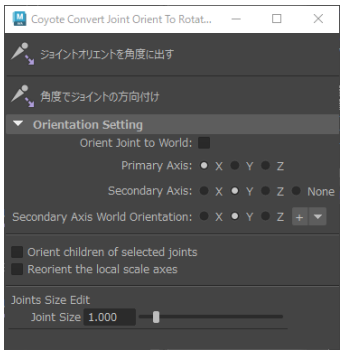
COYOTE 3DCG STUDIO所属、TAの小林です。
新人である自分が最初に携わったBOOTH対応について、新人目線での良かった点／苦戦した点などを紹介できればと思います。

まず初めに、軽く自己紹介の方をさせていただきます。
C&R社に当初は２Dアニメーターとして中途で入社しました。
2020年のCEDECにてTAという職種に興味を持ち、今までの経験を活かしつつTAを目指すため、３Dアニメーターへ転向し、さらに現在TA業務を対応しています。
プログラミングも未経験の自分が、TA業務の最初の仕事として対応させて頂いたのが、こちらのBOOTH対応になります。

早速ですが、今まで対応させて頂いた４つのツールについて、それぞれ行った対応や工夫、学んだことなど紹介させて頂きます。

「Joint Orient To Rot」

- **対応したこと**
  - 命名の変更
  - 警告文の追加
- **学んだこと**
  - Mayaコマンドリファレンス
  - エディタの使い方などの基礎
    - グレップ検索（Ctrl + G）
    - 検索（Ctrl + F）
- **苦戦したこと**
  - このツール専用のoptionVarの用意
  - 警告文の追加



3

一番初めに対応させて頂いたのが、「Joint Orient To Rot」になります。
こちらは**ジョイントの方向を回転に出すよう改良されたツール**になります。

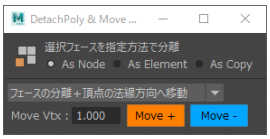
この時点ではあらかじめMelの基礎を教わり、printの使い方やif文／for文等の知識を身に着けている状態になります。
ですが、実際のまとまったコードを見るのは初めてで、処理の中を見ても知らないコマンドばかりなので、
Mayaコマンドリファレンスで調べたり、エディタの「検索」や「グレップ検索」を駆使し、初めて「多くのソースの中から必要な処理をピックアップする方法」を学びました。

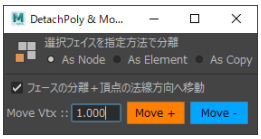
ウィンドウを開きなおしたときも同じ設定を維持するように作成したかったのですが、
元のツールは色んな機能をまとめており、optionVarが複雑だった為、このツール専用にoptionVarを用意する必要がありました。
しかし、ボタンの種類によって、optionVarのコマンドが違うため、苦戦しました。

また、警告文の追加に関しては、エラー文で終わらないように、エラーの出る状況をできるだけ洗い出すことが難しかったです。

「Detach Poly Move」

- **対応したこと**
  - 命名の変更
  - 警告文の追加
- **学んだこと**
  - このツール専用のoptionVarの用意
- **工夫したこと**
  - UIを少し変更
    （上がBooth用、下が元のツール）
- **苦戦したこと**
  - 警告文の追加



