

# TP Tooling 10%

Le but de ce TP est de faire un tout petit outil vous permettant d'automatiser un peu le packaging de vos jeux. Vous allez vous rendre compte pendant la session que tout outil vous permettra d'accélérer vos développements de jeu par un facteur non négligeable.

## Consignes

Ça ne compile pas => 0

À rendre pour le mardi 4 février

Vous pouvez choisir votre langage de programmation dans la liste suivante :

- C++
- C#
- Rust
- Golan

Votre rendu doit être un GitHub PUBLIC par personne

Ce TP exceptionnellement se fera seul

## Objectif

Développer un petit outil en ligne de commande capable de :

- Lire un fichier UPROJECT (2 pts)
  - Afficher le nom du jeu
  - Afficher la version de Unreal utilisée
    - Marquer « From Source » si cela est le cas
  - Afficher les plugins utilisés
- Compiler un projet Unreal (3 pts)
  - En utilisant UBT
- Packager un projet Unreal (5 pts)
  - En utilisant UAT

Je ne vous demande pas d'interface graphique. Néanmoins, j'offre jusqu'à 5 points bonus pour les GUI. Attention, une simple GUI ne suffira pas, je jugerai l'expérience utilisateur de celle-ci tout comme son extension pour les futurs développements.

## Utilisation de votre outil

Pour vous simplifier la vie, votre outils sera TOUJOURS lancé depuis la racine d'un build from source (donc pour accéder au dossier Engine vous pouvez faire ./Engine).

Je testerai ces trois commandes :

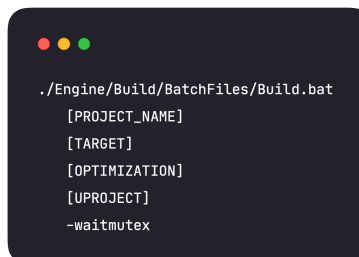
MyTool [CHEMIN DU UPROJECT] show-infos

MyTool [CHEMIN DU UPROJECT] build

MyTool [CHEMIN DU UPROJECT] package [CHEMIN DU PACKAGE]

## Quelques informations pour vous aider

- Comment je lance une ligne de commande depuis un programme ?
  - C# : <https://tinyurl.com/4vtazyzm>
  - C++ : <https://tinyurl.com/mr2cuc9m>
  - Rust : <https://tinyurl.com/yb24etyw>
  - Golang : <https://tinyurl.com/3hs3hdxv>
- Comment lire le fichier UPROJECT ?
  - Le fichier UPROJECT est un fichier JSON, tous les langages proposés ont des librairies de premier plan pour lire du JSON. Aller regarder le fichier et trouvez la bonne librairie pour votre programme
- Comment compiler avec UBT ?
  - Pour Windows :



```
./Engine/Build/BatchFiles/Build.bat  
[PROJECT_NAME]  
[TARGET]  
[OPTIMIZATION]  
[UPROJECT]  
-waitmutex
```

- Pour Mac :



```
./Engine/Build/BatchFiles/Mac/Build.sh  
[PROJECT_NAME]  
[TARGET]  
[OPTIMIZATION]  
[UPROJECT]  
-waitmutex
```

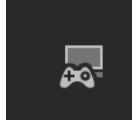
- Comment lancer UAT ?
  - Pour Mac



- Pour Windows



- Comment bien utiliser UAT ? Les paramètres sont très compliqués
  - Vous allez vous servir du **Project Launcher** de Unreal Engine
  - Dans votre éditeur : Tools -> Project Launcher
  - Ajoutez un Custom Launch Profile (gros bouton Add en bas)
    - Section Project
      - Vérifier que c'est le bon Project
    - Section Build
      - Choisissez **Build** au lieu de **Detect Automatically**
    - Section Cook
      - Choisissez **By the book** au lieu de **On the fly**
      - Choisissez votre plateforme (Windows ou Mac)
      - Choisissez vos Maps
    - Section Package
      - Choisissez **Package and store locally** au lieu de **Do Not Package**
    - Section Deploy
      - Choisissez **Do not deploy** au lieu de **Copy to device**
  - Revenez dans le menu précédent (en appuyant sur le bouton **Back** en haut)
  - Clic droit sur votre profile -> Rename
    - Donnez-lui un nom intéressant
  - Un bouton « **Launch this profile** » devrait être là (sinon corrigez votre configuration)



- Appuyez dessus
- Une nouvelle fenêtre devrait se lancer
  - Dans l'**Output Log** de cette fenêtre, tout en haut, cherchez **Parsing Command Line** :
  - Ce qui suit est la command-line à passer à UAT pour votre packaging process