

Le logiciel **Kirpi** et son utilisation en rig chez **Superprod**



Thierry Lauthelier
thierry.lauthelier@caleido-scop.com
<https://www.caleido-scop.com>



SUPERPROD
ANIMATION

Enrico Ferraris
eferraris@superprod.net
<https://www.superprod.net>

Présentation **RADI-RAF**

Sommaire

CALEIDO-SCOP. Présentation de Kirpi

- Programmation visuelle généraliste
- Intégration dans Maya
- Application à un rig Superprod

Retour d'expérience de SUPERPROD

- Superprod intro
- Présentation des besoins de Superprod
- Pistes envisagées
- Bilan des améliorations
- Retour sur Kirpi

Présentation de Caleido-scop

Domaines d'expertises

- Recherche et développement
 - Dossiers de Recherche
 - Logiciels commerciaux / in-house
- Conseil & Formation

Partenaires Techniques



Références



ILLUMINATION
MACGUFF



TamTO



MIKROS
ANIMATION

Fix
Studio

DWARF
ANIMATION STUDIO

france•tv
studio

ELLIPSE ANIME
PRODUCTIONS



cube
CREATIVE PRODUCTIONS

isotropix

BEASTMAKERS NIGHTSHIFT

Caleido

Computer Graphics R&D

- Société Coopérative
- Création en 2010
- 5 salariés (vétérans de la 3D)
- Accrédité CIP

lescop
SOCIÉTÉS COOPÉRATIVES
ET PARTICIPATIVES



Caleido-scop

255 avenue de Jouques

13400 Aubagne

Tél. 09 53 0173 88

<https://www.caleido-scop.com>

Présentation de Kirpi

Les Besoins

- Complexité ↗ + Budget ↘
- Des développements très variés avec des technologies et des logiciels différents:
 - IT (parc machines, comptes utilisateurs, render farm, etc.)
 - TD (rig, fx, pipeline, etc.)
 - R&D (plugins, logiciels in-house, etc.)
- **Pérenniser** ses solutions dans le temps:
 - Indépendance vis à vis des solutions commerciales.
 - Portabilité dans le temps et entre logiciels.
- **Accélérer** le processus de développement (conception utilisation maintenance).
- Améliorer **l'efficacité / les performances** .
- Préserver le savoir-faire (**confidentialité**).

Présentation de Kirpi

L'existant

- Solutions propriétaires fermées.
 - Nécessité d'apprendre un nouveau système / langage.
 - Un environnement de développement pauvre (debug, profiling, etc.).
 - Une intégration complexe.
- Peu ou pas de:
 - Confidentialité.
 - Portabilité.
 - Pérennité.
 - Efficacité.

Présentation de Kirpi - Écosystème

Pur
C++

Programmation
Visuelle

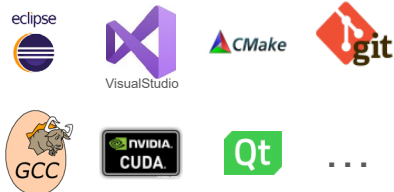
Compilation
Just in Time



Présentation de Kirpi - Écosystème

Pur
C++

Outils de développement



Outils de debug / profiling



Tout l'univers OpenSource et +



Outils de multi-threading



Facilité d'intégration

- ✓ Pas de binding.
- ✓ Code externe / interne.

Des outils puissants et éprouvés

Performances

Pérennité

Portabilité

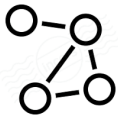
Indépendance

Présentation de Kirpi - Écosystème

Programmation Visuelle

Pour tous !

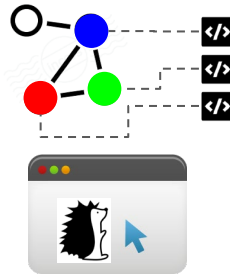
Utilisateurs généralistes



Utilisateurs techniques



Équipes de dev



Une API ouverte

- Des **composants Ui** facilement réutilisables et extensibles.
- **Intégration facile** dans n'importe quel DCC ou logiciel propriétaire.



