【B4】ケーススタディ

EMBARCADERO TECHNOLOGIES®

DEVELOPER CAMP

「Delphiでつくる！3DVRシミュレーション」
～VR-Studio(tm), UC-win/Road, UC-win/Road SDK～

株式会社フォーラムエイト　システム開発グループ主事
宮本　卓也

アジェンダ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- 会社概要
- VR-Studio, UC-win/Roadのご紹介
- UC-win/Road SDK

会社概要

FORUM 8　Company profile

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

株式会社　フォーラムエイト　FORUM8. Co., Ltd

設 立　　：1987年5月

資 本 金　：5,000万円

事業内容：設計支援ソフトウェアの開発／販売／サポート、各種ソリューション提供

社 員 数　：140名（平成21年6月現在、正社員数）

◎ 土木設計支援パソコン用パッケージソフト開発販売で創業　UC-1は、1981年発売開始

◎ 登録ユーザ数12,524（2009.4.17現在）、土木設計ソフトで、トップクラスのシェア

◎ ユーザ内訳：建設コンサル60%、官公庁、大学、建設、自動車、各種研究機関

◎工学博士9名、技術士6名（建設/情報部門）、ソフト開発・基本情報技術者41名

◎ 事業所：国内5・海外5、全国4営業所で営業サポート、海外輸出、大型プロジェクト受注

FORUM 8　Company profile

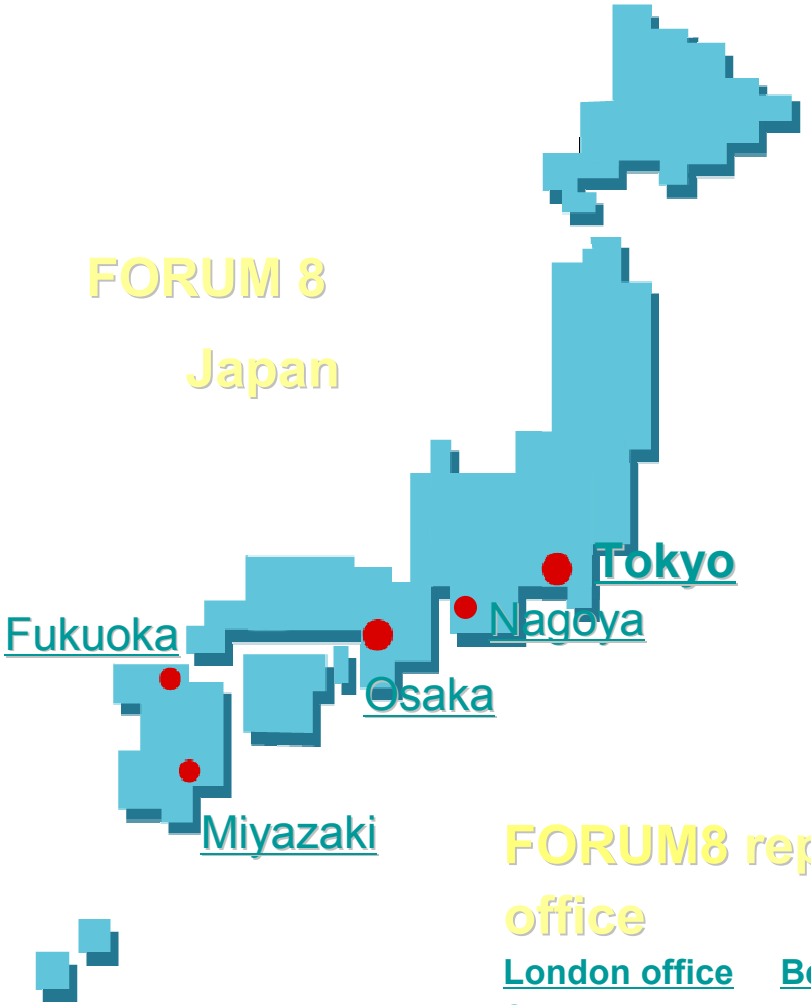
EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

FORUM8(China)

Shanghai China

FORUM 8

Japan



FORUM 8 (NZ)

New Zealand

FORUM8 representative office

London office　Beijing　Thai land New!
Singapore　New Delhi　Sydney
FORUM8 AZ (Arizona) New!

