



# Piscine Unity - D01

## Physique 2D, Tags, Layers et Scènes

Staff [staff@staff.42.fr](mailto:staff@staff.42.fr)

*Résumé: Ce document contient le sujet du jour 01 de la piscine Unity de [42](#).*

# Table des matières

<b>I</b>	<b>Consignes generales</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>4</b>
<b>III</b>	<b>Exercice 00 : Thomas et ses amis</b>	<b>5</b>
<b>IV</b>	<b>Exercice 01 : Par ici la sortie !</b>	<b>7</b>
<b>V</b>	<b>Exercice 02 : Niveau 2 !</b>	<b>9</b>
<b>VI</b>	<b>Exercice 03 : Interactivité</b>	<b>10</b>
<b>VII</b>	<b>Exercice 04 : Des boutons !</b>	<b>11</b>
<b>VIII</b>	<b>Exercice 05 : Un jeu mortel !</b>	<b>12</b>

# Chapitre I

## Consignes generales








- La piscine Unity est à faire entièrement et obligatoirement en **C#** uniquement. Pas de Javascript/Unityscript, de Boo ou autres horreurs.
- L'utilisation de fonctions ou de namespaces non autorises explicitement dans le header des exercices ou dans les regles de la journee sera considéré comme de la triche.
- Pour une utilisation optimale de Unity, vous devez travailler sur le ~/goinfre, qui est en local sur le mac que vous utilisez. Pensez à bien récupérer vos projets avant de vous delog car le goinfre local est vidé régulièrement.
- Contrairement aux autres piscines, chaque journée ne demande pas un dossier ex00/, ex01/, ..., exn/. A la place pour la piscine Unity, vous devrez rendre votre dossier projet qui aura pour nom le nom de la journee : d00/, d01/, .... Toutefois, un dossier de projet contient par default un sous-dossiers inutile : le sous-dossier "projet/Temp/". Assurez-vous de ne **JAMAIS** pusher ce dossier dans votre rendu.
- Au cas ou vous vous poseriez la question, il n'y a pas de norme imposée à 42 pour le **C#** pendant cette piscine Unity. Vous pouvez utiliser le style qui vous plaît sans restriction. Mais rappelez-vous qu'un code que votre peer-evaluateur ne peut pas lire est un code qu'elle ou il ne peut noter.
- Vous devez trier les assets de votre projet par dossier. Chaque dossier correspond à un et un seul type d'asset. Par exemple : "Scripts/", "Scenes/", "Sprites/", "Prefabs/", "Sounds/", "Models/", ...
- Assurez-vous de tester attentivement les prototypes fournis chaque jour. Ils vous aideront beaucoup dans la compréhension du sujet et du travail attendu.
- L'utilisation de l'Asset Store d'Unity est interdite. Vous êtes encouragés à utiliser les assets fournis chaque jour (quand nécessaire) ou à en chercher d'autres sur le net s'ils ne vous plaisent pas, sauf bien entendu pour les scripts car vous devez avoir écrit tout ce que vous rendez (hors scripts fournis par le staff, obviously). L'Asset Store est interdit car quasiment tout le travail que vous avez à faire s'y trouve déjà sous une forme ou sous une autre. Néanmoins l'utilisation des Standard Assets de Unity est autorisée voir meme conseillée pour certains exercices.

