

1. Git 環境を準備する

■ Git 関連インストーラーを入手する

3Di Viewer Rei のソースコードをダウンロードするには、まず、Git というソース管理環境を準備しなければならない。ここでは、Windows(XP)における Git 環境について説明する。

Git は、もともと Linux 用に開発されたものであるため、それを Windows でも利用できるように開発された `msysgit` というものが便利である。但し、Subversion との連携や、Web サーバ機能等ではできないが、ソースを取得する上では必要性はほとんどないため、問題ない。

`msysgit` のダウンロードサイト :

<http://code.google.com/p/msysgit/downloads/list>

ダウンロードファイル :

Git-1.6.4-preview20090730.exe

Filename	Summary + Labels
Git-1.6.4-preview20090730.exe	Full installer if you want to use official Git 1.6.4 Featured Beta
PortableGit-1.6.4-preview20090729.7z	Portable application if you want to use official Git 1.6.4 Fea
msysGit-netinstall-1.6.4-preview20090730.exe	Net installer if you want to hack on Git Beta
msysGit-fullinstall-1.6.4-preview20090729.exe	Full installer (self-contained) if you want to hack on Git Beta
msysgitlogo.png	MSysGit logo
gitlogo.svg	Git Logo as SVG

©2009 Google - [Code Home](#) - [Terms of Service](#) - [Privacy Policy](#) -
Hosted by [Google code](#)

次に、TortoiseGit というものをダウンロードする。先の `msysgit` は、Linux 用につくられているため、その操作はコマンドラインで行わなければならない。これを Windows のエクスプローラーに統合した GUI で簡単に操作できるようにするものが TortoiseGit である。現在 (2009 年 9 月時点) では、バージョンは、1.0.2 になっており、実用レベルで使用できるものとなっている。また、日本語プロジェクトもあるので、これを利用することにする。

日本語版 TortoiseGit サイト :

<http://sourceforge.jp/projects/tortoisegit/>

ダウンロードファイル :

TortoiseGit-1.0.2.0-32bit.msi

tortoisegit_1.0.2.0_lang_jp_r1.zip

パッケージ	バージョン	リリース日
TortoiseGit (英語版、インストーラ)	1.0.2.0-32bit	2008-09-23 16:26
TortoiseGit 日本語言語ファイル	1.0.2.0-1	2008-09-23 16:38
msysgit インストーラ	Git-1.6.3.2-preview20090608	2009-06-17 17:06

ニュース フォーラム チケット

最新プロジェクトニュース

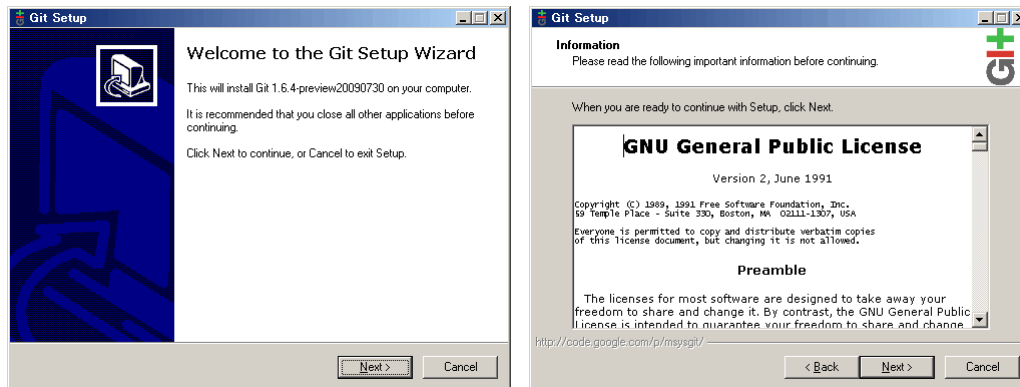
TortoiseGit 日本語言語ファイル (1.0.2.0向け) をリリースしました
hy/om