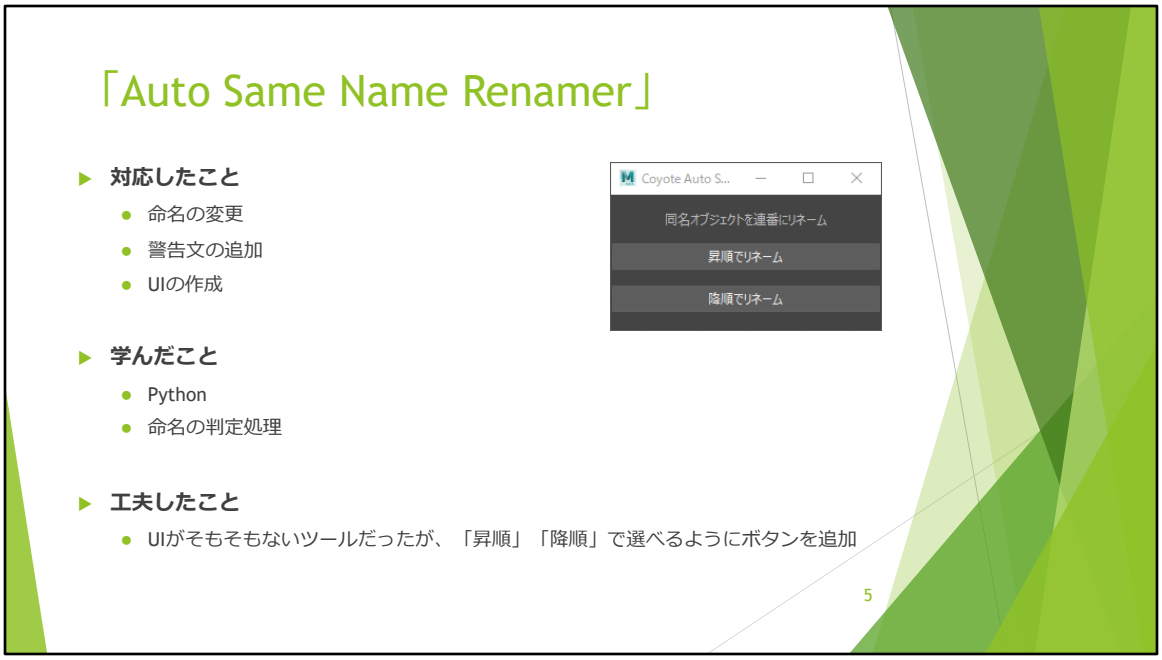


4

次に対応させて頂いたのが、「Detach Poly Move」になります。
こちらは**フェースの分離と移動を同時に行うことができるツール**になります。

こちらでは少し工夫をして、UIに少し手を加えました。
元ツールでは「フェースの分離＋頂点の法線方向へ移動」の所に「チェックが入っている場合」と「入っていない場合」で、
文面が変わるような処理をしていたのですが、プルダウン式の方が分かりやすいのではないかと思ったため変更しました。

また、前回とは違うUIもあったため、optionVarの対応と警告文の追加対応に苦戦しました。



3つ目に対応させて頂いたのが、「Auto Same Name Renamer」になります。
こちらは**同名オブジェクトを自動でリネームするツール**になります。

今まではmelでの作成でしたが、こちらのツールからPythonで対応しました。
同名オブジェクトを判定する処理があるのですが、初めて見た時は正規表現がただの謎の記号にしか見えませんでしたが、
要素を一つ一つかみ砕いて調べたり、教えていただき、正規表現の使い方の一つとしてとても勉強になりました。

元はそもそもUIが無いツールでしたが、自動でリネームを行うので選択肢を持たせるため「昇順」「降順」を選べるようにUIを作成しました。

4つ目に対応させて頂いたのが、5/17に頒布開始しました。「Set Joint Label」になります。
こちらは**ジョイントラベルの命名や表示非表示を行うツール**になります。