Généraliste

Intégration facile

Présentation de Kirpi - Écosystème

Compilation
Just in Time

Écriture / compilation à la volée



Optimisé pour chaque machine

- ✓ pré-compilation bytecode
- ✓ finalisation sur chaque architecture machine.

Noeuds de workflow puissants

- ✓ parallel
- ✓ vectorizer
- ✓ if..then..else...

Prototypage rapide

Inférence de types

Efficacité

conception utilisation maintenance

Présentation de Kirpi - Écosystème

Pur
C++

Programmation
Visuelle

Compilation
Just in Time



Kirpi

Des liens qui libèrent...

Un ensemble d'outils

- **kpedit** - Édition interactive de noeuds et de graphes.
- **kpexec** - Exécution de graphes depuis la ligne de commande.
- **nodegen** - Génération facile de noeuds depuis une librairie externe.

Possibilité d'exporter ses graphes:

- en **Blackbox** (cache l'intelligence du graphe).
- en **pur C++** (plus besoin de kirpi).

Une politique de licence simple:

- **Besoin** d'une licence pour **éditer** des graphes.
- **Pas besoin** de licence pour **exécuter** des graphes.

Confidentialité

Indépendance

Simplicité

CRÉATION D'UN GRAPHE



PROFILING
&
MULTI-THREADING



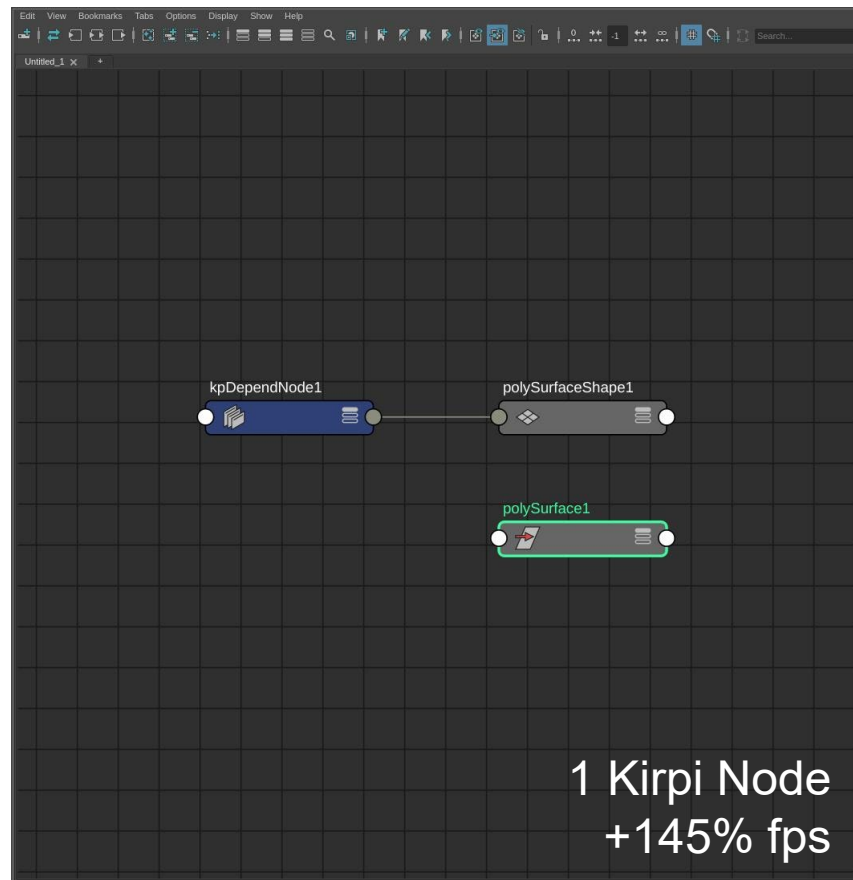
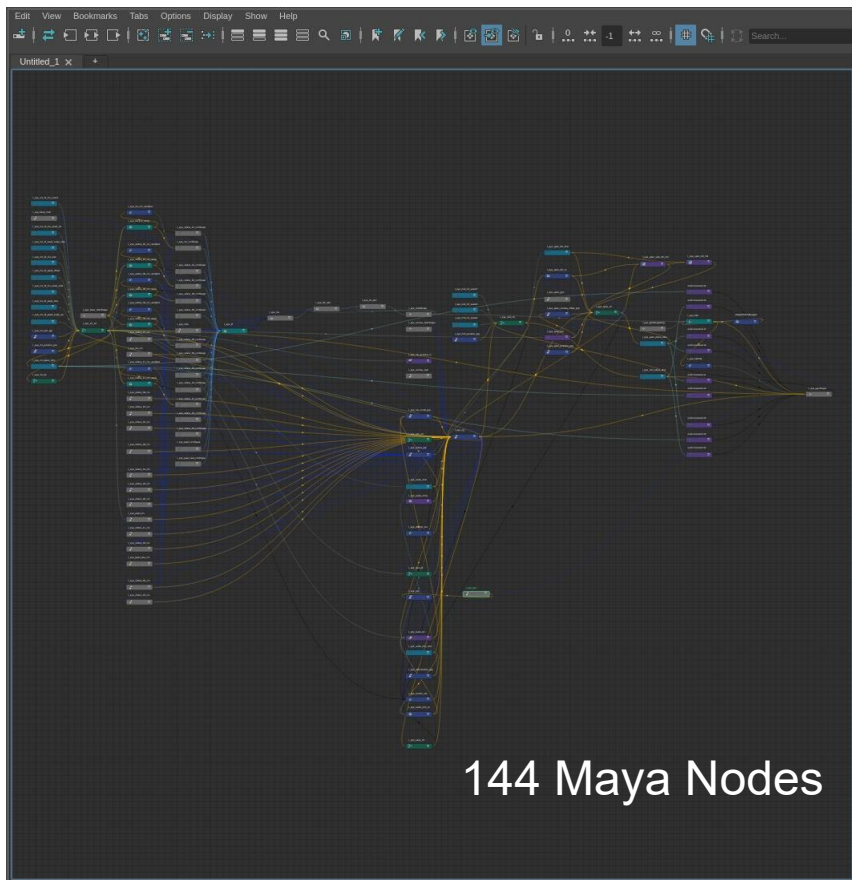
INTEGRATION DANS MAYA