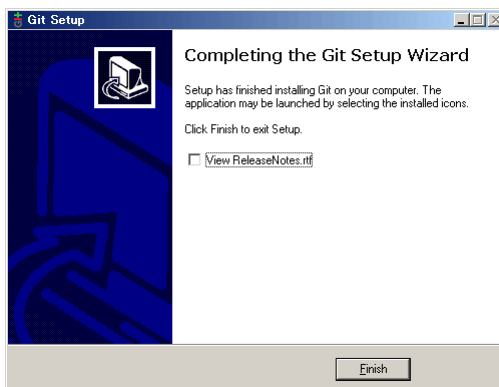
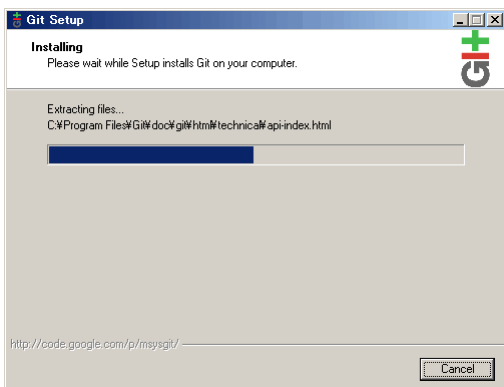
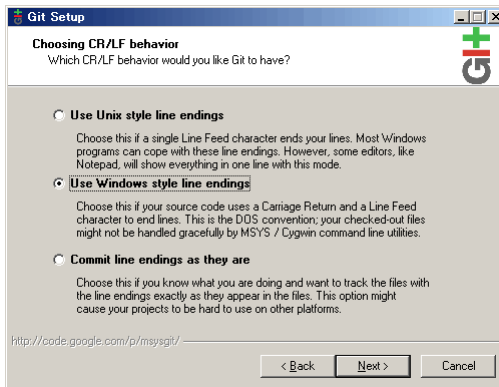
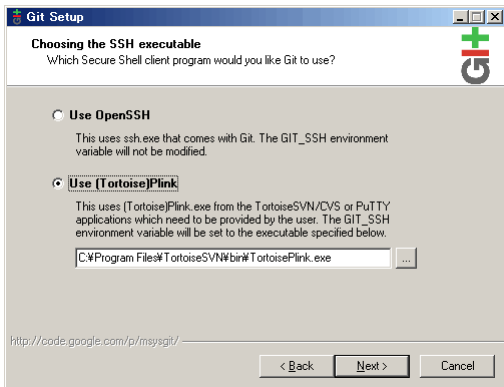
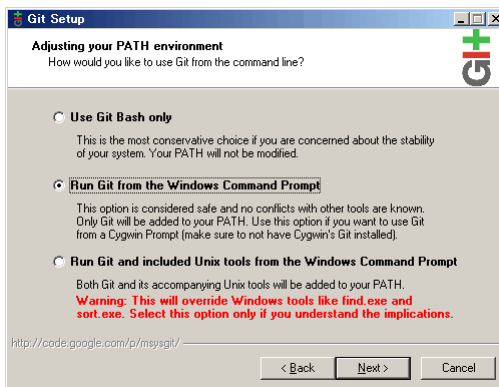
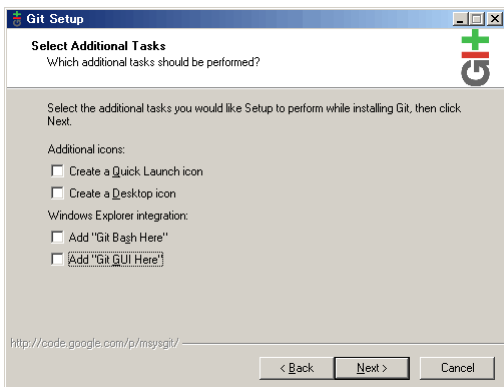
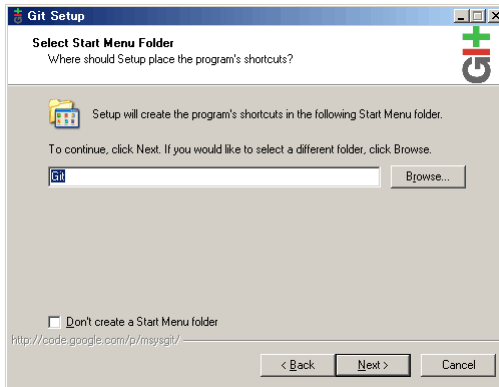
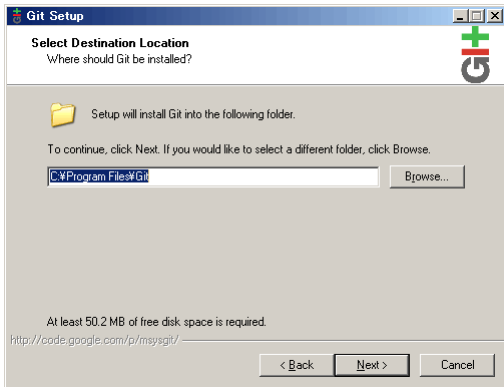
TortoiseGit 日本語言語ファイル (0.9.1.0向け) をリリースしました
hy/om

