

Video Game porting via API calls hooking



Feral Interactive Limited



Plan

- Feral Interactive
- Problématique
- Tâches principales
 - DiRT4
 - Améliorations du projet principal
 - Menu principal du projet principal
- Bilan du stage
 - Technique
 - Professionnel
 - Personnel
- Conclusion

Feral Interactive Limited

Feral Interactive

Où ?



Quoi ?

Portage de jeux vidéos

Bibliothèques de traduction des appels aux API graphiques

macOS



iOS



Qui ?

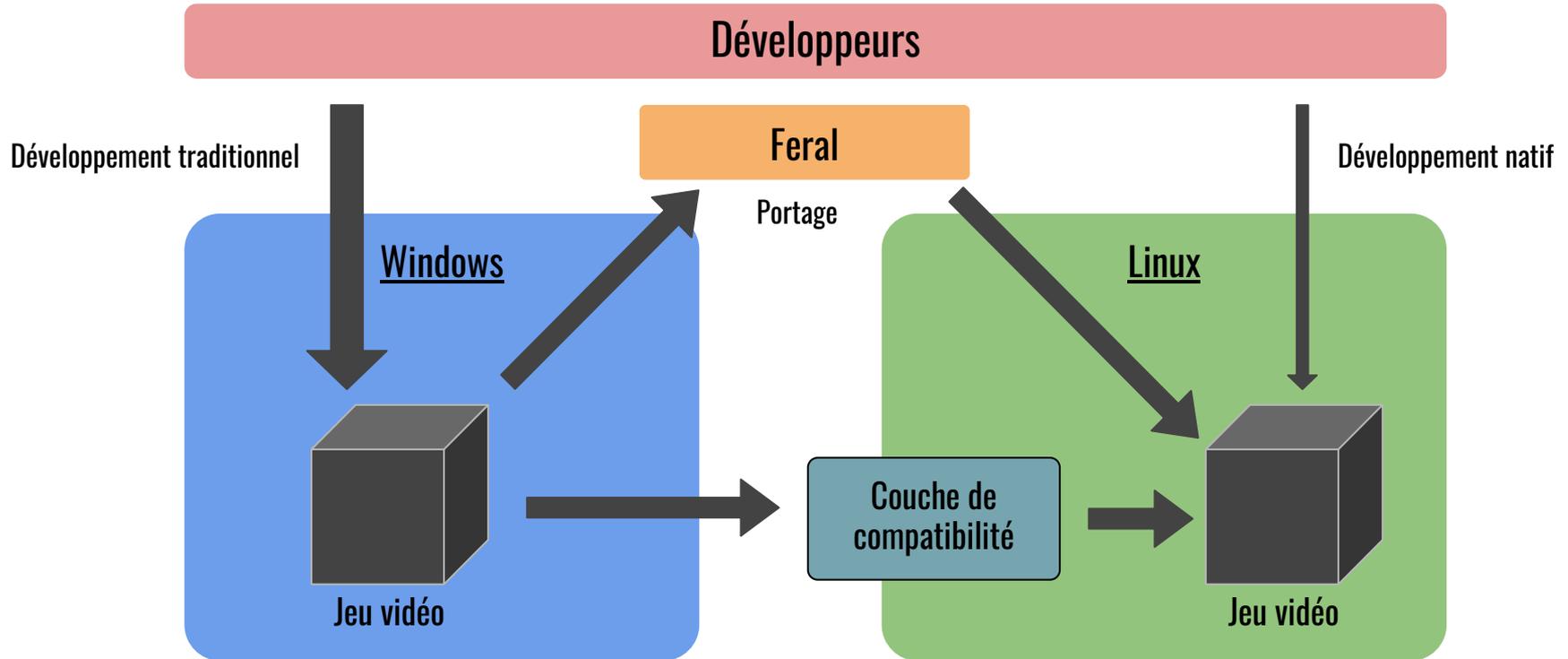
~60 personnes dont ~30 développeurs

Développeurs spécialisés par projets

QA (testeurs), artistes, traducteurs, community managers

Problématique

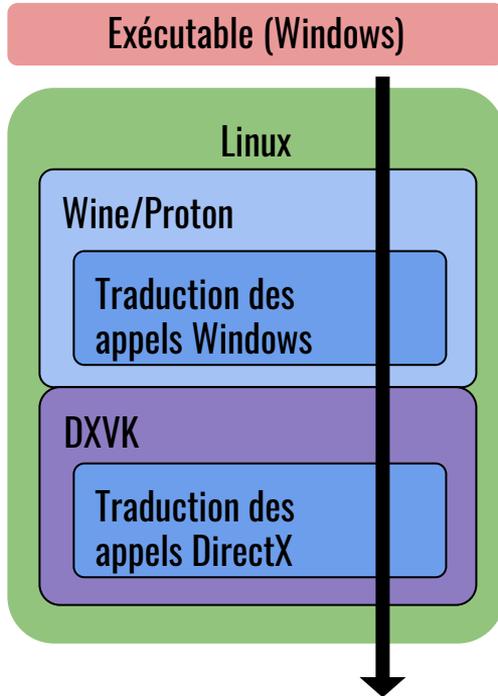
Problématique - Contexte



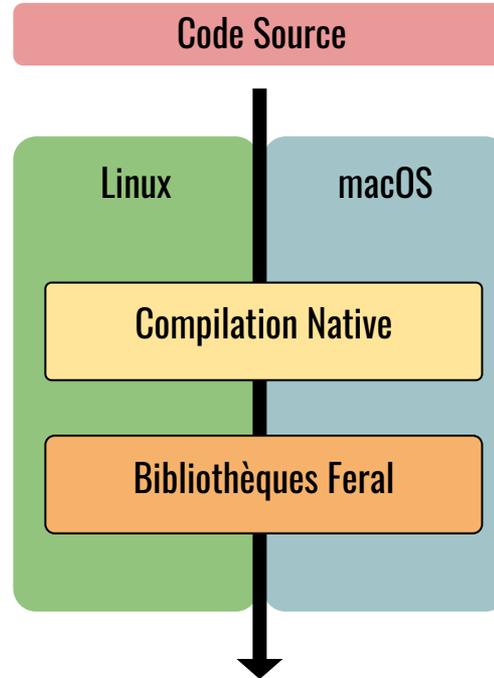
Problématique - Principe

Couche de compatibilité

ex: Linux



Portage



Problématique - Sujet

Intitulé: "Portage de jeu vidéo par hooking d'appels aux API graphiques"

Affectation: "Partie Linux du portage d'un jeu non annoncé"

Dans les faits: "Développement multi-plateforme générique"

Raisons:

- Bibliothèques de traduction déjà implantées et optimisées
- Reprise d'un travail préalable de portage du dit jeu
- Ré-évaluation de la gestion de projet du dit jeu

Tâches principales

Tâches principales



Jeu de courses de **rallye**