- Pour les corrections à partir du d03 il vous sera demandé de build les jeux pour les tester. **C'est le correcteur** qui doit build le jeu vous devez donc évidemment toujours push vos projets/sources. De ce fait votre projet doit correctement configuré pour le build. **Aucun** réglage de dernière minute ne doit être toléré.
- Important : Vous ne serez pas évalués par un programme, sauf si le contraire est explicite dans le sujet. Cela implique donc un certain degré de liberté dans la façon que vous choisissiez de faire les exercices. Toutefois, gardez en tête les consignes de chaque exercice, et ne soyez pas FAINÉANTS, vous passeriez à côté de beaucoup de choses intéressantes.
- Ce n'est pas grave d'avoir des fichiers supplémentaires ou inutiles dans votre dossier de rendu. Vous pouvez choisir de séparer votre code en différents fichiers au lieu d'un seul, sauf si le header d'un exercice mentionne explicitement les fichiers à rendre. Un fichier ne doit définir qu'un et un seul comportement, pas de namespaces donc. Toute cette consigne ne s'applique bien évidemment pas au sous-dossier "projet/Temp/" qui n'a pas le droit d'exister dans vos rendus.
- Lisez le sujet en entier avant de commencer. Vraiment, faites-le.
- Le sujet pourra être modifié jusqu'à 4h avant le rendu.
- Même si le sujet d'un exercice est relativement court, ça vaut le coup de passer un peu de temps à comprendre parfaitement le travail attendu pour le faire au mieux.
- Parfois il vous sera demandé un soin particulier sur la qualité artistique de votre rendu. Dans ce cas, cela sera mentionné explicitement dans le sujet correspondant. N'hésitez alors pas à tester plein de choses différentes pour vous donner une idée des possibilités offertes par Unity.
- Par Odin, par Thor ! Réfléchissez !!!
-

# Chapitre II


## Foreword

Today's starring :

-  Thomas - Lonely and slightly naive, Thomas likes to list his observations about the world - and is absolutely fantastic at falling.
-  Chris - Pessimistic, irritable and suspicious - Chris might not be the best jumper, but he was doing just fine on his own.
-  John - Rather proud of his agility and sportiness, John quite likes an audience so decides to look after Thomas and Chris.
-  Claire - Claire lacks confidence, she moves slowly and considers herself rubbish at jumping. But then she discovers that she might be a superhero.
-  Laura - Laura isn't great at jumping, although she does have her own unique ability - which, sadly, she's too ashamed to tell anyone about. An ominous pixel cloud has been following her around lately, and this worries the others.
-  James - James had always been different. Not least because of his unique disregard for Newtonian laws.
-  Sarah - On a quest to find the fountain of knowledge and learn the truth about her world, Sarah sees herself as rather more intelligent than the other "lesser" quadrilaterals.

# Chapitre III

## Exercice 00 : Thomas et ses amis

	Exercice : 00
Exercice 00 : Thomas et ses amis	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : Le fichier de scène "ex00", un script "playerScript_ex00" sur chaque personnage, un script pour la caméra	
Fonctions interdites : Aucune	
Remarques : n/a	



ATTENTION ! A partir d'aujourd'hui et jusqu'à la fin de la piscine, vous devez rendre tous vos exercices dans Assets/ comme expliqué dans les consignes générales et PAS dans des dossiers ex00, ex01 ... Ne tenez PAS compte de la ligne "Dossier de rendu" du cartouche des exercices, il est automatiquement rempli lors de la création du pdf

Pour commencer, lancez la démo pour vous faire une idée de ce qu'on va faire aujourd'hui. Vous pouvez déplacer les personnages avec les flèches du clavier et la barre espace pour sauter. Les touches 1, 2 et 3 permettent de changer de personnage actif. L'entraide entre les personnages est obligatoire pour réussir les niveaux !

Petit exercice d'échauffement. Créez une scène avec un sol, une caméra et 3 personnages : Claire, John et Thomas.

On doit pouvoir sélectionner un personnage en appuyant sur 1, 2 ou 3 et le déplacer à l'aide des flèches. La caméra se centre automatiquement sur le personnage sélectionné.

Chaque personnage doit pouvoir sauter, se déplacer vers la gauche et vers la droite. On doit pouvoir réinitialiser la scène en appuyant sur une touche, par exemple la touche R ou la touche Backspace.




Attention ! Vous devez impérativement créer UN SEUL script qui sera appliqué sur les 3 personnages.



Vous êtes libres de créer d'autres scripts indépendants, pour la gestion de la caméra par exemple.

# Chapitre IV

## Exercice 01 : Par ici la sortie !

	Exercice : 01
Exercice 01 : Par ici la sortie !	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : La scène "ex01", un script "playerScript_ex01" sur chaque personnage, et tout ce qui vous semble pertinent	
Fonctions interdites : Aucune	
Remarques : n/a	



Unity n'aimant pas les doublons dans les scripts vous devriez créer un `playerScript_ex01` et y copier votre code de l'exercice précédent pour pouvoir le compléter avec ce nouvel exercice.