※TortoiseGit 公式サイト :

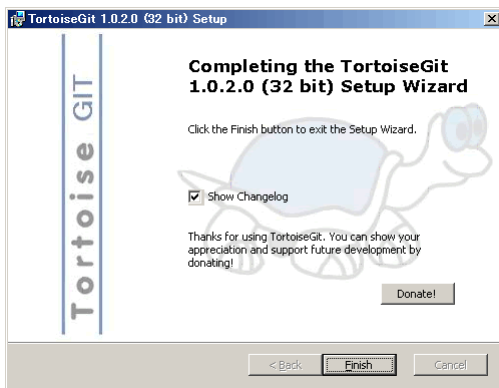
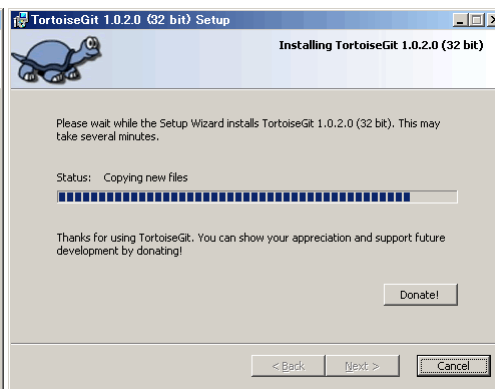
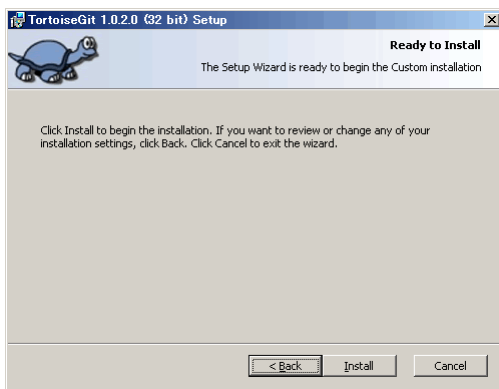
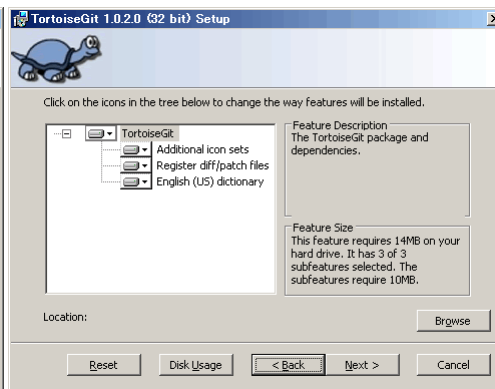
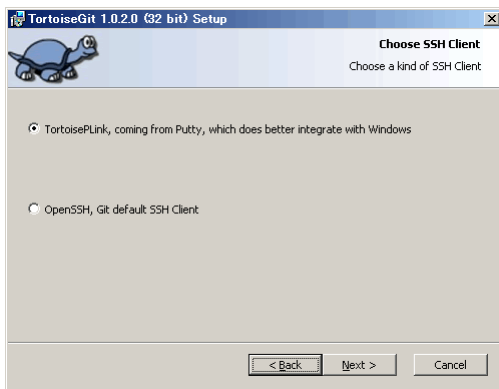
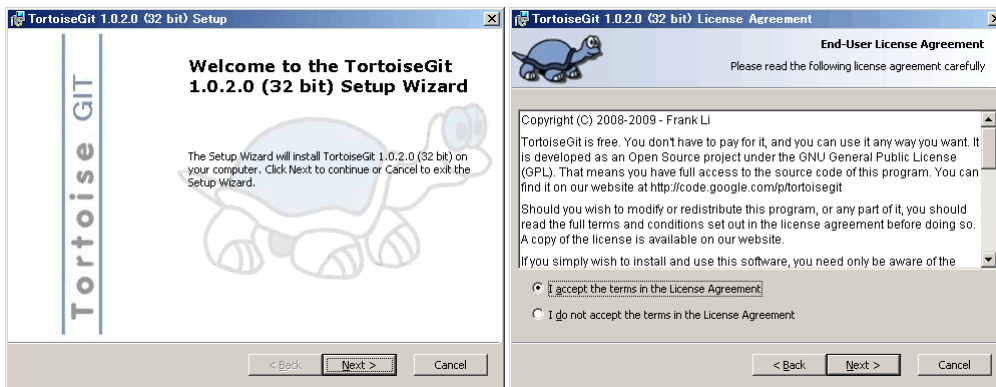
<http://code.google.com/p/tortoisegit/downloads/list>