INTEGRATION DANS MAYA
Exemple du Rig de l'oeil



Intégration dans Maya © - Exemple du rig de l'oeil



Kirpi - Aller plus loin...

ALLER PLUS LOIN...





SUPERPROD
ANIMATION



france-tv



PASSION®



max



NETFLIX



france-tv



THE GROUP



SUPERPROD
ANIMATION



SUPERPROD
FILMS



REDMONK
STUDIO



SUPERPROD
DRAMA



SUPERIGHTS
SALES



4I0HZ
MUSIC



THE CO-PRODUCTION
COMPANY

AWARDS

(WINNER)



(NOMINEE)



(OFFICIAL SELECTION)



PREVIOUS WORK





Besoins de Superprod

- Amélioration des performances des rig
- Stabilisation des scènes :
 - Perte occasionnelle de la parallel evaluation
- Simplification de rig
- Amélioration des déformations:
 - Meilleure propagation du scale dans les modules complexes (Bras, jambes, spine..).





Besoins de Superprod

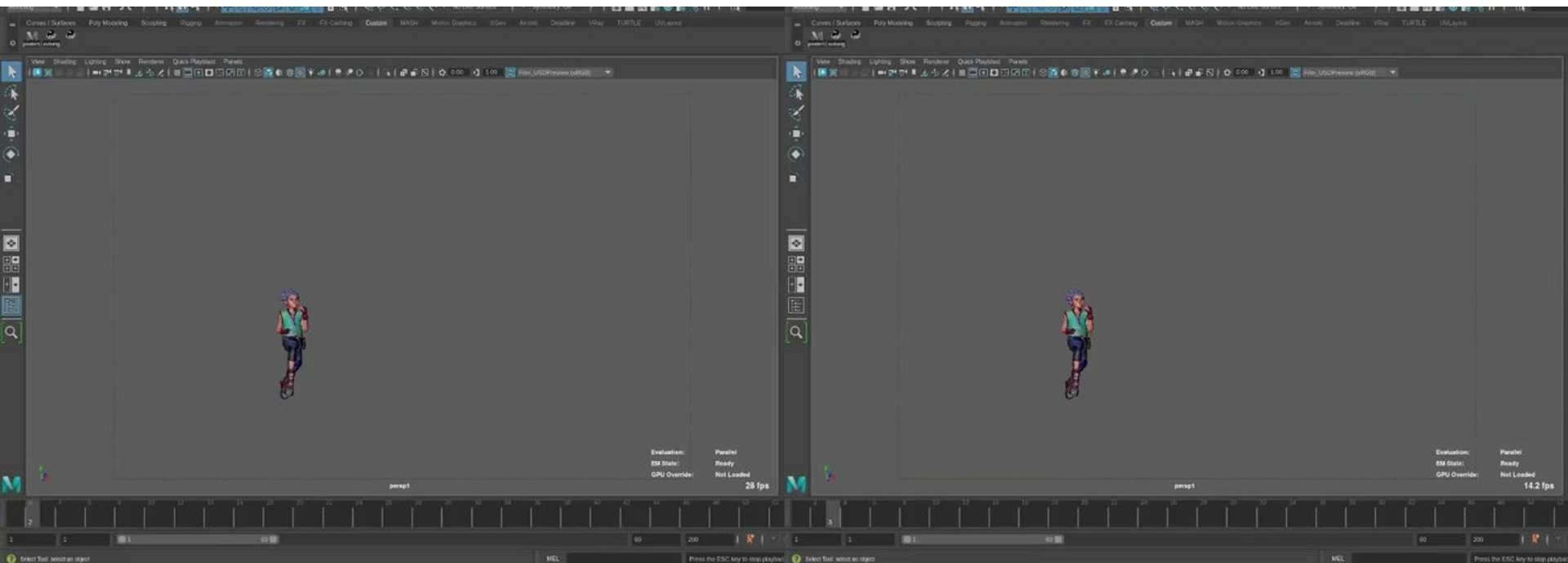
- Amélioration des performances des rig
- Stabilisation des scènes :
 - Perte occasionnelle de la parallel evaluation
- Simplification de rig
- Amélioration des déformations:
 - Meilleure propagation du scale dans les modules complexes (Bras, jambes, spine..).

Premier test sur matrix constraints

Conversion du système nodal au plug-in python

- 9 nodes par constraint
- 150 / 200 constraints par rig
- Gain de ~1500 nodes

Matrix Constraint en Plugin Python



systeme nodal

26 fps

plugin python

14 fps



Besoins de Superprod

- Amélioration des performances des rig
- Stabilisation des scènes :
 - Perte occasionnelle de la parallel evaluation
- Simplification de rig
- Amélioration des déformations:
 - Meilleure propagation du scale dans les modules complexes (Bras, jambes, spine..).

Nouvelles pistes envisagées

Nodes C++

- Matrix Constraint
- Angle Extractor
- Eyes
- Sticky Lips
- Ribbon

Optimisations

- Parallélisation des Rivets
- Parallélisation des Sticky Lips
- Compression de la chaîne de déformation



Développement des nodes C++

- Formation à Kirpi
- Définition du cahier des charges
- Partenariat pour le développement des nodes
- Test/Debug
- Mise en production
- Feedback sur Kirpi en tant que premiers utilisateurs





Gain de performances

- Des gains de performances sur chaque système isolé
- Remarquable gain sur des rig complets

Stabilisation des scènes

d'animation

- On n'a plus de passage en serial
- Identification du coupable

Amélioration des déformations

- Gestion du scale sur des chaînes complexes

Simplification des rig

- Réduction du nombre de nodes
- Poids des scènes
- Temps de build des modules de l'autorig





