ポケットモンスターＸＹ

　ポケモンモデル　グラフィック表現

■杉森健のイラスト（公式イラスト）の再現■

・ライトテーブルを用いたトゥーンシェーディング

　ライトテーブルは、ハイライト用とトゥーン用の２種を使用しています。

　トゥーンにハイライトが入ることで、より立体感が増します。

　また後述の影カラーテクスチャを用いた影カラーの表現を行っています。

・影カラーテクスチャの採用

　公式イラストに近づけるため、

　影色の減衰は、固定値で行うのではなく、

　影カラーテクスチャを用いることで表現しています。

　光の当たる部分は、通常カラーテクスチャを用いてレンダリングし、

　影になる部分は、影カラーテクスチャを用いてレンダリングしています。

　もちろん容量削減や、持てるテクスチャを別用途で用いたい場合などの理由により、

　固定値を使用する場合もあります。

・ハイライトテクスチャマップを用いた法線マップの加工

　どうしても特定の部位には影を落としたい場合に用います。

　３ＤＳではテクスチャを３枚までしか持てないので、

　カラーマップ、影カラーマップ、法線マップ、ハイライトテクスチャマップと、

　４枚は使用できません。

　そこでデータ出力時にハイライトテクスチャマップを用いて、

　法線マップを加工しています。

　法線マップは一般的に長さが１の法線ベクトルをカラーに変換したものですが、

　この長さに対してハイライトスケールを掛け、カラーに変換しています。

　こうすることで、ライティングの内積計算後にスケールを掛けるのと同等の効果が得られます。

・法線は２つ持つ。

・影モデルの自動生成

・３種のエッジによる輪郭線表現

　１つのエッジシェーダーでは完璧にアウトラインを作れない場合に、

　お互いを補うために用意しています。

　①法線エッジ

　　一般的な法線エッジです。

　　オフスクリーンバッファにモデル法線をカラーとしてレンダリングし、

　　ポスト処理で、それを法線テクスチャとして用い、アウトラインを生成しています。

　　３ＤＳはマルチサンプリングは行えませんが、

　　１回のレンダリングでテクスチャを３枚まで持つことができます。

　　これを用いて、同じ法線テクスチャを３枚分設定し、

　　それぞれのＵＶを少しずつずらすことで、３点サンプリングを表現しています。

　　また、法線ベクトルのカラー化に当たっては、

　　プラグメント処理（コンバイナ処理）の中ではカラーベクトルの正規化ができないため、

　　法線カラーを書き込んだスフィアマップを用意し、

　　これをモデル法線ベクトルから変換したＵＶにてサンプリングしレンダリングしています。

　　これによりかなり精度の高い法線エッジを生成するのに成功しました。

　②頂点カラーを用いたエッジ

　　法線エッジでは出にくいアウトラインを補う為に用意しました。

　　頂点カラーαに色の差分が出るようなグレースケール値を入れておき、

　　法線エッジの際に用いたオフスクリーンバッファのαにその値をレンダリングします。

　　そして法線エッジ同様に、３点サンプリングを行い、

　　周囲のピクセルに差分がある位置にアウトラインをレンダリングするようにしています。

　③ＩＤエッジ

　　一般的なＩＤエッジです。

　　ＩＤ情報はステンシルバッファにレンダリングしています。

　　そしてポスト処理にて、法線エッジ同様３点サンプリングを行い、

　　周囲のピクセルに差分がある位置にアウトラインをレンダリングするようにしています。

・ジオメトリシェーダの使用（ゴース、コータス等で使用）

　体に追従する炎のエフェクトなどに使用しています。

　これを一般的な、エミッタより発生するパーティクルで代用しようとした場合、

　大量のエミッタを配置することになりますが、

　ジオメトリシェーダによりビルボードを生成しているので処理負荷軽減につながります。

　ビルボード生成元の頂点は、モデルのメッシュになりますので、

　スキニングにも完全に追従します。

■その他のポケモンモデルにおけるグラフィック表現■

・フォン・リム・逆光などのライティングに対応。

　金属的な表現や昆虫の羽の質感設定を行う際に使用します。

・スフィアマップ、キューブ環境マップ、投影マップの使用。

　光沢の表現に使用します。投影マップはモデルに模様を投影させるケースで使用します。

・３種の法線方向修正ツール

　なめらかなトゥーンを作り出すためには、モデル法線の調整は不可欠です。

　その為に、任意の頂点法線をコピーペーストする物や、

　選択メッシュに対し、それを覆う任意のメッシュの法線を転写するプラグイン等を

　独自に作成し、使用しています。

・カラーテクスチャに対してのリアルタイムシェーディング

カラーマップに対して、エフェクト効果を掛けられる仕組みになっています。

例えば、ポケモンが怒った時は、全身が赤みを帯びますが、　これはカラーマップに対して赤いシェーディングをかけています。

これも単純に赤を足すのではなく、カラーマップのＲＧＢ各成分を足し合わせ平均値をとり、黒のグレースケールイメージをコンバイナ内で生成した上で、任意の色と掛け合わせています。 この仕組みにより、バトルエフェクト表現の幅が広がりました。

またこの仕組みの為に、実は影カラーテクスチャは、データ出力時に通常カラーテクスチャからの差分カラーテクスチャに変換しています。

そしてフラグメント処理（コンバイナ処理）で双方を足し合わせ、　影カラーを復元しています。　こうすることで、影カラーテクスチャに対するエフェクトレンダリングを省略しています。

また、レンダリングするためにビデオメモリに乗せる必要もなくなります。

■モデル製作環境・運用関連■

・maya上で３ＤＳのフラグメント処理を再現するためにシェーダープラグインの開発

上記作業を円滑に行う為、　Maya環境にてコンバイナの設定と、そのシェーディングが確認できるカスタムシェーダープラグインを作成しました。

　デザイナは、Maya上にて、シェーディングの設定と確認ができるようになっています。

　またその設定はプリセットとして外部ファイルに保存でき、

　第三者が、そのプリセットを使ってシェーディングの設定を行うことができるようにもしています。

・データベースを用いた数百体のポケモンモデルの管理・運用

数百体あるポケモンのモデルの作成進行や管理を行うにあたって作業を円滑に進めるために

社内独自のデータベースを作成しています。