■ msysgit をインストール





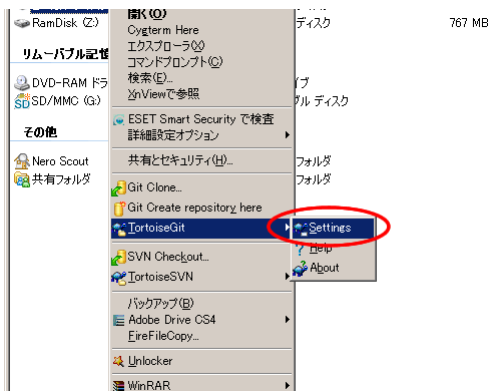
■ TortoiseGit をインストール



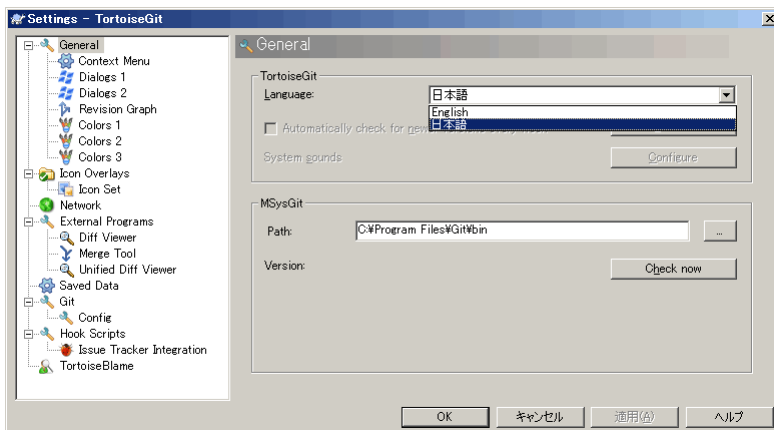
■ TortoiseGit 日本語化

tortoisegit_1.0.2.0_lang_jp_r1.zip を解凍すると、TortoiseMerge1041.dll および TortoiseProc1041.dll という2つのファイルが見つかります。これを TortoiseGit がインストールされたディレクトリ内にあるの「Languages」というフォルダの中にコピーします。

続いて Windows の適当なフォルダを右クリックするとメニューに TortoiseGit というのが追加されているのがわかります。これを選択してでくるサブメニューの「Settings」を開きます。