Gain de performances isolés



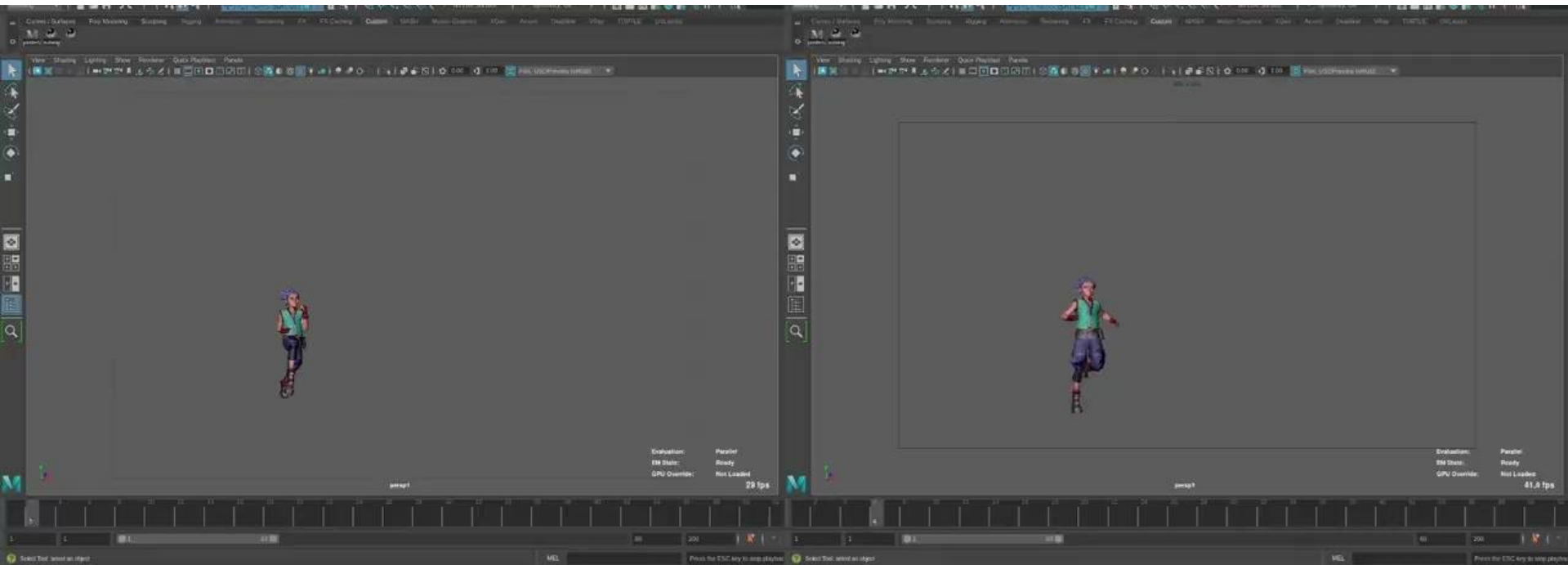
Node C++

- Matrix Constraint +25%
 - Comparé au node python +400%
- Angle Extractor +50%
- Eyes +145%
- Sticky Lips +10%
- Ribbon (arm/leg et spine) +20%

