



Factory Frenzy

Projet final

Démarrage à 13h35

Sommaire

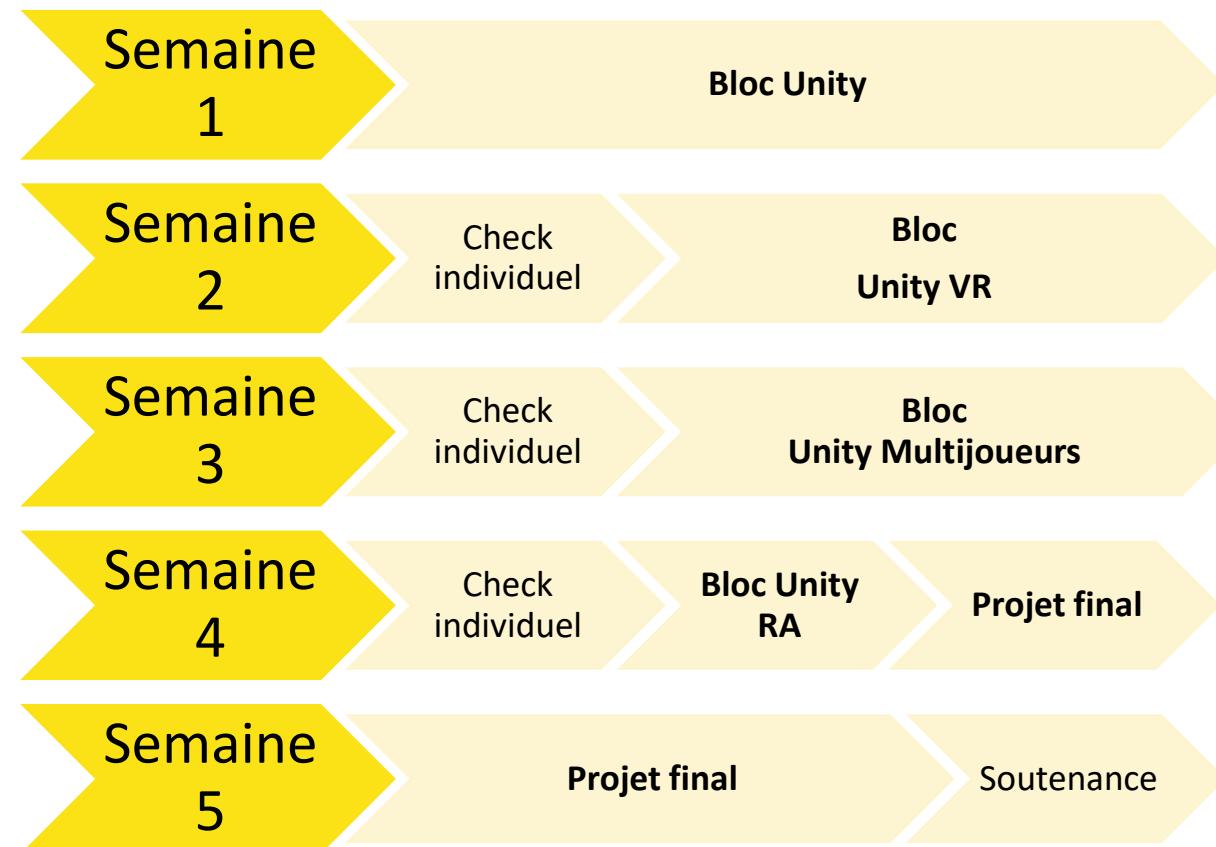
- 1. Rappel du planning**
- 2. Contexte**
- 3. Fonctionnalités**
- 4. Évaluation du projet**



Factory Frenzy

Rappel du planning

Synoptique



Logiciels et le matériel

Logiciels à Installer

- Installation Unity (version 2022.3.1 - 2022.3.3 - 2022.3.8) + Visual Studio Community
A installer grâce au Unity Hub <https://unity3d.com/fr/get-unity/download>

Ressources

Les ressources sont sur le [OneDrive](#)

