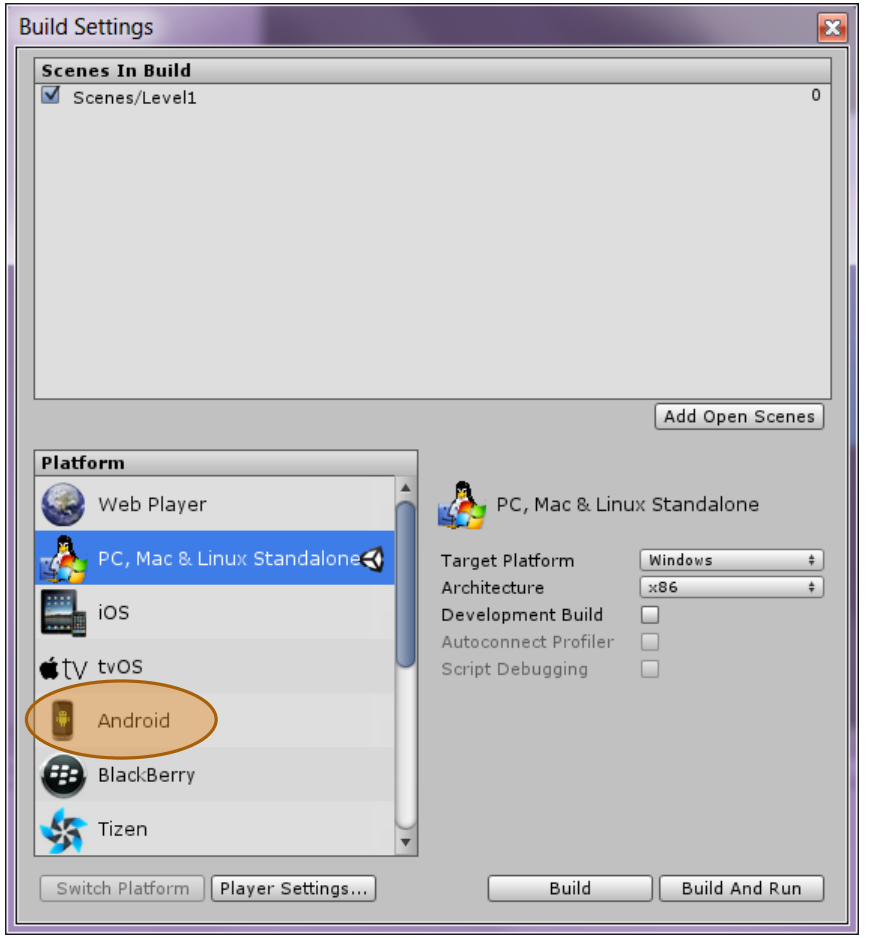
Programmation Unity

ANDROID

Configuration de l’environnement – 1

Vérifier que la compilation pour Android est activée :



Si ce n’est pas le cas, il faut malheureusement réinstaller Unity. Lors de l’installation, pensez bien à cocher 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 forcement supportée par Unity). Attention à utiliser une version 32/64 bit correspondant à votre machine !

Installez le JDK (Java Devlopment 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 la réinstallation d'Unity.

Configuration de l’environnement – 3

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

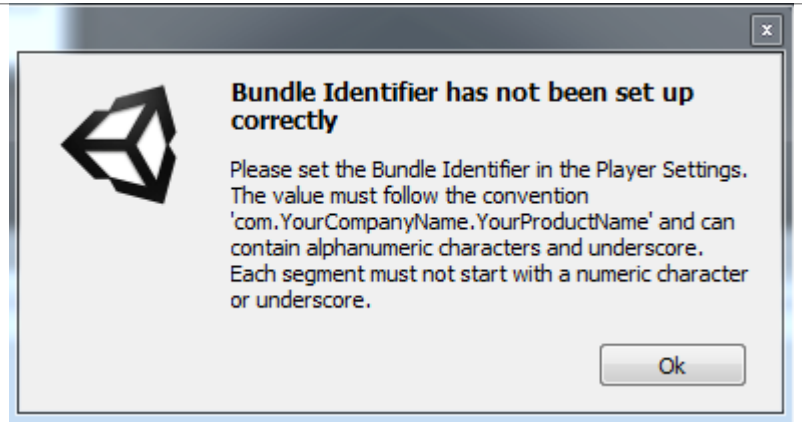
Si l’erreur suivante survient :

C’est que le chemin d’accès n’a pas était mis à jour dans votre système d’exploitation. Attention à faire très exactement la manipulation suivante :

- Allez dans les variables d’environnement
  - Pour 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.
  - Pour Win10, tapper simplement « path » dans la bare de recherche pour trouver les propriété systèmes
- Ajoutez une nouvelle variable utilisateur PATH si elle n’éxiste 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 slashs 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 faire comme tout le monde et utiliser 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.TP2-<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.