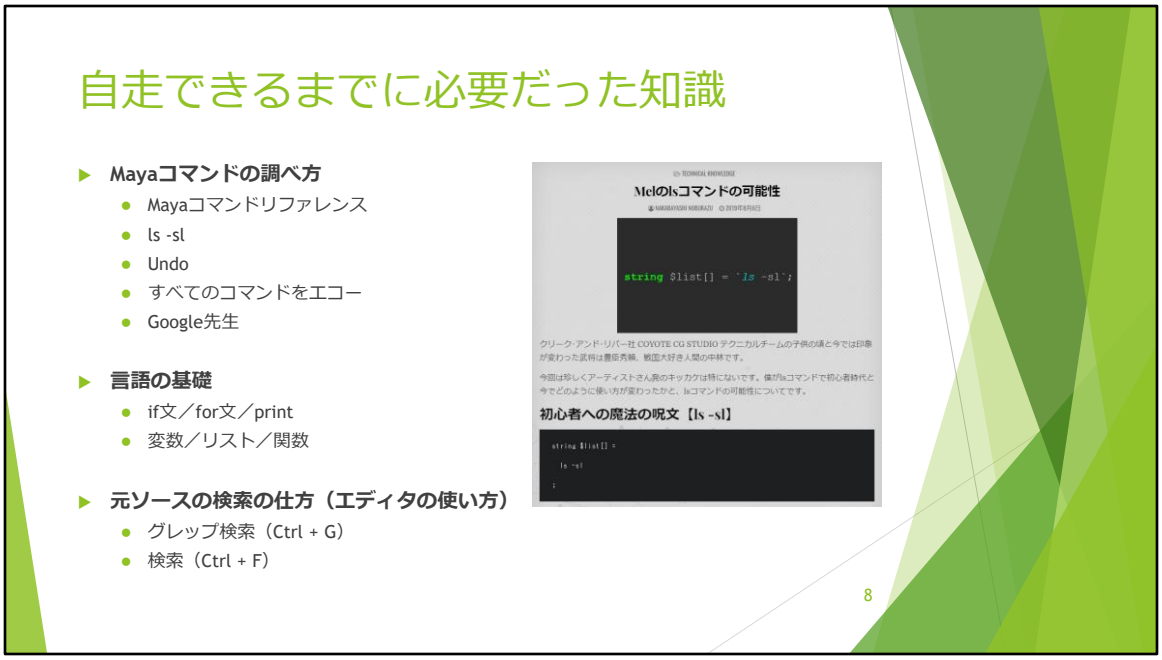
このツールの見どころが、「左右にジョイントがあった場合の命名処理」で、左右判定が幅広く対応できるようになっており、自分にはまだない発想の処理だったのでとても勉強になりました。

また、このツールからQt Designerを初めて使用し、QtでUIを作成した際のファイル構造等を学ぶことができました。

最後に

・自走できるまでに必要だった知識
・
・苦戦したこと
・楽しかったこと

をまとめて紹介させて頂きます。



自走できるまでに必要だった知識としまして、簡潔にまとめるとこちらの3点になります。

・Mayaコマンドの調べ方
・言語の基礎
・元ソースの検索の仕方

この3つを知った時に、ある程度自分で調べて解決ができるようになりました。

Mayaコマンドの調べ方については、（「Mayaコマンドリファレンス」「ls -sl」「Undo」「すべてのコマンドをエコー」「Google先生」）この5つの手段を駆使しました。
また、後で紹介しますが、COYOTEブログにてまとめている記事もとても参考になりました。

言語の基礎について、こちらは「mel」と「Python」で共通なのですが、「if文／for文／print」「変数／リスト／関数」を知った時に、コードの全体像が分かりやすくなり、見え方が特に変わりました。

ツールの**元ソースの検索**の仕方に関して、
こちらは主にエディタの使い方になるのですが、特に「グレップ検索」を
知ってからフォルダ間をまたいでいる処理も探しやすくなりました。

主に「**プログラミングの全体構造**」や「**検索の方法**」を知った時に目の前の
景色が変わったような印象を受けました。

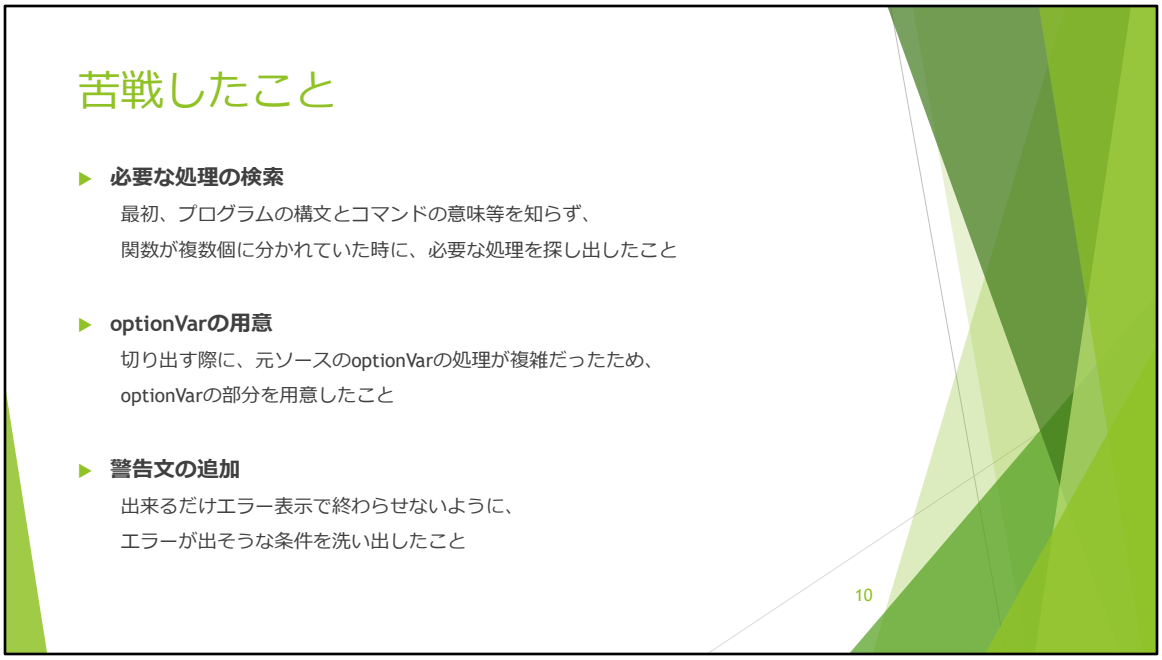
自走にあたって、特に参考にしたCOYOTEブログ記事を4つ、軽く紹介させて頂きます。

