GUIなんてない方がいい。

どんなに優れたGUIがあるより利用しないで仕事が済むほうがいい。
単機能なCUIツールで実現できないか検討します。

GUIツール内にスクリプトシステムを用意するのが有効な場合もある。
.NETアプリケーションであれば、IronRuby(*2)、IronPython(*3)、
C++であれば、ゲーム開発者におなじみのLua(*5)などを組み込むことによって、
容易にスクリプトシステムを実装することができます。

CUIなどは外部から扱いやすく、
自動化の仕組み、ビルドシステムにも組み込みやすい側面も持っています。

視覚にうったえかけろ!

エクセルシートのように注目させたい項目に
色をつけます。視認性がぐんと上がります。
「ポリゴン数が0のノード」
「マテリアル数オーバー」
「名前は半角英数字のみ」
など、ゲームの仕様に合わせたルールを定義し、
不合格であれば色を付けて知らせます。

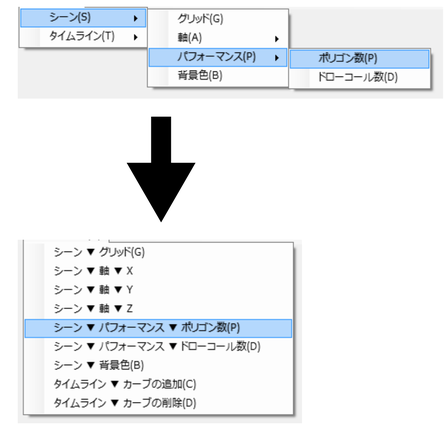
ツールの設計なんて簡単!?
それ、大きな誤解です。

優秀なゲームプログラマがツール設計もできるとは限らない。
プラグインシステムを用意できないか検討します。
ただし、プラグインはなかなか書いてもらえないかもしれません。
軽量な機能実装の手段として、
前述のスクリプトシステムが利用できないか検討します。

機能をつけたのに使ってくれない。
それにはこんな理由が。

存在が知られていなければ使ってもらえないわけです。
ポップアップメニュー機能呼び出しは注意が必要です。
機能の存在を知らせる手段として、
マウスオーバー時に説明を表示させます。

重要な機能がメニューの深いところにありませんか?
深いと目にする機会が減り、存在感が減ります。
ということで、深さを圧縮します。



成功するツール設計

ゲーム開発には必要不可欠なツール。
様々な開発現場を見て、聞いて、作って、学んだ、
成功するツール設計のポイントを紹介します。
ゲーム開発会社でインハウスのツールを設計開発に携わるプログラマ、
またはツール開発に関心のある開発者に向けの内容になっています。

シリコンスタジオ株式会社
伊藤 義弘 / yh-ito@siliconstudio.co.jp

.NETで解決!

各ゲーム機メーカーから提供されている開発環境を考えると、
Windowsのみで動けば問題ないでしょう。過去の資産が使え生産性の高い、
「C#でWindows Forms(と最小限のC++/CLI)」で作ります。

Q1. WinFormsでOpenGLやDirectXで3Dビューポートは実装できるのか?
問題なくできます。C++/CLIでラッパーアセンブリを用意し、
C#側から参照することで実装できます。

Q2. C#をつかわずC++/CLIだけで作ればいいのではないか?
C#と制限してC++/CLIを使うのが答えでしょう。
C++/CLIの資料は非常にすくなく、非常に複雑なC++の仕様に
さらにマネージドのための独自拡張されているため、学習コストが高いです。

いつまでも初心者じゃない
ユーザーのスキルについて。

ユーザーははじめは初級者ですが、中級者・上級者に成長します(*1)。
一般ユーザを対象としたツールとは大きく違うところです。

初心者は、一度に多くの情報を処理することができないが、上級者になるに連れて、
一度に処理することができるようになり、効率よく仕事をこなすようになります。

上級者の作業効率と初級者の学習効率を考え、自由に配置できる
ドッキングウィンドウシステムを用意し熟練度にあったレイアウトを用意します。
モードごとにコントロールを隠すのも一つの解決策です。

英語化ってやっぱり必要?
多言語対応について考える。

今現在日本人のみで開発しても、いつ海外の開発者と仕事を共にするかわかりません。
多言語に対応する仕組みをはじめに考えておくのが正しい選択でしょう。
最初は日本語で作り、主要な部分から英語にするのでも遅くありません。
例えば.NETのWinFormsには多言語に対応する仕組みが用意されています。
VisualStudioを用いると簡単に多言語対応のツールを作成することができます。(*5)

広げた風呂敷は
適度にたたみましょうか。

なぜツールを用意するかといえば、ゲームを効率よく(=安く)開発するためです。
にもかかわらず、気づいてみれば結構なコストがかかっていないか?
規模が大きくなってしまい小回りが効かなくなっていないか?
何でもかんでもできる統合ツールより単機能のツール群の組み合わせで安く作れないか?

プログラマ的に美しく理想的なものが正しいとは限りません。
あまり野心的になりすぎないように。着地地点を再設定する機会を設けてみましょう。

参考

1. Alan Cooper,Robert Reimann,David Cronin共著
『About Face 3 インタラクションデザインの極意』長尾高弘訳 アスキー・メディアワークス 2008年 p.63
2.『IronRuby』
http://ironruby.net/
3.『IronPython』
http://www.codeplex.com/wikipage?ProjectName=IronPython
4.『The Programming Language Lua』
http://www.lua.org/
5.『Windowsフォームを多言語対応にするには?』
http://www.atmarkit.co.jp/fdotnet/dotnettips/314winmultilang/winmultilang.html