

Programmation Unity

ATELIER: ANDROID

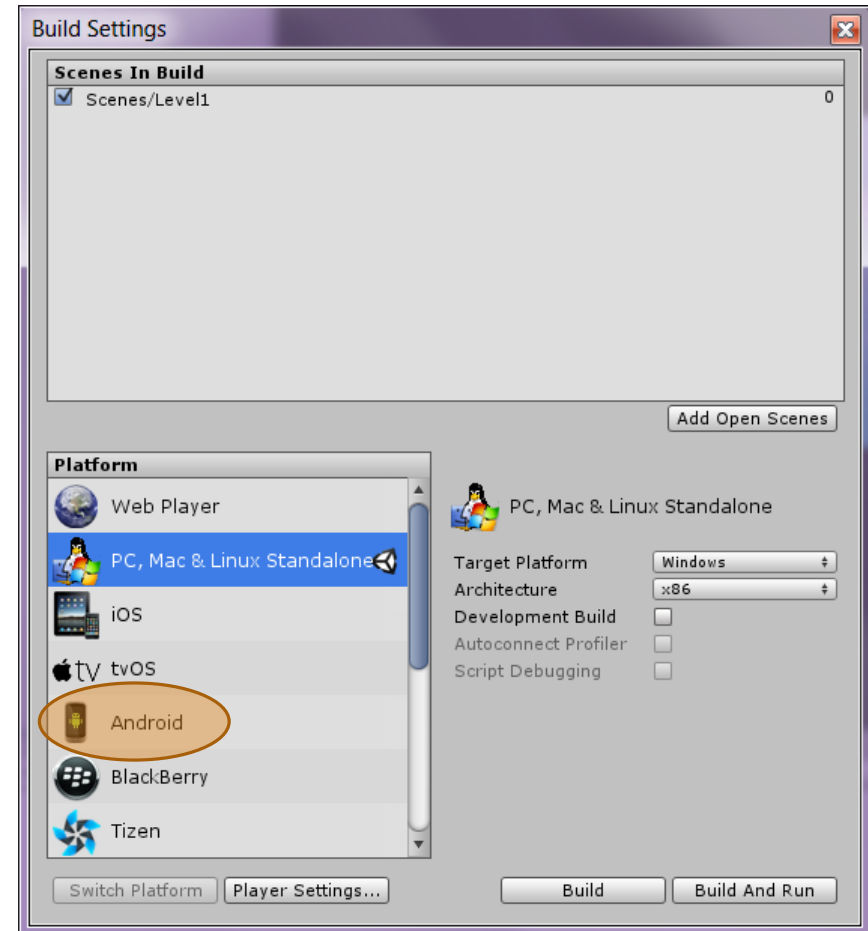
CONFIGURATIONS

Configuration de l'environnement – 1

Vérifier que la compilation pour Android est activée :

Sélectionner Android. Si il indique « no module found » vous ne l'avez pas activé.

Si elle n'est pas activée, il faut retélécharger l'installateur correspondant à votre version d'Unity et le lancer. Lors de l'installation, décocher tout et cocher uniquement l'option pour pouvoir compiler pour Android. Pendant ce temps, lisez bien la suite du TP !



Configuration de l'environnement – 2

Installez [SDK Tool](#) pour Android. SDK Tools Only devrait suffire. Android Studio rajoute un IDE pour développer (une alternative à Mono ou Visual mais qui n'est pas forcément supportée par Unity). Attention à utiliser une version 32/64 bit correspondant à votre machine !

Installez au besoin le [JDK \(Java Development Kit\)](#) version 6 ou ultérieur correspondant à votre machine pour pouvoir exécuter SDK Tool.

Allez [ici](#) pour les instructions d'installation. Aller jusqu'à l'opération 4 incluse de Adding SDK Packages. A l'opération 4, il faut ajouter au moins une plateforme avec une API 2.3 ou plus. Il faut aussi les drivers USB et le Platform Tools.

Ceci peut se faire en parallèle de l'installation du module Android d'Unity.

Configuration de l'environnement – 3

Ouvrez Unity et allez dans File -> Build. Sélectionnez Android. Lancez le build de la scène vide. Unity vous demandera le chemin vers le SDK Android que vous avez renseigné à la slide précédente. Il ne demande ceci que la première fois.

Si l'erreur suivante survient :

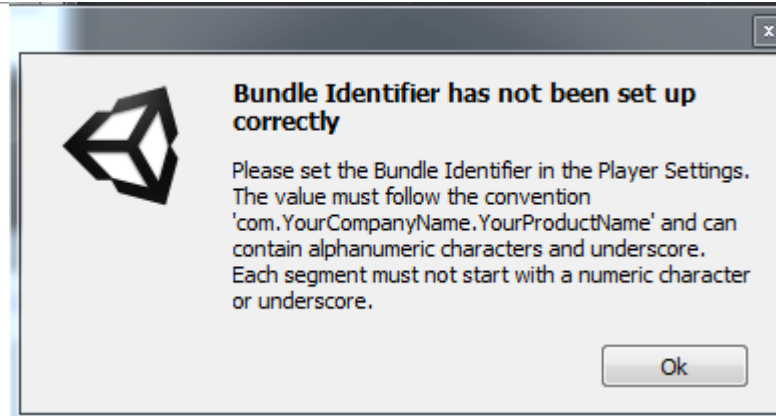


C'est que le chemin d'accès n'a pas été mis à jour dans votre système d'exploitation. Attention à faire très exactement la manipulation suivante **(DANS LE DOUTE, APPEL L'ENSEIGNANT!!)** :