「General」項目の「Language」で「日本語」を選択し、「OK」をクリックすると完了です。

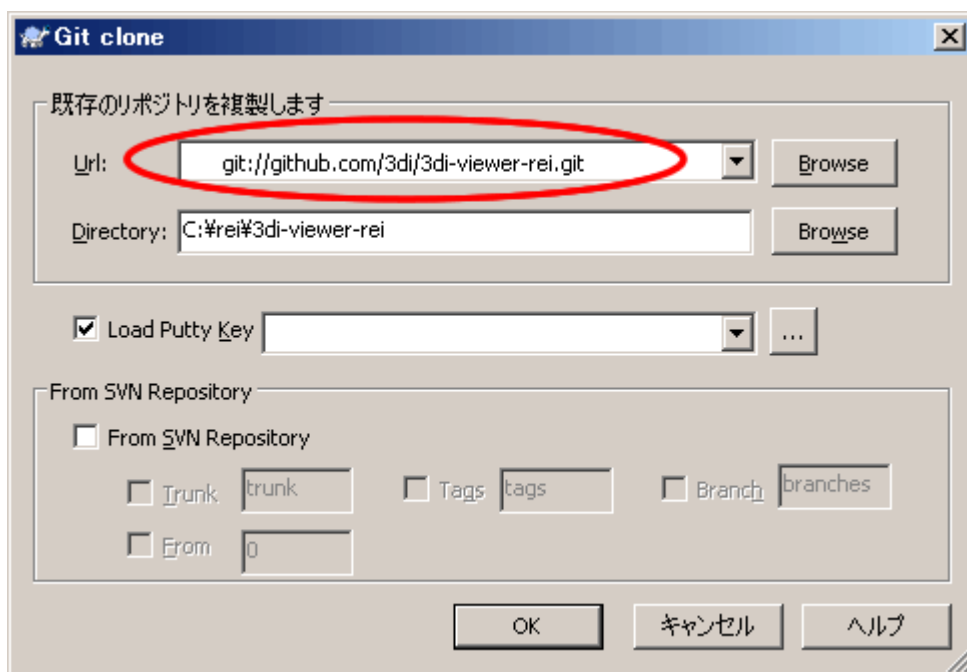


2. Rei のソースコードをダウンロードする

まず、ソースコードを格納するためのフォルダを用意します。基本的にどこに作成しても

構いませんが、念のため、Cドライブのルートに **Rei** というフォルダ(フォルダ名は何をつけても構いません)を作成します。

このフォルダを右クリックし、**Git 複製(clone)**を選びます。



url には、**Rei** のソースコードが管理されているサーバーを指定します。なお、**Directory** の項目は、ダウンロードする **Git** にあわせたフォルダ名に自動的に変更されるようになっています。そして **OK** ボタンでダウンロードが開始されます。

以上の方法により、以下の4つ **GitURL** からのソースプログラムを入手してください。

`git://github.com/3di/3di-viewer-rei.git`

`git://github.com/3di/3di-viewer-rei-libs.git`

`git://github.com/3di/3di-viewer-rei-api.git`

`git://github.com/3di/3di-viewer-rei-firefox.git`

2. ビルド環境の準備

■ Visual Studio 2008 Express Edition のインストール

Rei は、基本的に .NET Framework で開発されている。そのため、Microsoft のサイトから無償で提供されている Visual Studio 2008 Express Edition を利用することができる。ちなみに、最近では、OpenSim 開発者は、C#3.0 仕様でコーディングをしている場合も多くなってきており、例えば、

```
public int Value { get; set; }
```

のような自動実装プロパティによるコーディングがあると、Visual Studio 2005 ではビルドに失敗する。なので、可能なかぎり 2008 以降を使用することをお勧めする。

ダウンロードサイト：

<http://www.microsoft.com/japan/msdn/vstudio/Express/>

※iso 版 (すべてのツールが収まった DVD イメージ)

<http://www.microsoft.com/japan/msdn/vstudio/2008/product/express/offline.aspx>

■ DirectX の準備

さて、最後に、もう 1 つだけ必要な準備があります。DirectX ランタイムの確認です。Rei は、Microsoft DirectX 9c を用いて開発されていますので、このランタイムライブラリがインストールされていない場合は、インストールする必要があります。ゲーム等でもよく使われているものですので、既にインストールされている場合も多いと思いますが、されていない場合は、以下よりインストーラーをダウンロードしてインストールを行ってください。

<http://www.microsoft.com/downloads/details.aspx?displaylang=ja&FamilyID=886acb56-c91a-4a8e-8bb8-9f20f1244a8e>

