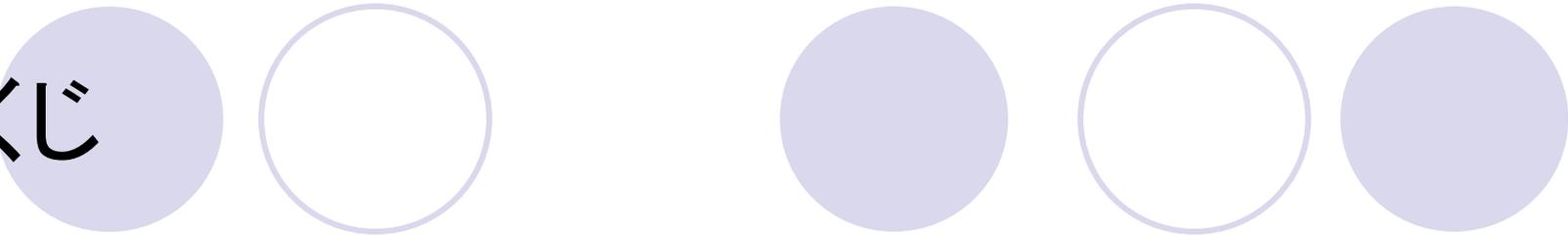


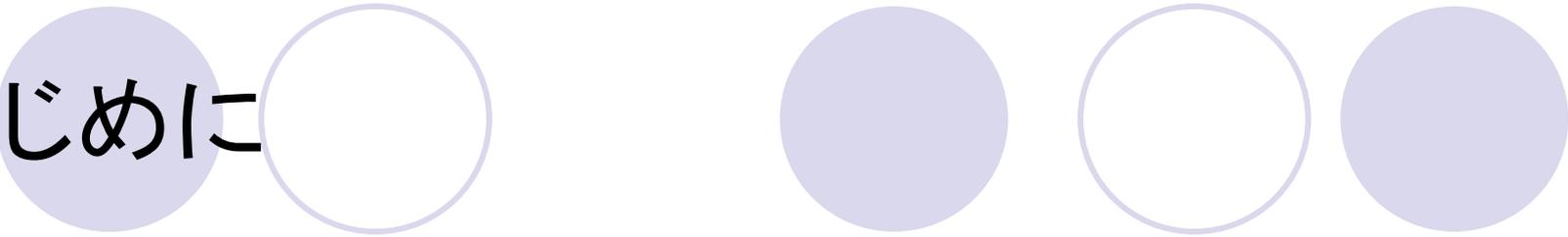
いもうとデスクトップが出来るまで

話すひと: 646



# もくじ

- 大雑把に分けて 3 項目について話します
  - いもうとデスクトップと何かとの違い。
  - いもうとデスクトップの構造、仕様。
  - いもうとデスクトップのこれから。
- 時間と準備が整えば……
  - リモート操作、音声認識のデモ(予定)



はじめに

- **伺か(何か)ではありません**
  - 仕様もコンセプトも何もかもが違う。
  - 一部参考にした部分は存在するが、あくまでも参考程度。
  - 目指していく気もありません。
  - あくまでもいもうとデスクトップなので。

# いもうとデスクトップとは？

- 2006年の「窓の社」で公開されたのが元ネタ
  - ツンデレスライダーや年齢調整機能など、細かい設定が話題を呼ぶ。
  - 以後2chのスレで開発が進められるが停滞。
- 今回のいもうとデスクトップは
  - 今年2月後半から開発を開始、4/1に0.12を公開。
  - OpenDAEのソースコードを流用することで、開発負荷を削減。

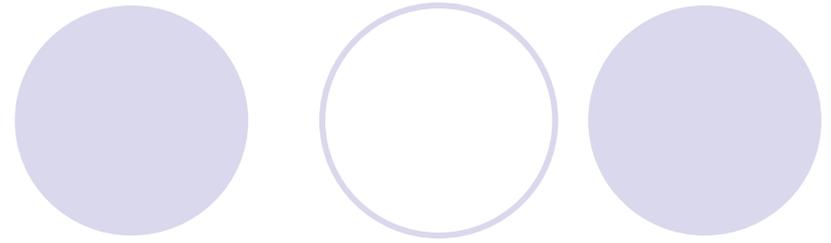
# 実行中の画面表示



# 伺かとの違い、似ている部分

- 外見上の違いはあまりない
  - しかし内部構造、仕様などはまったく違う。
- コンセプトは大きく異なる
  - 偽AIを目指していないわけではない、しかし表現のためのものではない。
  - つまり、あくまでも**いもうと**。
- 一部伺かを参考にした部分あり
  - スクリプト周りなど、後述。