- Allez dans les variables d'environnement :
 - **Win7** : Ouvrir le menu démarrer et faire un clic droit sur Ordinateur, Propriété. Modifier les paramètres. Puis dans Paramètre systèmes avancez, cliquez sur Variables d'environnement
 - **Win8 / Win10** : Chercher « système (panneau de configuration) ». Allez dans « paramètres système avancés » puis cliquez sur « variables d'environnement ».
- Ajoutez une nouvelle variable utilisateur PATH si elle n'existe pas.
- Ajoutez le chemin vers le <JDK>\bin à PATH sans supprimer les éventuels autres chemins. (; pour séparer)

Configuration de l'environnement – 4

Maintenant, vous devriez voir l'erreur suivante :



Les applications Android et de manière générale, les applications java (car Android c'est du java, rappelons-le) sont toutes identifiées par un package. Un package, c'est une chaîne de caractères séparée par des points. Imaginez l'arborescence d'un dossier, remplacez les slashes par des points et ça vous donnera un ordre d'idée de ce que ça représente.

Il vous faut donc identifier votre projet par un package. Sachant que les packages doivent être uniques sur le Play Store, plutôt que d'avoir à le changer à la dernière minute pour pouvoir le publier, autant le faire tout de suite en utilisant la convention de nommage classique : `com.nomDeSociete.nomDuProjet`.

Edit -> Project Settings -> Player. Pour Android, dans le champ Other Settings, renseigner le champ Bundle Settings : `com.ICAN.TPAndroid-<initialesdugroupe>`

Configuration du smartphone Android

Activez le USB Debugging sur votre smartphone Android :

- Cliquez plusieurs fois sur le Settings -> About Phone -> Build Version débloquera le menu Settings -> Developer
- Dans Settings -> Developer, activez USB Debugging.

Alternativement, on pourra créer une APK qu'on transfèrera sur notre smartphone et installera grâce à une application tierce comme [APKInstaller](#).

Dans tout les cas, il faut autoriser l'installation d'application de source inconnues dans les paramètres de sécurité. En effet, pour tester un build qui n'est pas encore validé par le Store, vous en aurez besoin. (Pensez à retirer cette option après le TP si vous n'en avez plus besoin, pour des raisons de sécurité)

Accessoirement, on pourra ajouter des applications tel [Astro](#) pour parcourir la carte mémoire de votre Android afin de faciliter le débogage.

Ceci peut se faire en parallèle de l'installation du module Android d'Unity et du SDK Android.

Première scène Android

Réalisez une scène simple :

- Un cube qui tourne sur lui-même.

Installez l'application sur votre smartphone et lancez la. Si tout ce passe bien, vous avez validé l'installation, c'est-à-dire l'étape la plus difficile à mon sens pour développer sous Android. Faites valider cette étape par l'enseignant.

DÉVELOPPEMENT MOBILE

Généralité sur Android

La tailles des écrans est variables d'un Android à l'autre. Faites toujours très attention à utiliser une UI adaptative. Utilisez les ancres !!

Il n'est pas permis de quitter une application depuis ellemême. En effet, dans les standards d'Android, il est clairement établie que l'utilisateur devra tuer l'application via le manager d'application. De ce fait, la commande `Application.Quit()` n'aura pas le même effet sur PC et sur Android. Au mieux, elle mettra l'application en arrière plan.

Les types dynamiques (le mot clef `var`) sont désactivés sur Android. Ils peuvent donc causez des erreurs de compilations et/ou d'exécution. Il faut donc les proscrire.

Fonctionnalité de l'appareil – caméra

L'accès à la caméra se fait par l'intermédiaire de la classe [WebCamTexture](#).

Essayez d'appliquez le script d'exemple fourni dans le lien dans une image (UI) et sur un cube.

WebCamTexture fonctionne aussi avec les webcam de votre PC.

Classiquement, les Android ont souvent deux caméras. Vous pouvez retrouver quelle est la bonne caméra grâce au booléen `isFrontFacing` de [WebcamDevice](#). Essayez de mettre la caméra frontale sur l'image.

Il est possible d'acquérir une image du flux vidéo avec la méthode [GetPixels32](#).

Fonctionnalité de l'appareil – Accéléromètre et gyroscope

On peut directement accéder à l'accéléromètre via la class Input, grâce au vecteur accélération :

```
transform.Translate(Input.acceleration.x, 0, -Input.acceleration.z);
```

Cette instruction permet de déplacer un objet selon l'horizontale de l'écran et dans la profondeur en fonction de l'orientation de l'appareil. L'horizontale de l'écran est défini suivant si l'application est joué en portrait ou en paysage. Ceci peut se régler dans les Player Settings pour Android.