Les ressources fournies ont un poids élevé. *Merci de les synchroniser la veille au soir pour ne pas perdre de temps et surcharger le réseau pédagogique en journée.*

Matériel

Les matériels VR, Vive ou Quest + PC VR, seront utilisés **uniquement**:

- Dans le bloc **Unity VR** (Semaine 2)
- Dans le **projet** (fin de semaine 4 & semaine 5)

Pour les autres blocs (Unity, Unity Multi-joueurs et Unity AR)

- Rester sur votre propre machine



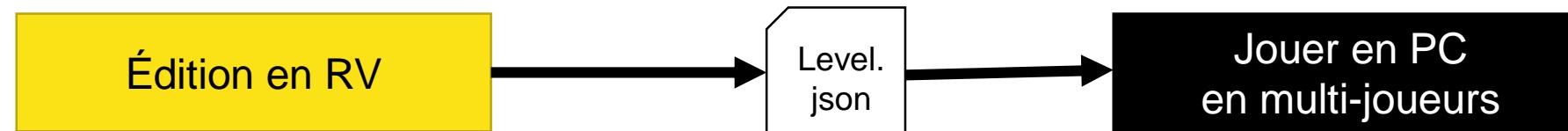


Factory Frenzy Contexte

Factory Frenzy

Contexte

- Vous venez d'être embauché·e dans une nouvelle entreprise du jeu-vidéo.
- En tant que challenger, elle souhaite mettre en place un jeu-vidéo permettant à des personnages d'effectuer une course qui sera composée de différents éléments (plateformes et pièges); l'idée est de faire un jeu similaire à Fall guys. Le jeu se décomposera en deux applications permettant:
 - En réalité virtuelle: d'**éditer** des niveaux de jeux et de les exporter dans un fichier JSON
 - Sur PC classique: de **jouer en multi-joueurs** au niveau (KMS: Keyboard Mouse Screen).
- Le logiciel s'appellera **Factory Frenzy**
- Afin de tester la viabilité de l'approche, un premier POC doit être développé en 10 jours.
- Le document suivant décrit les fonctionnalités souhaitées.



Factory Frenzy

Éléments composants le niveau

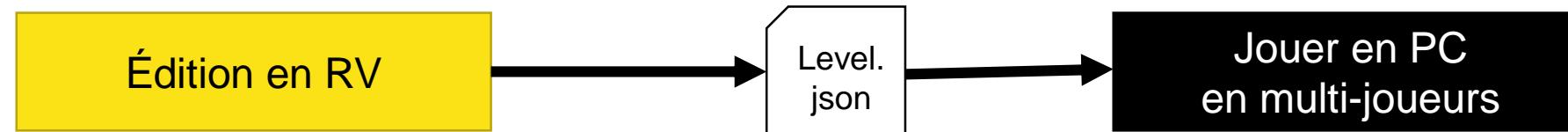
- Plateforme début/fin (verte/rouge) *
- Plateforme classique
- Plateforme mobile
- Bumper
- Lanceur (statemachine recherche de joueurs / tir)
- Trampoline
- Ventilateur
- Checkpoint *

Plus de détails seront fournis dans la partie concernant chacun des éléments.

Factory Frenzy

Projet initial

- Le projet initial contenant les ressources est disponible dans **Factory-Frenzy-2.0.0-Platform-Trap.zip**. (Assets/FactoryFrenzy Prefabs et Scene)
- Il fonctionne sous Unity 2022.3.8.
- Il faudra créer deux projets distincts pour travailler. L'un pour l'édition en réalité virtuelle, l'autre pour le jeu multi-joueurs sur PC. C'est le fichier d'export de niveau qui permettra de faire le lien entre les deux applications.