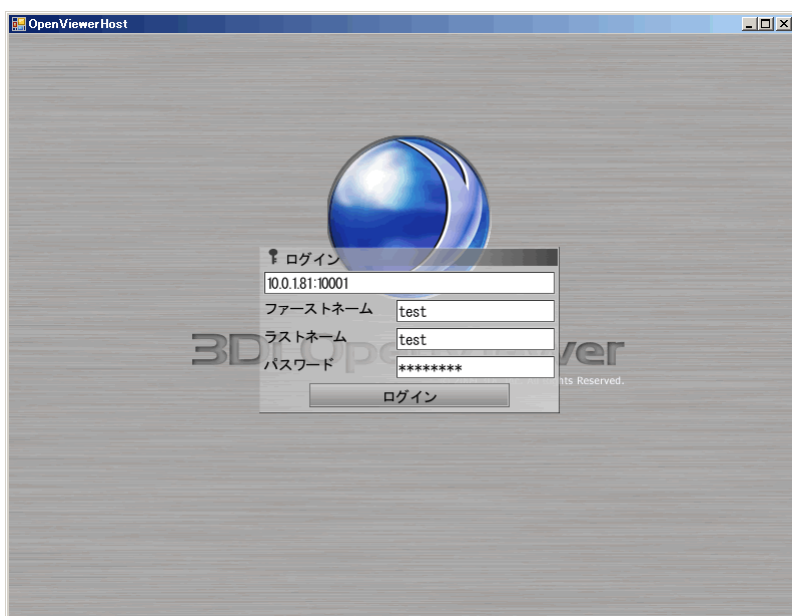
■ ビルドの実行

Git で作成された 3di-viewer-rei を見ると、runprebuild2008.bat というファイルが見つかります。これをダブルクリックし実行すると、Visual Studio 2008 用のソリューションファイル 3DiOpenViewer.sln が生成されます。これをダブルクリックすると Visual Studio が

起動しますので、**C#**を選択し、ビルドを実行します。うまくできていれば、正常終了となるはずです。ちなみに、ビルドにはもう一つ方法があり、ソリューションファイル生成の際に **compile.bat** というバッチファイルが合わせて生成されています。これを実行した場合は、**Visual Studio** の起動なしで、自動ビルドすることができます。

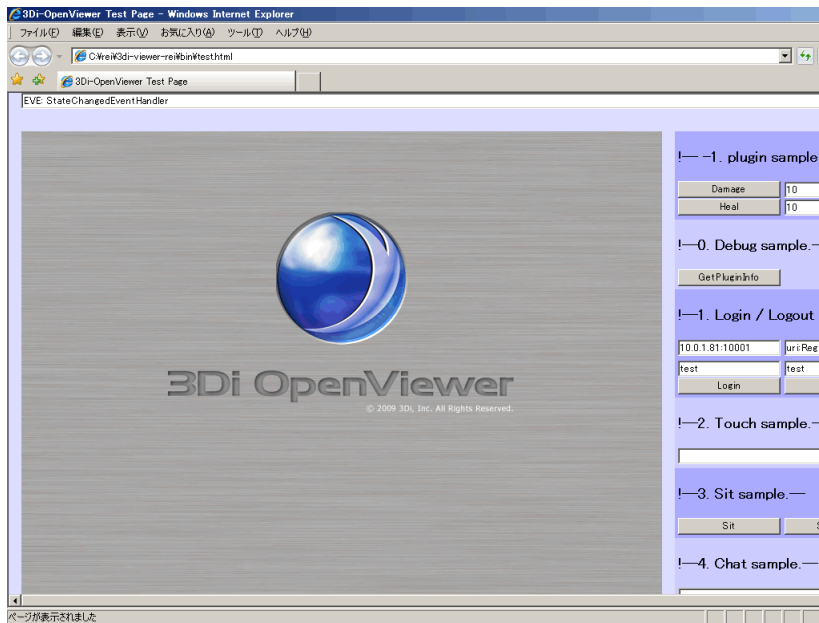
では、ビルドでできた **Rei** を起動させてみましょう。最も簡単な方法は、アプリケーション版 **EXE** を実行する方法です。

bin フォルダを見ると、**OpenViewerHost.exe** というものが見つかるはずですので、これをダブルクリックすると **Rei** を起動することができます。



先ほどは、アプリケーションとして起動しましたが、**Rei** の醍醐味はなんといっても、**HTML** に埋め込んで使用することですので、今度は、その方法をご説明します。アプリケーション版 **Viewer** と同じ **bin** フォルダを見ると、**test.html** というファイルがありますので、これを **Internet Explorer** (※バージョン 8 は非対応) で開いてください。ActiveX を実行してよいかどうかの警告バーがでますので、クリックし「ブロックされているコンテンツを許可」を選択してください。

下記のように表示されていれば成功です。



url には、OpenSim サーバーの IP アドレス、ポート番号 (9000 や 8002 など) を入力し、そのサーバーに登録したアカウントを入力すればログインできるはずです。