Maintenant les personnages doivent également posséder des caractéristiques différentes :

- Claire, le carré bleu avance moins vite et saute moins haut que les autres.
- John, le baton jaune avance plus vite et saute plus haut que les autres.
- Thomas se situe entre les deux avec une vitesse et une hauteur de saut moyennes.



**Vous devez toujours impérativement utiliser le même script pour tous vos personnages.**

Il ne doivent pas pouvoir sauter plusieurs fois sans retomber sur une surface - sol ou autre personnage - donc pas de jump infini ou de wall jump !




Ils doivent traverser un premier niveau qui les force à coopérer et atteindre la sortie. La sortie de chaque personnage est indiquée par un contour le représentant. Lorsque chaque personnage est aligné sur sa sortie le joueur a réussi le niveau. Vous devez afficher un message l'indiquant - un simple debug peut suffire pour le moment.



Le niveau doit absolument forcer la coopération donc il ne doit pas être possible pour les personnages d'arriver à la sortie indépendamment les uns des autres.

# Chapitre V

## Exercice 02 : Niveau 2 !


	Exercice : 02
Exercice 02 : Niveau 2 !	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : La scène "ex02" et tout ce qui vous semble pertinent	
Fonctions interdites : Aucune	
Remarques : n/a	

Vous devez créer un second niveau en utilisant les layers physiques dans le level design :

- Les plateformes sont soit blanches soit de la couleur d'un personnage.
- Les personnages ne peuvent utiliser que les plateformes blanches ou de leur couleur, ils passent au travers des autres.
- Vous devez relier les deux niveaux entre eux, de façon à passer d'un niveau à l'autre lorsque les personnages réussissent un niveau.

# Chapitre VI

## Exercice 03 : Interactivité

	Exercice : 03
Exercice 03 : Interactivité	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : La scène "ex03" et tout ce qui vous semble pertinent	
Fonctions interdites : Aucune	
Remarques : n/a	

C'est maintenant que ça va devenir intéressant ! Créez un niveau avec en plus des téléporteurs et des plateformes mouvantes.


A vous de voir si vous voulez faire des passages speed tout en technique ou alors des ascenseurs sadiques comme les vieux jeux NES où il faut attendre 10 secondes pour avoir le bon alignement des plateformes. Le but est vraiment de créer des niveaux intéressants à parcourir et pas seulement une démonstration technique.



N'oubliez pas d'ajouter vos niveaux au build au fur et à mesure.

# Chapitre VII

## Exercice 04 : Des boutons !

	Exercice : 04
Exercice 04 : Des boutons !	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : La scène "ex04" et tout ce qui vous semble pertinent	
Fonctions interdites : Aucune	
Remarques : n/a	

Maintenant créez un niveau avec des switches qui ouvrent des portes. Mieux encore, des switches de couleur qui ouvrent des portes de leur couleur.


Ou encore mieux, des switches blancs qui prennent la couleur du personnage qui les active et qui ouvrent les portes de la couleur en question...

Et enfin, des switches qui changent la couleur des plateformes, donc le chemin que peuvent prendre les personnages.

Vous l'aurez compris, vous êtes assez libre dans le level design, tant qu'il y a des boutons qui fonctionnent comme expliqué au dessus.

# Chapitre VIII

## Exercice 05 : Un jeu mortel !

	Exercice : 05
Exercice 05 : Un jeu mortel !	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : La scène "ex05" et tout ce qui vous semble pertinent	
Fonctions interdites : Aucune	
Remarques : n/a	

Pour le dernier niveau on va monter un peu la difficulté, pour que le joueur ait moins l'impression de se promener :

- Créez des tourelles de couleur avec tir régulier. Le tir ne touche que les personnages de la même couleur.
- Créez des pièges, sous forme de zone au sol ou d'obstacle aérien.
- Créez des trous. La caméra ne suit pas un personnage qui tombe dans un trou.
- Si un personnage est touché par un tir de tourelle, touche un piège ou tombe dans un trou c'est Game Over.



Il est possible de tout coder avec les notions d'hier et d'aujourd'hui donc n'utilisez pas d'autres systèmes, notamment les actions timées et coroutines, pour gérer les tirs de tourelle. Vous verrez tout ça plus tard.