Liste des fonctionnalités présentées

PC (* fonctionnalités obligatoires)

- * PC-PLA-P1 – Character Controller TPS en multijoueurs
- * PC-PLA-P2 – Plateforme de début de course
- * PC-PLA-P3 – Plateforme de fin de course
- * PC-PLA-P4 – Plateforme classique et gestion de la chute
- PC-PLA-P5 – Plateforme mouvante
- PC-PLA-P6 – Plateforme trampoline
- PC-PLA-P7 – Checkpoint

PC-TRA-T1 – Piège Bumper

* PC-TRA-T2 – Piège Lanceur

PC-TRA-T3 – Piège Ventilateur

* PC-LOB-L1 – Lobby du jeu

PC-LOB-L2 – Import du niveau édité en réalité virtuelle

VR

- VR-FP1-1 – téléportation de l'avatar RV
- VR-FP1-2 – sélectionner et créer des éléments dans une UI
- VR-FP1-3 – attraper et placer des objets
- VR-FP1-4 – supprimer des éléments du niveau
- VR-FP1-5 – export json du niveau
- VR-FP1-6 – son et vibration

VR-FP2-1 – Placement et snap

VR-FP2-2 – Verrouiller un élément

VR-FP2-3 – paramétrage des éléments de la plateforme mobile



Factory Frenzy – jeu PC en multijoueurs Fonctionnalités à développer

Fonctionnalités de priorité 1 – les plateformes (PLA)

PC- présentation générale

Cette partie concerne le jeu de course à développer en multi-joueurs. Les **personnages**, vus à la **troisième personne**, doivent se déplacer pour évoluer sur le niveau de jeu. Le jeu à réaliser est équivalent à un **Fall Guys**.

Le premier arrivé remporte la course

Bien sûr il faudra **intégrer le niveau édité** depuis la réalité virtuelle, mais afin de pouvoir avancer le développement en parallèle, il faudra **initialement placer des éléments directement dans la scène pour tester** le développement PC.



Les fonctionnalités marquées d'une **étoile bleue** doivent être **obligatoirement** développées

★PC-PLA-P1 – Character Controller TPS en multijoueurs

L'objectif est de permettre à plusieurs joueurs connectés dans la même Room de faire la course sur le même niveau de jeu.

Il faut donc:

- intégrer **un avatar** par joueur connecté;
- le caractère représentant le joueur doit être **suivi par une caméra à la troisième personne**;
- Le joueur doit pouvoir sauter de plateforme en plateforme;
- les différents joueurs doivent pouvoir **se gêner (= entrer en collision)** lors des déplacements;

Technique:

- L'application devra utiliser le système Unity [netcode](#) pour la partie multijoueurs
- Pour le caractère à la troisième personne, il faudra utiliser les Starter Assets et le transformer en prefab en multi-joueurs.
<https://assetstore.unity.com/packages/essentials/starter-assets-third-person-character-controller-up-196526>



★PC-PLA-P2 – Plateforme de début de course

Il faut définir la plateforme de **début de course** avec ces fonctionnalités:

- représenter la **ligne de départ en vert**;
- **aligner l'ensemble des joueurs** sur la ligne de départ;
- **gérer le décompte** pour signaler le début de la course;
- tant que le décompte n'atteint pas 0, les joueurs ne peuvent pas se déplacer.

Technique:

Pour gérer le décompte et le moment auquel tous les joueurs pourront déplacer leur personnage il faudra utiliser les Remote Procedure Call (RPC). Il en existe deux types:

- **ClientRpc**
- **ServerRpc**

A vous de choisir la plus fiable.

★PC-PLA-P3 – Plateforme de fin de course

Il faut définir la plateforme de **fin de course** avec ces fonctionnalités:

- représenter **la ligne d'arrivée en bleu**;
- le **premier joueur** qui passe la ligne est classé **1er**;
- Les autres sont classés **2nd**, puis **3^{ème}** etc;
- **10 secondes après que le premier joueur est arrivé**, la course s'achève et tous les joueurs voient une interface fournissant le résultat de la course;
- les joueurs n'ayant pas atteint la ligne d'arrivé sont **affichés sans classement**.

Technique:

De même il faudra utiliser les **RPC** pour signaler les différentes arrivée, et gérer l'affichage de l'interface de fin.