# 伺かとの違い(1)



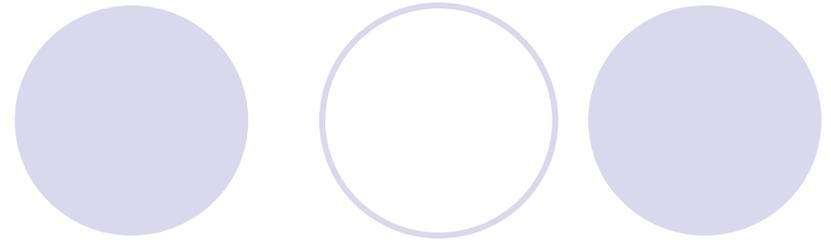
- 重要な違い

- スクリプト実行エンジンを本体が持つ。
  - 別途配布する必要がない。
- サーフェスにアルファ付きPNGを使用する。
  - アルファだけ別ファイルにするほうが手間がかかる。
- 自発トークが存在しない。
  - 基本的に入力されたコマンドに対するレスポンスのみ。
- いもうと、吹き出しはユニークなIDを持つ。
  - 現時点は GUID を使用している。

# 伺かとの違い(2)

- あまり重要じゃない違い
  - 定義ファイルは全て XML。
    - ほとんどの言語で扱えるので、ツールなどが作成しやすくなる。
  - オーナードローメニューはない。
    - メニューの重要性は低いので必要がない。
  - 当たり判定などのイベントなどが存在しない。
    - 必要なかった。
  - サーフェス合成やアニメーションなどはない。
    - サーフェスが1枚しかないので無意味だった。

# 伺かとの違い(3)



- 別にどうでもいい違い

- 開発言語が C#。

- 開発環境の整備が非常に進んでいる。
- 最新の開発環境がフリーで手に入るので、ソースを公開したとしても負担が少ない。

# 伺かと似ている部分(1)

- ほぼ同じ部分

- デスクトップ上に不定形ウィンドウで表示される。
  - MS Agent や Apricot などを見ても分かるように方法はこれぐらいしか存在しない。
- 吹き出し状のウィンドウを利用してテキストなどを出力する。
  - 同じく表示する方法はこれぐらいしかない。

# 伺かと似ている部分(2)

- よく似ている部分

- さくらスクリプトライクな独自スクリプトを実行する。
  - 3割ほどの互換性はある。
- イベントドリブン方式。
  - スクリプト実行エンジンにイベントを投げる。

# いもうとデスクトップの構造(1)

- 主な構成要素

- クライアント

- 狭義のいもうとデスクトップ。

- サーバ

- クライアントからの指示を実行する。

- いもうと

- リアクション辞書やサーフェスを持つ。

- 吹き出し

- サーフェスや矢印画像など。

# いもうとデスクトップの構造(2)

- サーバとの通信

- サーバとはTCP+バイナリシリアライズで接続。
  - 早い話 .NET Remoting。
  - 開発のし易さで選択、特に問題は出ていない。

- デスクトップへの表示

- WPFを使って実現。
  - これもまた開発のし易さで選択。
  - 技術的には何でもやりたい放題。

# 補足：WPFとは？

- .NET 3.0 で導入された、次世代の描画フレームワーク。
  - 今までと違い、表現力の高いアプリを作成可能。
  - ほぼ全ての描画がハードウェア処理。
  - 半透明ウィンドウ時もハードウェア処理。
    - これは WPF 3.5 から。
    - 半透明ウィンドウ自体はVista上でハードウェア処理。
  - 音楽、動画などを非常に簡単に扱うことができる。
  - まだ使用例は少ない……。

# いもうとデスクトップの仕様(1)

- ユニークID

- いもうとや吹き出しに対し、1つ割り当てられる。
  - 名前やディレクトリ名での判別を行わない。
  - デフォルトである「さくら」には「F3EC60A3-C5FB-443a-B05E-C3345AB37269」が割り当てられている。

- サーフェス画像

- アルファチャンネル付きの32ビットPNGを使用。
  - 伺かのようにPNAを用意しなくても、綺麗な半透明表示が可能。

# いもうとデスクトップの仕様(2)

- スクリプト言語

- 美坂ライクな言語を採用。
  - ただしフロー制御や変数システムなど大幅に強化済み。
- 本体が言語エンジンを持つため、別に配布するような必要がない。
  - HDD 使用量の面からでも安心。
  - XML ベースの言語エンジンも開発していたが廃止。
- ロード、アンロードの必要がない。
  - クライアント起動と同時に読み込まれる。
  - いもうと単位でエンジンのインスタンスを作成。