【MelのIsコマンドの可能性】
こちらは、Isコマンドの基本的な使い方についてまとめられています。

【Mayaツール開発入門！コマンド調査テクニック】
こちらは、分からないコマンドがあった時の調べ方についていろんな方法がまとめられています。

【Melのglobal変数とoptionVarでメニューの値を保存】
こちらは、自分がとても苦戦した、optionVarの使い方についてまとめられています。

【超初心者向け、MelからPythonへのコマンド変更方法】
こちらは、使用言語が「Mel」から「Python」へ移行した際に、何度もこちらの記事を確認しました。

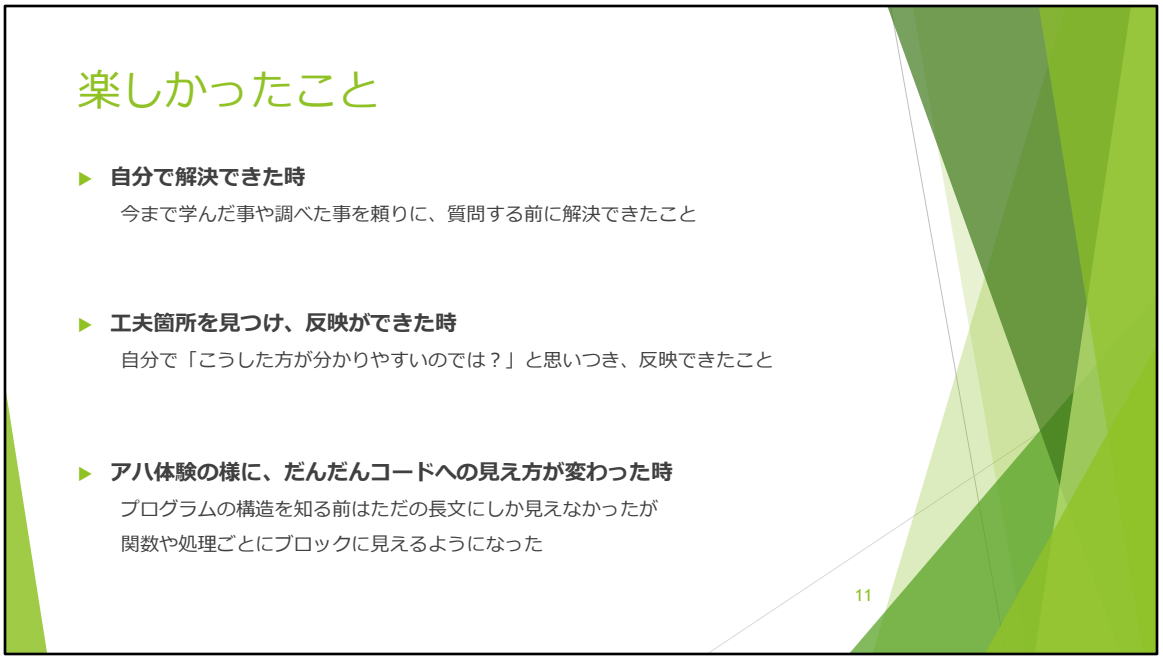


次に、苦戦したことについて、主に苦戦した３つについて紹介させて頂きます。

【必要な処理の検索】
最初の頃は「特に」ですが、プログラムの構文やコマンドに不慣れな時
関数が分かれている場合に、処理のつながりを探すことが難しかったです。

【optionVarの用意】
UIとoptionVarを結びつけるところが、変数の型ごとにコマンドが少し違っていたため、思ったような処理に中々できず苦戦しました。