★PC-PLA-P4 – Plateforme classique et gestion de la chute

La **plateforme classique** est un modèle 3D permettant aux personnages de se déplacer dessus.

Lorsque le **joueur tombe**, il faut détecter sa chute et le replacer **au dernier checkpoint**.

Le premier **checkpoint** est représenté par la plateforme de départ.

Technique:

Pour détecter la chute, il faut placer une **Collider** en dessous du niveau et détecter lorsque le joueur rentre dedans avec l'événement **OnTriggerEnter**.

PC-PLA-P5 – Plateforme mouvante

La **plateforme mouvante** est paramétrée dans la partie éditeur avec une **position de début** une **position de fin** et un **temps de déplacement**. Il faut donc programmer la plateforme pour qu'elle se déplace entre les deux points

Technique:

Il existe plusieurs fonctions pour réaliser une interpolation entre deux points **Vector3.Lerp()** ou une interpolation aller-retour entre deux valeurs **Mathf.PingPong()**

PC-PLA-P6 – Plateforme trampoline

La plateforme **trampoline** permet au joueur de rebondir fortement vers le haut et d'atteindre les plateformes situées plus haut dans un parcours.

Technique:

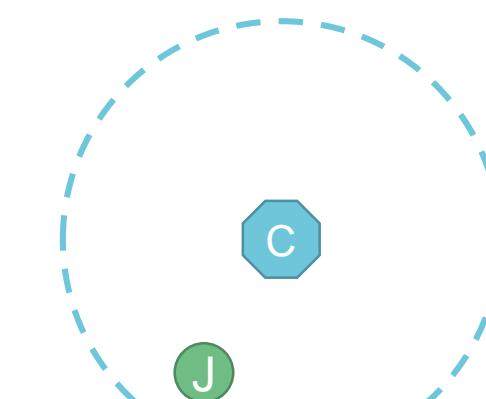
Il faut utiliser le **AddForce** du **Rigidbody** afin de projeter le joueur vers le haut

PC-PLA-P7 – Checkpoint

Le **checkpoint** permet au joueur de repartir de ce point lorsqu'il tombe pendant le parcours, au lieu de repartir du début du niveau.

Lorsque le joueur passe dans la zone de détection du checkpoint:

- le **checkpoint s'active**,
- un **retour visuel** est affiché pour le joueur
- et le joueur **repart de ce checkpoint** lorsqu'il **tombe** dans le vide.





Factory Frenzy – jeu PC en multijoueurs Fonctionnalités à développer

Fonctionnalités de priorité 1 – les pièges (trap TRA)

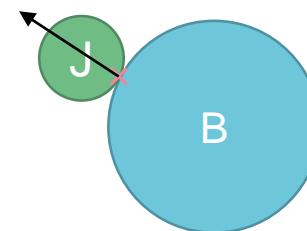
PC-PIE-T1 – Piège Bumper

Le **bumper** est un piège qui va créer une **force** envoyant le joueur **dans la direction opposée** à celle du point de contact entre le joueur et le bumper par rapport au point de contact (voir figure à droite).

Technique:

Vue de dessus du Bumper (B en bleu), du Joueur (J) en vert et du point de contact en rouge.