# スクリプト言語のサンプル

- デフォルト「さくら」から抜粋

```
// 辞書 - おとな
```

```
#Common
```

```
{ $if ( { $age } >= 15 ) }
```

```
// 起動時
```

```
$OnBoot_default; { $if ( { $hour } < 6 || { $hour } >= 23 ) }
```

```
{ $if ( { $tsunderelevel } <= 5 ) { { $if ( { $tsunderelevel } > { $random(4) } + 1 )
```

```
{ { $OnBoot_night_normal } } else { { $OnBoot_night_tsun } } } } else { { $if ( { $tsunderelevel } > { $random(4) } + 5 ) { { $OnBoot_night_dere } } else { { $OnBoot_night_normal } } } } }
```

```
$OnBoot_default
```

```
{ $if ( { $tsunderelevel } <= 5 ) { { $if ( { $tsunderelevel } > { $random(4) } + 1 ) { { $OnBoot_normal } } } else { { $OnBoot_tsun } } } } } else { { $if ( { $tsunderelevel } > { $random(4) } + 5 ) { { $OnBoot_dere } } } else { { $OnBoot_normal } } } } }
```

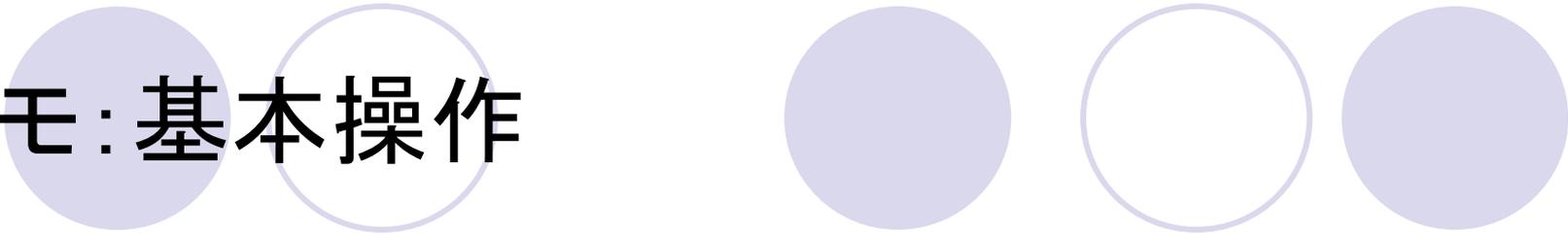
```
$OnBoot_tsun
```

```
なに？¥w5相手してほしいの？¥n¥w5少しだけならいいわよ。  
相変わらずヒマそうね。
```

# いもうとデスクトップのこれから

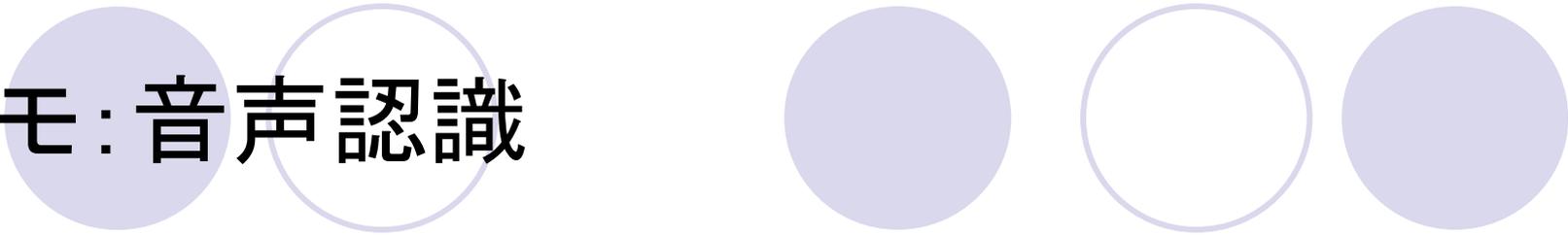
- より元ネタに近く
  - OpenCV を使った画像判別。
    - 現在C#から呼び出し可能なライブラリを開発中。
    - ポジティブサンプルの収集に苦勞している。
    - Intel の中の人ごめんなさい。
- オープンソース化？
  - まだそこまでは考えていない。
- 最終目標
  - 窓の杜に取り上げてもらうこと。

# デモ：基本操作



- 今日と同マシソ内でのデモ
  - 起動、接続。
  - カレントディレクトリ移動、ファイル列挙。
  - DOS コマンド実行。
  - ツンデレ度合い変更。
  - などなど…

# デモ：音声認識



- 特定のコマンドは音声で実行可能
  - Vista の音声認識機能を利用。
  - 実行可能なコマンドを制限することで、音声認識の精度が向上。