Rei は 3Di 社のメッシュアバター方式を採用しているため、セカンドライフのアバターとは異なるオリジナルアバターが表示されていますが、その他のプリムはほぼ問題なく表示できます。(セカンドライフのプリムは表示可能ですが、3Di 社のメッシュプリムと比較するとパフォーマンスがあまりよくありません) ちなみに、3Di 社のメッシュアバター方式は、セカンドライフとは大きく概念が異なり、どちらかという、オンラインゲーム等に近い方式を使っています。3DCG ソフトでモデリングされたアバターを自由にインポートできるようになっていますので、やろうと思えば、見た目上はセカンドライフのようなアバター表現が可能です。また、セカンドライフでは、タイニーアバターのようにボーン変形と複雑なプリムの組み合わせで実現しなければならなかった特殊なアバター形状も、メッシュアバター方式の場合は自然な制作方法で実現することが可能です。(ボーン構造も自由)

右側にたくさん、HTML フォームが並べられているのが見えると思いますが、これらは、Rei が持つ様々な API のサンプルコードです。API の詳しい使い方については、3Di 社ホームページ 3Di OpenViewer のページに掲載されている「Web 制作マニュアル」(PDF)を ご覧になるとよいでしょう。

<http://3di-opensim.com/openviewer/> (本ページ中段付近にあります)

いかがでしたでしょうか。Rei は、BSD という非常に自由度の高いライセンスで提供され

たオープンソースプロジェクトです。3Di 社の **OpenViewer** は、3Di **OpenSim** に最適化したものですが、今後は、それ以外のプラットフォームにも自由に接続可能な共通ビューアとして発展してゆくことでしょう。この結果として、特殊な技術に思われがちな 3D ではなく、誰もが日常的に使う 3D インターネットの時代へと変化することを願っています。

■ Irrlicht ライブラリのビルド

3di-viewer-rei-libs¥Irrlicht_r1883¥Irrlicht SDK¥source¥Irrlicht 内にある

Irrlicht9.0.sln を Release ビルド

3di-viewer-rei-libs¥Irrlicht_r1883¥bin¥Release に Irrlicht.dll が生成されます。

3di-viewer-rei-libs¥Irrlicht_r1883¥IrrlichtW 内にある

IrrlichtW.sln を Release ビルド

3di-viewer-rei-libs¥Irrlicht_r1883¥bin¥Release に IrrlichtW.dll が生成されます。

3di-viewer-rei-libs¥Irrlicht_r1883¥Irrlicht.NET 内にある

runprebuild2008.bat を実行すると Irrlicht.NET.sln が生成されます。

Irrlicht.NET.sln を VisualC#2008 で開きます。Visual C# 2008 Express は、インストール直後では、ビルドメニューに「構成マネージャ」というメニューが表示されません。その場合、ツール>オプションで、すべての設定を表示にチェックを入れ、プロジェクトおよびソリューション>全般の中の「ビルド構成の詳細を表示」にチェックを入れてください。

これで、ビルドメニューにある「構成マネージャ」で、ソリューション構成を **Release** に変更し、ビルドを行います。(警告が表示される場合がありますが無視して構いません)

Irrlicht.NET.dll と Irrlicht.Extensions.dll が bin ディレクトリに生成されるはずですが。

■ OpenMetaverse ライブラリのビルド

3di-viewer-rei-libs¥libomv_r2418¥trunk 内にある

runprebuild2008.bat を実行すると OpenMetaverse.sln が生成されます。

「ソリューションフォルダはこのアプリケーションではサポートされていません」というメッセージがでますが無視して構いません。そのまま進めてください。

ソリューション構成を **Release** に変更し、ビルドを行います。(警告が表示される場合がありますが無視して構いません)

3di-viewer-rei-libs¥libomv_r2418¥trunk¥bin の中に下記 5 つの dll が生成されます。

OpenMetaverse.dll

■ JavaScript API のビルド

JavaScript API のビルドでは、Rei ビューアのビルドで生成される OpenViewer.dll and

Interfaces.dll の 2 ファイルがないとビルドできません。まず先に、Rei をビルドした後、これら 2 つの dll を 3di-viewer-rei-api の bin ディレクトリにコピーしてください。