La flèche noire représente la force à appliquer au Joueur



★PC-PIE-T2 – Piège Lanceur

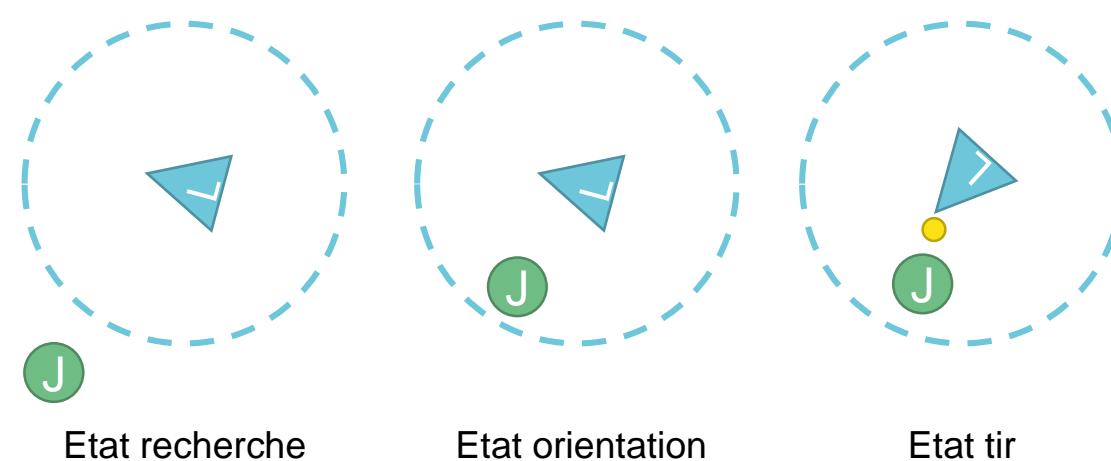
Le **Lanceur** (voir triangle bleu) est un élément statique qui cherche les joueurs et tir dessus. Le lanceur est composé de trois états:

- **Etat recherche** - quand aucun joueur n'est dans sa zone de détection en pointillé bleu. Le lanceur tourne entre deux angles définis.
- **Etat orientation** – quand un joueur est dans la zone de détection du lanceur et que le joueur est à plus de 30° de l'axe de tir du lanceur. Le lanceur s'oriente vers le joueur sur lequel tirer. S'il y a plusieurs joueurs, il s'oriente vers le premier rentré dans la zone.
- **Etat tir** – quand un (ou des) joueur(s) est (sont) dans la zone de détection du lanceur et que le(s) joueur(s) est à moins de 30° de l'axe de tir du lanceur. Le lanceur tire sur le(s) joueur(s) dans son axe de tir.

Quand le tir touche le joueur, le joueur est projeté fortement hors du parcours

Technique:

Pour développer le lanceur il faut utiliser le principe de **Finite State Machine**

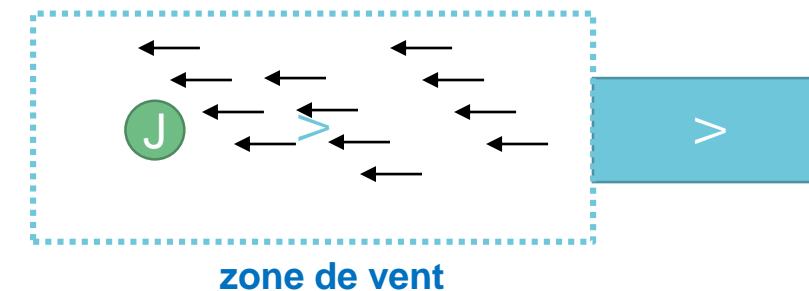


PC-PIE-T3 – Piège Ventilateur

Le **Ventilateur** (rectangle bleu) est un élément statique qui envoie une **force régulière** qui fait dévier le joueur lorsqu'il se trouve dans son champ d'action

La **zone de vent** est définie dans l'axe du Ventilateur

Pour **représenter physiquement le vent**, il faut utiliser les effets de particules de Unity





Factory Frenzy – jeu PC en multijoueurs Fonctionnalités à développer

Fonctionnalités de priorité 1 – le lobby (LOB)

★PC-LOB-L1 – Lobby du jeu

Puisque le jeu est **multi-joueurs**, il est nécessaire de créer un Lobby afin de pouvoir connecter les utilisateurs.

Le **lobby** doit permettre de connecter jusque **maximum 10 joueurs** dans la même session de jeu.

S'il n'y a **pas de session en cours**, une **nouvelle session** est créée.

Technique

Le Lobby sera mis en place lors du bloc Multijoueurs de l'option et pourra être réutilisé dans le projet final.

PC-LOB-L2 – Import du niveau édité en réalité virtuelle

Le **lobby** doit permettre au **Master** de **sélectionner le niveau du jeu à charger** parmi les différents fichiers json édités en réalité virtuelle.