Développé et édité par **Codemasters**

Sorti en **Juin 2017** sur consoles et Windows

Introduction aux méthodes et outils de l'entreprise

Développement exclusivement sous macOS

Corrections apportées:

- Sauvegarde d'options graphiques
- Gestion de mémoire
- Correspondance contrôleur - saisie en jeu
- Interface du "Launcher"
- Interface en jeu

Sortie le 28 Mars 2019 sur macOS et Linux

Tâches principales - Améliorations du projet principal

Ré-organisation des Headers

Problème:

Frameworks Apple en collision avec le code
CoreFoundation et CoreGraphics inclus
dans les bibliothèques

Solution:

Centraliser les inclusions problématique

Classe "interface":

- Isoler le code du jeu des bibliothèques trop bas niveau
- N'organiser les headers qu'une seule fois

Parallélisation - Proof of concept

Problème:

Optimiser le calcul de déplacement
Le "path-finding" et la résolution d'état
occupe la plupart du calcul d'une image

Solution:

Considérer l'optimisation et le LTO

Finalement utilisation de plusieurs threads

Code incompatible et résultats mitigés

Tâches principales - Menu principal du projet principal

Refonte graphique du menu principal

- Édition de la disposition des menus
- Correction et amélioration de la logique du code
- Ajout du matériel graphique re-travaillé