【警告文の追加】
やさしさ機能として、できるだけエラー表示で終わらせないよう、デザイナーに分かりやすい警告文を心掛けて追加しました。
しかし、エラーが出る条件を洗い出したと思い、テストしたときに新たなエラーを見つけてしまったりして、
まだ甘かった…ということが多々あり、苦戦しました。



さいごに、Booth対応に携わってみて、楽しかったことを３つあげて話をしめたいと思います。

・まず一つ目が、
今まで学んだことや調べたことを頼りに、質問をする前に色々試してみて、自分で解決できた時

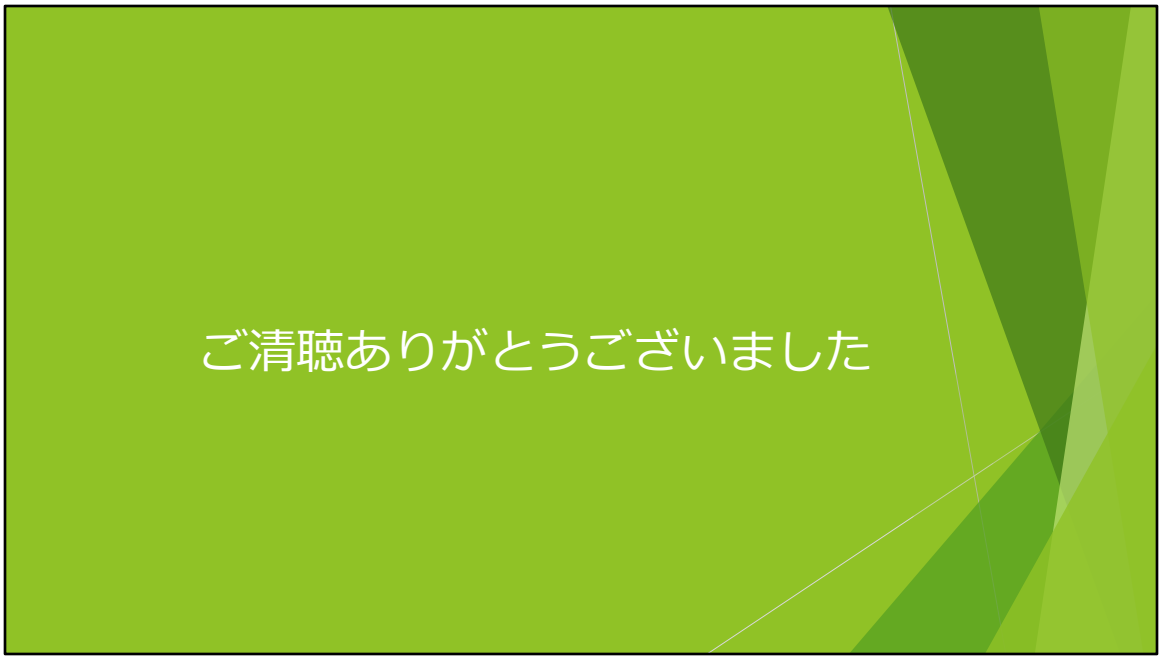
・２つ目が、
自分で「こうした方が分かりやすいのでは？」と工夫箇所を見つけ、反映ができた時

・３つ目が、
プログラムの構造を知る前はただの長文にしか見えなかったのが、アハ体験の様に、だんだん関数や処理ごとにブロックに見えるようになったり、コードへの見え方が変わった時

になります。

TA業務の根幹にもなるかと思いますが、「解決」「工夫」「成長」を自覚で

きる場面があり、苦戦した部分もありましたが、すごく楽しいと感じました。



以上で「**BOOTH対応の新人の取り組み**」についての紹介を終わります。

ご清聴ありがとうございました。