Lorsque **le fichier est sélectionné** et que la session est pleine, alors **le niveau est chargé** pour l'ensemble des joueurs, puis la partie peut commencer.

Technique

Il faut évidemment que le groupe **définisse précisément le format de fichier** et les classes à utiliser pour sérialiser et déserialiser le niveau édité.



Factory Frenzy – édition en réalité virtuelle Fonctionnalités à développer

Fonctionnalités de priorité 1

VR-FP1-1 – téléportation de l'avatar RV

L'utilisateur en RV doit pouvoir se déplacer par téléportation

- La vidéo montre un exemple d'un projet précédent.



VR-FP1-2 – sélectionner et créer des éléments dans une UI

Chaque niveau est composé de différents éléments (plateformes + pièges):

- Plateformes:
 - **Plateforme début**
 - **Plateforme fin de course**
 - **Plateforme classique**
 - **Plateforme mobile (distance parcourue et durée de déplacement fixes)**
- Pièges:
 - **Bumper**
 - **Lanceur (statemachine recherche de joueurs / tir)**
 - **Trampoline**
 - **Ventilateur**

Cette fonctionnalité doit permettre de proposer **une interface Humain-Machine** pour permettre de **sélectionner** et **créer** des éléments

L'objectif de cette UI est de proposer une interface permettant d'accéder le plus rapidement et efficacement à l'élément à créer dans la scène.

- L'évaluation se fera sur **l'accessibilité et l'utilisabilité** de l'interface
- Et plus précisément sur le **temps moyen** pour accéder à un objet et le placer dans le niveau ensuite



Exemple de l'interface VREX

VR-FP1-3 –attraper et placer des objets

Chaque niveau est composé de différents éléments (plateformes + pièges)

L'utilisateur en RV doit pouvoir:

- **attraper les objets** permettant d'éditer le niveau
- **placer les objets** dans la scène
- **modifier la pose** d'un objet déjà placé

La vidéo montre un exemple d'un projet précédent montrant le placement d'objets.

Puisque le jeu permet d'éditer des niveaux, il ne faut bien sûr pas avoir de gravité sur les éléments composants le niveau.



VR-FP1-4 – supprimer des éléments du niveau

Lorsque le niveau est en cours d'édition il faut pouvoir **supprimer un élément placé** dans le niveau.

De la même manière, il faut proposer un système permettant de supprimer le plus rapidement et efficacement possible un élément du niveau.

Optionnellement et si vous avez terminé, proposez un système qui permet de supprimer un ensemble d'éléments sélectionnés



VR-FP1-5 – export json du niveau

Proposez un **modèle de donnée** permettant d'exporter le niveau dans un fichier json afin de pouvoir le transférer de l'éditeur en RV vers le jeu sur PC

Créer une interface en RV permettant de **définir le nom du fichier et d'exporter en json** avec ce nom.

Pensez à **signaler l'écrasement du fichier** s'il existe déjà.

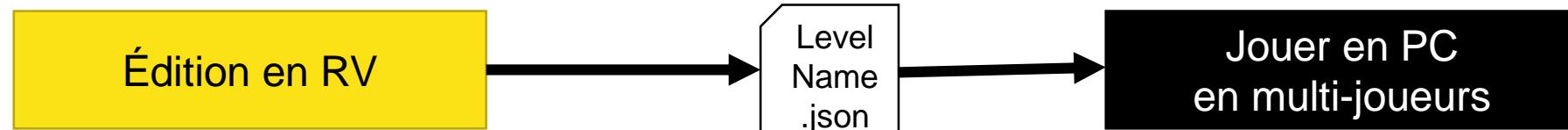
Technique:

Pour pouvoir exporter en JSON, il faut définir une classe **[Serializable]** et utiliser le **JsonUtility** de Unity.

```
[Serializable]
public class PlayerScore
{
    public int level;
    public float timeElapsed;
    public string playerName;
}
```

Un exemple est fourni sur la documentation Unity pour cela

<https://docs.unity3d.com/2022.3/Documentation/ScriptReference/JsonUtility.html>



VR-FP1-6 – son et vibration

L'objectif de cette fonctionnalité est d'ajouter des **sons** et/ou des vibrations sur les différentes fonctionnalités

- **Sélection**
- **Placement**
- **Suppression**
- **Sauvegarde**
- ...