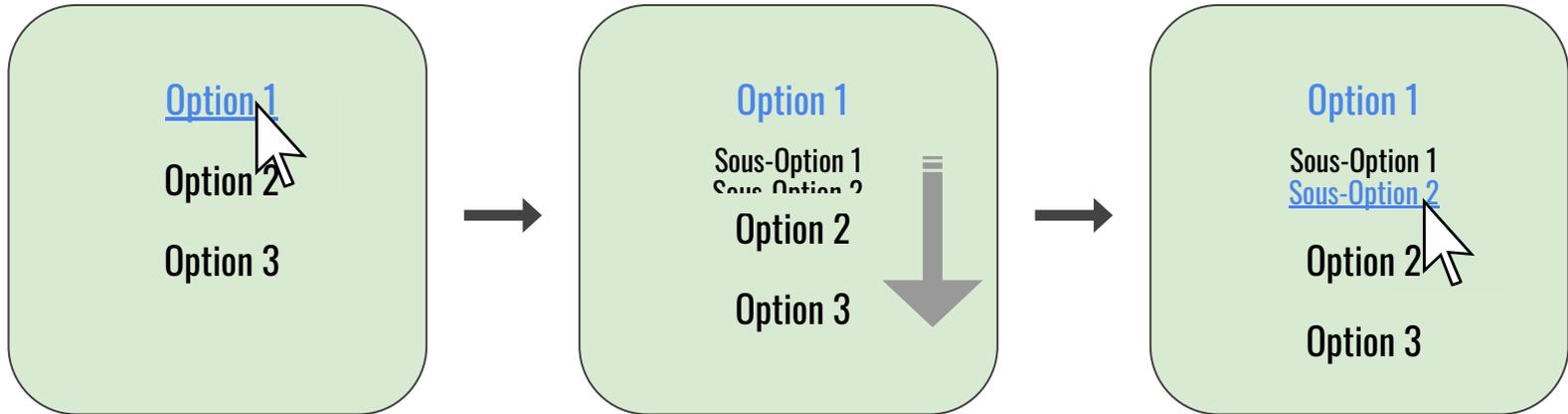
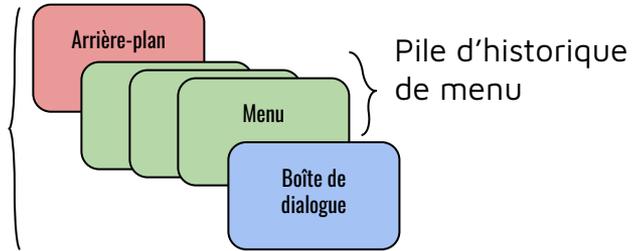
De mi-juin à septembre, durée de la tâche particulièrement sous-estimé

Tâches principales - Menu principal du projet principal

Menu latéral coulissant

Mécanisme d'origine incompatible avec la refonte

Priorité des
conteneurs



Réponse du menu latéral à un événement de clic

Bilan du stage

Bilan - Technique

Amélioration des compétences en C++

Aspect multi-plateforme

Apprentissage de nouveaux outils



XCode Instruments



RenderDoc



TexturePacker

Bilan - Professionnel

Participation à la campagne de **débug** avant la **distribution** d'un jeu

Forte implication dans le **développement**, **correction** et **refonte** d'un autre jeu

Travail en coopération avec des **équipes techniques** comme **artistiques**

Application et amélioration des notions en **gestion de projet**

Bilan - Personnel

Expérience forte: vivre à l'étranger, amélioration en Anglais

Découverte plus en profondeur de la **production** d'un **jeu vidéo**

Participation à mon humble échelle à la fortification du marché **Linux**

Souhait de continuer en **rendu temps-réel**

Conclusion

Proposition d'emploi acceptée

Annonce du jeu prochainement

Requête de développement en rendu prise en compte

- Développeur "*Graphics*"
- Développeur Vulkan

Questions