Accessoirement, on pourra ajouter des applications tel Astro pour parcourir la carte mémoire de votre Android afin de facilité le débogage.

Cette étape peux se faire en parallèle de la réinstallation d'Unity.

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 validez cette étape par l'enseignant.

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 !!

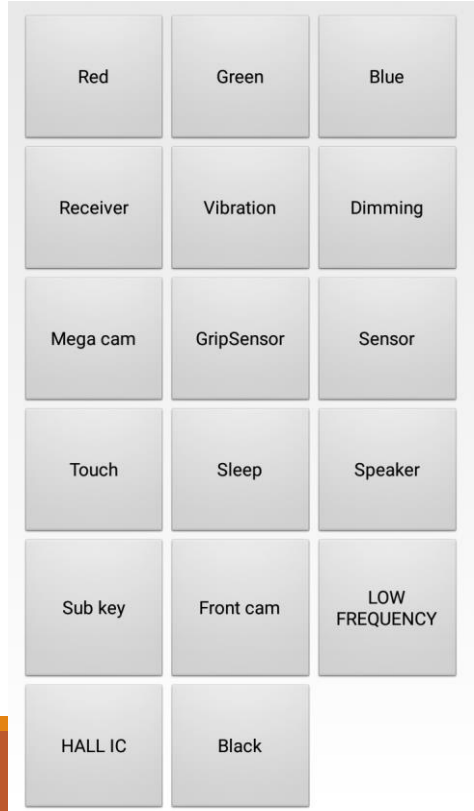
Il n'est pas permis de quitter une application depuis elle-même. En effet, dans les standards d'Android, il est clairement établie que l'utilisateur devra tuer l'application via le manageur d'application. De ce fait, la commande Application.Quit() n'aura pas le même effet sur PC et sur Android. Au mieux,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

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

Connaitre les informations matériels de votre androïd

Tout les android ont une série de code « secret » dépendant de la marque.Il permettent d'accéder à différent test interne.

Par example, sur les Samsung, si on appelle ce numéro : ***#0*#**
on objtent le menu de test suivant :



Sensor permet entre autre de vérifier si votre appareil est équiper d'un giroscope ou non et s'il fonctionne

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.

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'évènement 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 max

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

myTouch.position correspond aux coordonnées en pixel depuis le coint 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 doigts (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 store.

Application à rendre – EnigmOn like

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 certaine distance.

4 - La caméra devra tourner en même temps que le smartphone tourne. Précisez en texte à l'écran la technologie utilisée (gyroscope ou autre).

5 - Les objets de la scène devront avoir pour couleur, la couleur dominante (moyenne) dans l'image acquise par la caméra arrière, sauf une arche décomposé en morceau qui aura une couleur fixe.

- À chaque frame ou toutes les secondes, on updatera la couleur moyenne acquise.

6 – L'arche doit être construite de manière à ne voir une arche que d'un certain point de vue.

7 - Le but du jeu 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 d'avoir le point de vue adéquate pour visualiser l'arche. Un feedback de victoire doit être effectué (genre texte « Vous avez résolu l'énigme ! »).

- Pour l'erreur, regarder si la Hue obtenable avec **Color.RGBToHSV** est éloigné de 10% de la Hue de l'arche.

Modalité de rendu

Rendre une archive (.rar, .zip, .tar.gz, .7z) à l'adresse suivante : remi.slysz@gmail.com

L'archive devra s'appeler 3GD<AouB>_Enigmon_<noms des membres du groupe> soit par exemple : 3GBA_Enigmon_Athos_Portos_Aramis.rar

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

Le rendu est pour la fin du cours. Allez jusqu'où vous pouvez.

Le travail se fera en groupe de 4 personnes maximum.

Dans le mail de rendu, j'attends une explication succincte sur la façon dont a été réalisé le TP. Cette explication doit être compréhensible sans ligne de code.