さらに irrlicht と openmetaverse のビルドで生成したファイルもコピーします。3di-viewer-rei-libs¥irrlicht_r1883¥bin¥Release に生成されたすべての dll と 3di-viewer-rei-libs¥libomv_r2418¥trunk¥bin に生成された OpenMetaverse.dll を 3di-viewer-rei-api¥bin の中にコピーしてください。

ここまで準備ができたなら、3di-viewer-rei-api 内の runprebuild2008.bat を実行し Adapter.sln を生成します。これを Release モードでビルドします。(なお警告が表示されますが無視して構いません)

3di-viewer-rei-api¥bin の中に Adapter.dll が生成されます。

以上すべてのライブラリのビルドが完了したら最後に 3di-viewer-rei¥bin の中に 3di-viewer-rei-libs¥irrlicht_r1883¥bin¥Release に生成されたすべての dll 3di-viewer-rei-libs¥libomv_r2418¥trunk¥bin に生成された OpenMetaverse.dll 3di-viewer-rei-api¥bin に生成された Adapter.dll をコピーして使用してください。

■ Firefox 用 plugin のビルド

Firefox のプラグインをビルドするには、まず、InternetExplorer 用のビルドを事前に行う必要があります。

Gecko SDK をダウンロードする

https://developer.mozilla.org/en/Gecko_SDK#Downloading

Firefox3.0 用の xulrunner-1.9.0.14.en-US.win32.sdk.zip をダウンロードします。

Zip ファイル内の xulrunner-sdk フォルダが 3di-viewer-rei-firefox と同一階層になるよう解凍をします。

3di-viewer-rei¥bin の中から以下の 6 つの dll ファイルを 3di-viewer-rei-firefox¥bin にコピーしてください。

OpenViewer.dll
Interfaces.dll
Adapter.dll

OpenMetaverse.dll
OpenMetaverseTypes.dll
Nini.dll

VisualStudio C# 2008 Express を起動し、メニューのファイル>プロジェクトを開くから 3di-viewer-rei-firefox¥3DiOpenViewer¥bootstrap の中の OpenViewer.bootstrap.csproj を開きます。ソリューションエクスプローラーの OpenViewer.bootstrap を右クリックし、ビルドを選択します。「名前をつけてファイルを保存」というダイアログが表示され、OpenViewer.bootstrap.sln という名前で保存するかどうか聞かれますので、そのまま保存してください。警告は表示されますが、ビルドが正常に終了していれば問題ありません。

次に VisualStudio C# 2008 Express を起動し、メニューのファイル>プロジェクトを開くから 3di-viewer-rei-firefox¥npOpenViewer の中の npOpenViewer.vcproj を開きます。このとき、「このバージョンのアプリケーションではサポートされないプロジェクトの種類です」というエラーが2度表示されますが、無視して進めてください。ソリューションエクスプローラーの npOpenViewer を右クリックし、ビルドをします。警告は表示されますが、ビルドが正常に終了していれば問題ありません。

■インストール

Firefox 版 Rei をインストールする場合、3Di 社製の OpenViewer はアンインストールしておく必要があります。

3di-viewer-rei-firefox¥bin 中にある次の2ファイルを Firefox がインストールされているディレクトリ（通常は、C:¥Program Files¥Mozilla Firefox¥plugins）にコピーします。

npOpenViewer.dll
npOpenViewer.xpt

次に、3di-viewer-rei-firefox¥bin 中にある bootstrap.dll を Firefox がインストールされているディレクトリと同じ場所（通常は、C:¥Program Files¥Mozilla Firefox）にコピーします。

3di-viewer-rei-firefox¥bin 中にある test.html を Firefox で開いて、ビューアの初期画面が表示されていれば成功です。

■オリジナルプラグインの開発

Rei は、プラグイン機能を搭載しており、オリジナルの機能を自由に追加可能なしくみになっています。サンプルのプラグインとして、**TestPlugin** というものが以下のフォルダにありますので、調べてみてください。

3di-viewer-rei¥3DiOpenViewer¥Plugins

このサンプルプラグインをビルドしてできる **TestPlugin.dll** を **Rei** と同一ディレクトリに置くと機能させることができます。**test.js** の中に、**DoDamage()**と **DoHeal()**という関数が定義されていますが、これを呼ぶことで、アバターの頭上にあるグラフが減ったり増えたりするというものです。