Optimisations des chaînes de déformations

- Gains moyen de 25% sur un rig complet

Before/After



Before

26 fps

After

40 fps

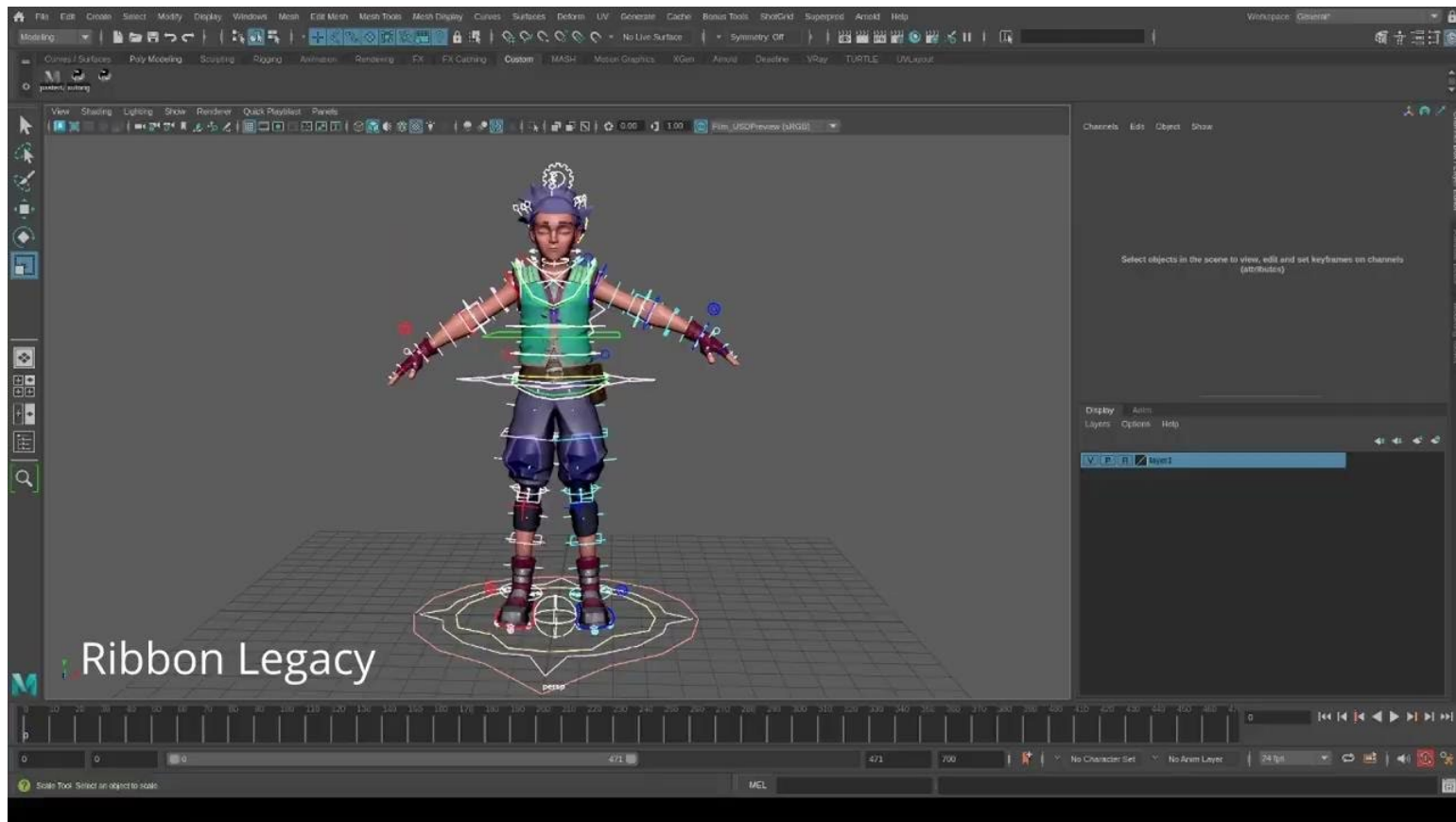


Passages en serial evaluation

- On n'a plus de passages en serial sur les nouvelles séries
- Le problème a continué sur les séries utilisant les anciens rig
- Dialogue avec autodesk
- Bug du node wtaddMatrix du système nodal de la matrix constraint



Amélioration des déformations



Simplification des rig



Nombre de node

- Sticky Lips 337 -> 1
- Eye 144 -> 1
- Arm/Leg 120 ->1
- Total 15500 -> 11500
- ~ 15 MB

Temps de build de l'autorig

- Sticky Lips -50%
- Eye -55%
- Arm/Leg -28%
- ChainIK -20%





Retour d'utilisation de Kirpi

- Gros avantage dans le prototypage, possibilité de tester live les modifications du code ou du graph sans besoin de recompiler à l'extérieur
- Plus de facilité dans la créations de node C++ grâce au système nodal
- Possibilité de création d'une librairie de node custom
- Flexibilité de l'output: graph kirpi, compilation node

Inconvénients

- Nécessite des connaissances de C++
Mais en simplifiant le processus de compilation

Attentes pour le futur

- Une plus grande librairie de base
- Une communauté d'utilisateurs pour le partage d'expérience

Merci !

Des Questions ?



Thierry Lauthelier
thierry.lauthelier@caleido-scop.com
<https://www.caleido-scop.com>



SUPERPROD
ANIMATION

Enrico Ferraris
eferraris@superprod.net
<https://www.superprod.net>