La différence avec un gyroscope est que l'on ne prend pas en compte la différence mais uniquement le positionnement. Vous pouvez trouver un exemple d'initialisation de gyroscope [ici](#)(tips pour Unity4 et antérieur).

L'accès au gyroscope se fait par [Input.gyro](#) en regardant par exemple **attitude** pour faire de la VR. Il est nécessaire d'avoir un positionnement de référence pour calibrer le gyroscope.

Fonctionnalité de l'appareil – Tactile

Techniquement, utiliser un doigt sur l'écran est la même chose que de cliquer avec la souris. Ceci dit, il est possible de détecter plus d'évènements afin d'être plus précis.

L'utilisation de plusieurs doigts touchant l'écran se récupère ainsi :

```
Touch myTouch = Input.GetTouch(idx);          // idx est l'index à considérer : 0 pour le premier doigt  
                                              Input.touchCount est le nb
```

myTouch.fingerId est l'index dans le tableau de doigts GetTouch

myTouch.position correspond aux coordonnées en pixel depuis le coin inférieur gauche.

myTouch.deltaPosition est la différence de position du doigt entre deux frames

myTouch.deltaTime est le temps mis depuis la dernière update du Touch

myTouch.tapCount est le nombre de clic donné par le doigt (frappe de l'écran au même endroit)

myTouch.phase peut prendre les valeurs suivante { Began, Canceled, Ended, Moved, Stationary }

Publication de l'application

Pour soumettre l'application sur iOS, vous pouvez [suivre ce tutorial](#).

Et [ici](#) pour l'android (Play Store).

TRAVAIL À RENDRE

Application à rendre – EnigmOn like

Mécaniques :

1. Réalisez une scène 3D avec la caméra en son centre.
2. L'application devra pouvoir effectuer un zoom de la scène en utilisant les conventions classiques smartphone (avec deux doigts qui s'écartent ou se rapprochent).
 - Pensez à utiliser `camera.orthographicSize` ou `camera.fieldOfView` suivant le type de caméra.
3. En double cliquant sur l'écran, la caméra avancera dans la profondeur d'une distance fixée.
4. La caméra devra tourner en même temps que le smartphone tourne.
5. En finissant un niveau, on passe au suivant. (chargement de scène)

Niveaux :

1. ColorMatcher :

1. Il y a moult objets 3D ici (30? Cube?/sphère?). Les objets de la scène devront avoir pour couleur, la couleur dominante (moyenne) dans l'image acquise par la caméra arrière, sauf un objet (ou assemblage d'objet) : une arche qui aura une couleur fixée.
2. À chaque frame ou toutes les secondes, on updatera la couleur moyenne acquise.
3. Le but du niveau est d'ouvrir l'arche en s'arrangeant pour que tous les objets soit à peu près de la couleur de l'arche (une erreur de 10% est tolérée) et de passer sous l'arche ouverte. Un feedback de victoire doit être effectué (genre texte « Vous avez résolu l'énigme ! » ou son).
4. Pour la marge d'erreur, regarder si : $\text{abs}(R - R_{\text{moyen}}) + \text{abs}(G - G_{\text{moyen}}) + \text{abs}(B - B_{\text{moyen}}) < 0,1 * 3$, Sinon comparer la hue dans l'espace de couleur HSV.

2. MultiToucher :

1. Afficher 3 cubes à l'écran (1 Rouge, 1 Vert et 1 Bleu) variant suivant l'intensité de la couleur dominante acquise par la caméra dans chacun des trois canaux de couleur. Au dessus de ces trois cubes, trois autres cubes de valeur fixée (1 Rouge, 1 Vert et 1 Bleu). Tout les cubes sont ici fixe quelque soit la rotation du téléphone.
2. Le but est encore une fois que le joueur film avec la caméra un objet ayant à peu près la même couleur que la couleur de référence mais cette fois ci on lui affiche la décomposition de chaque canal couleur (RGB). Il devra valider les trois canaux séparément ou en même temps.
3. Pour valider un canal, il devra drag'n'droper le cube sur le cube référent (au dessus). Si le canal correspond, les cubes fusionnent. Sinon le cube revient à sa position d'origine.
4. Une fois tout les canaux validés un feedback de victoire doit être effectué (genre texte « Vous avez résolu l'énigme ! » ou son).

Modalité de rendu

A rendre à l'adresse suivante : remi.slysz@gmail.com

Le nom du mail devra être : [ProgrammationUnity] TPAndroid <noms des membres du groupe>

Le rendu est pour la semaine prochaine (le 11 novembre à 23h59 au plus tard).

Dans le mail de rendu, j'attends:

- Une archive (.rar ou .zip) du projet
- Une application .apk
- Un document expliquant la répartition des tâches au sein du groupe et pourquoi les avoir réparties ainsi. Ce document devra aussi expliquer succinctement comment le code est organisé et pourquoi. Cette explication doit être compréhensible sans aucune ligne de code.