A l'usage, il se peut que trop de retours sonores ou haptiques soient gênants, vous justifierez donc vos choix.





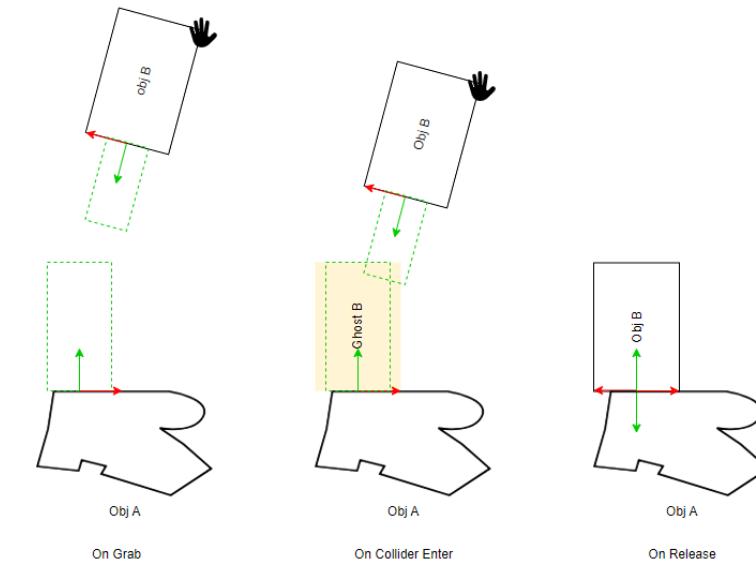
Factory Frenzy – édition en réalité virtuelle Fonctionnalités à développer

Fonctionnalités de priorité 2

VR-FP2-1 – Placement et snap

L'objectif de cette fonctionnalité est de développer un **placement intelligent** afin de **faciliter le placement des objets** avec des techniques d'interactions plus avancées:

- La première consiste à permettre de **snaper** un objet sur un autre (voir sur le schéma et l'image présentée).
- La seconde, **en contrignant les angles** de rotation par des pas de 10° par exemple. La valeur du pas doit être paramétrable. Cela permet de ne faire que des orientations communes, évitant ainsi les placements fastidieux.
- Vous pourrez activer ce mode de **placement intelligent** par l'appui sur une des touches du contrôleur, passant ainsi de placement libre à placement intelligent



Principe d'un snap

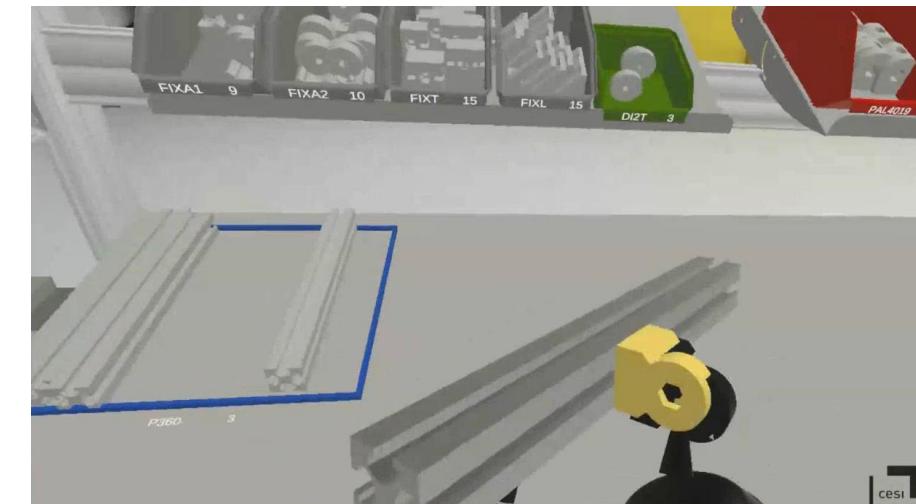


Illustration d'un snap

VR-FP2-2 – Verrouiller un élément

Afin d'éviter d'attraper et déplacer un des éléments par erreur, il faut proposer une interaction permettant de **verrouiller/déverrouiller** un élément du niveau en cours d'édition et afficher l'information de verrouillage.



