X, Y puis le stick gauche et la croix directionnelle (indice : noms identiques).



Figure 2 :

Mapping Unity-Xbox 360 controller (Archive de <https://wiki.unity3d.com/index.php?title=Xbox360Controller>)

**NB** : Attention au mapping avec le contrôleur Xbox One potentiellement différent selon la version de windows utilisée...