VR-FP2-3 – paramétrage des éléments de la plateforme mobile

Un des éléments qui peut être placé est la **plateforme mobile**. Il faut pouvoir proposer un ensemble d'interactions pour pouvoir paramétriser cette plateforme mobile:

- **Position de départ** de la plateforme mobile et **position d'arrivée**
- **Temps de déplacement**





Factory Frenzy évaluation du projet

Synthèse des fonctionnalités

PC (* fonctionnalités obligatoires)

- * PC-PLA-P1 – Character Controller TPS en multijoueurs
- * PC-PLA-P2 – Plateforme de début de course
- * PC-PLA-P3 – Plateforme de fin de course
- * PC-PLA-P4 – Plateforme classique et gestion de la chute
- PC-PLA-P5 – Plateforme mouvante
- PC-PLA-P6 – Plateforme trampoline
- PC-PLA-P7 – Checkpoint

PC-TRA-T1 – Piège Bumper

* PC-TRA-T2 – Piège Lanceur

PC-TRA-T3 – Piège Ventilateur

* PC-LOB-L1 – Lobby du jeu

PC-LOB-L2 – Import du niveau édité en réalité virtuelle

VR

- VR-FP1-1 – téléportation de l'avatar RV
- VR-FP1-2 – sélectionner et créer des éléments dans une UI
- VR-FP1-3 – attraper et placer des objets
- VR-FP1-4 – supprimer des éléments du niveau
- VR-FP1-5 – export json du niveau
- VR-FP1-6 – son et vibration

VR-FP2-1 – Placement et snap

VR-FP2-2 – Verrouiller un élément

VR-FP2-3 – paramétrage des éléments de la plateforme mobile

Évaluation du projet



Fonctionnalités PC – coefficient 3

A

- 6/6 des fonctionnalités imposées ET au moins 3 des autres fonctionnalités

B

- 6/6 des fonctionnalités imposées ET au moins 2 des autres fonctionnalités

OU

- 5/6 des fonctionnalités imposées ET au moins 2 des autres fonctionnalités

C

- 4/6 des fonctionnalités imposées ET au moins 1 des autres fonctionnalités

D sinon

Fonctionnalités VR – coefficient 3

A

- 6/6 des fonctionnalités P1 ET au moins 2 P2

B

- 6/6 des fonctionnalités P1

OU

- 5/6 des fonctionnalités P1 ET au moins 1 P2

C

- 4/6 des fonctionnalités P1 ET au moins 1 P2

D sinon

Évaluation du projet



Conception du projet – coefficient 2

A

- Diagramme de classe du modèle de données utilisé pour l'échange des niveaux entre PC et VR
- Présentation des deux repositories git des projets PC et VR avec les stats git insight

B

- Un seul des deux éléments

D

- Aucun des deux éléments

UI/UX RV – coefficient 1

A

- Cohérence du mapping des boutons du contrôleur
- Lisibilité des textes
- Le jury estime que la proposition permet d'éditer un niveau de manière cohérente et agréable

B

- 2 des trois éléments

C

- 1 des 3 éléments

D

- aucun des 3 éléments

Format de la soutenance



Présentation - 25 minutes

- Contexte
- Gestion de projet et git
- Présentation UX
- Présentation du modèle pour gestion des niveaux et JSON
- Vidéos de présentation des fonctionnalités développées.

Q/R en groupe - 5 minutes

Démonstration et tests des fonctionnalités choisies par le jury -
15 minutes

Q/R en individuel - 5 minutes par personne

Individuel – note des pairs - 5 minutes

Délibération - 10 minutes

MERCI

CESI ■ LINEACH

BANQUE DE PICTOS