VR-Studio(tm), UC-win/Roadのご紹介

VRへの取り組み

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

クライストチャーチはガーデンシティーとも呼ばれる美しい街です。
そこに住んで見ると、日本では個々の事業に過大な費用をかけながら美しい街を作れないのは何故か、事業が途中で頓挫したり紛糾したりするのは何故かという疑問が強くなりました。
パッケージソフトとして構造解析の開発と販売を中心としてきましたが広義の設計にはその周辺をも含め色々な視点からデザインを確認、協議できるVRのソフトの提供が必要との結論を得ました。win/Roadという製品になって実を結び現在も成長を続けています。（会長 和田忠治）

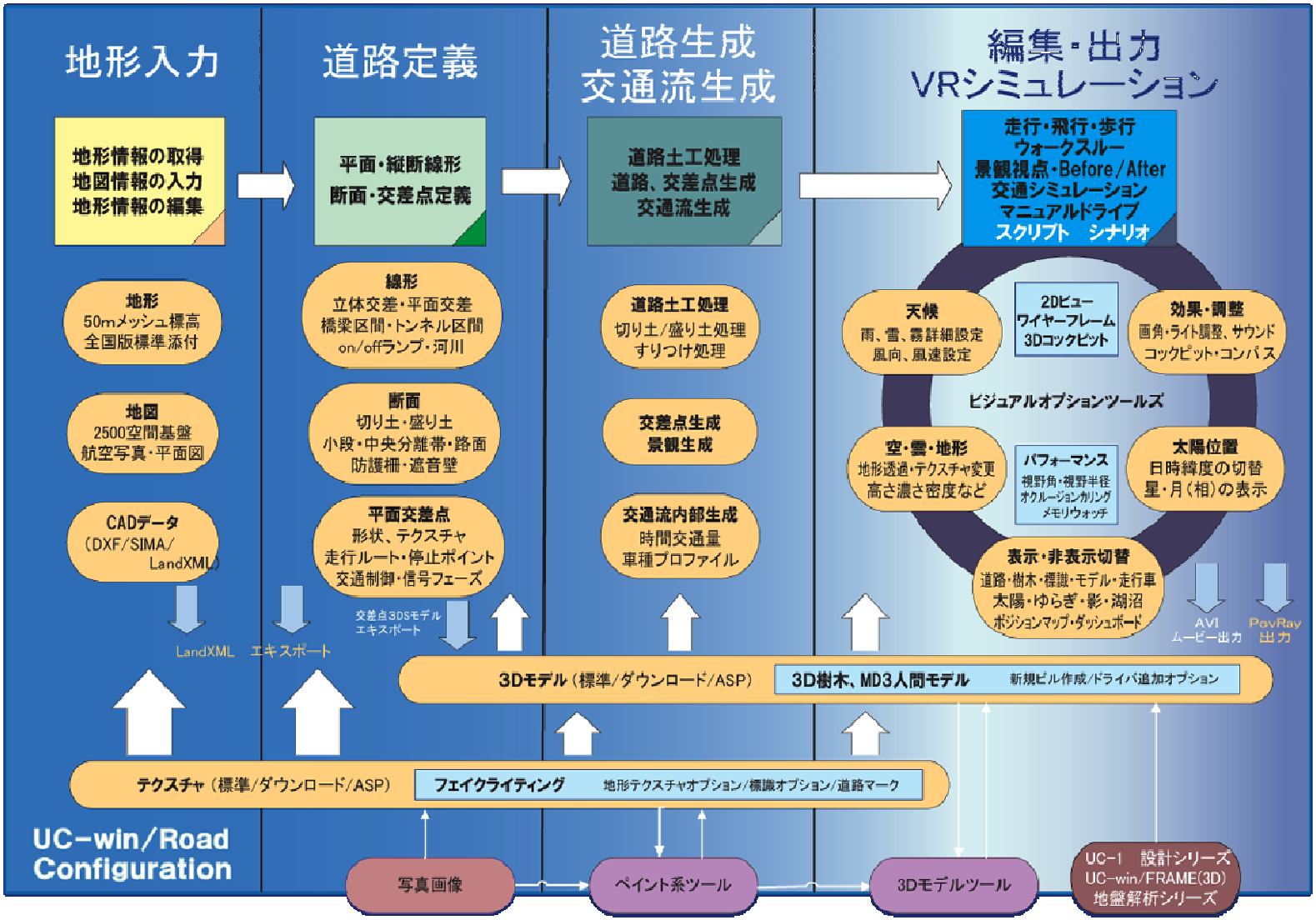
1999年 2月	着想
1999年 4月	開発着手
2000年 5月	初版リリース
2001年	2版、3版、4版
2002年 9月	ソフトウェア・プロダクト・オブ・ザ・イヤー2002受賞
2002年11月	受賞記念第1回 VRコンテスト

GRAND PRIX

★2008年グランプリAVI

UC-win/Road Configuration

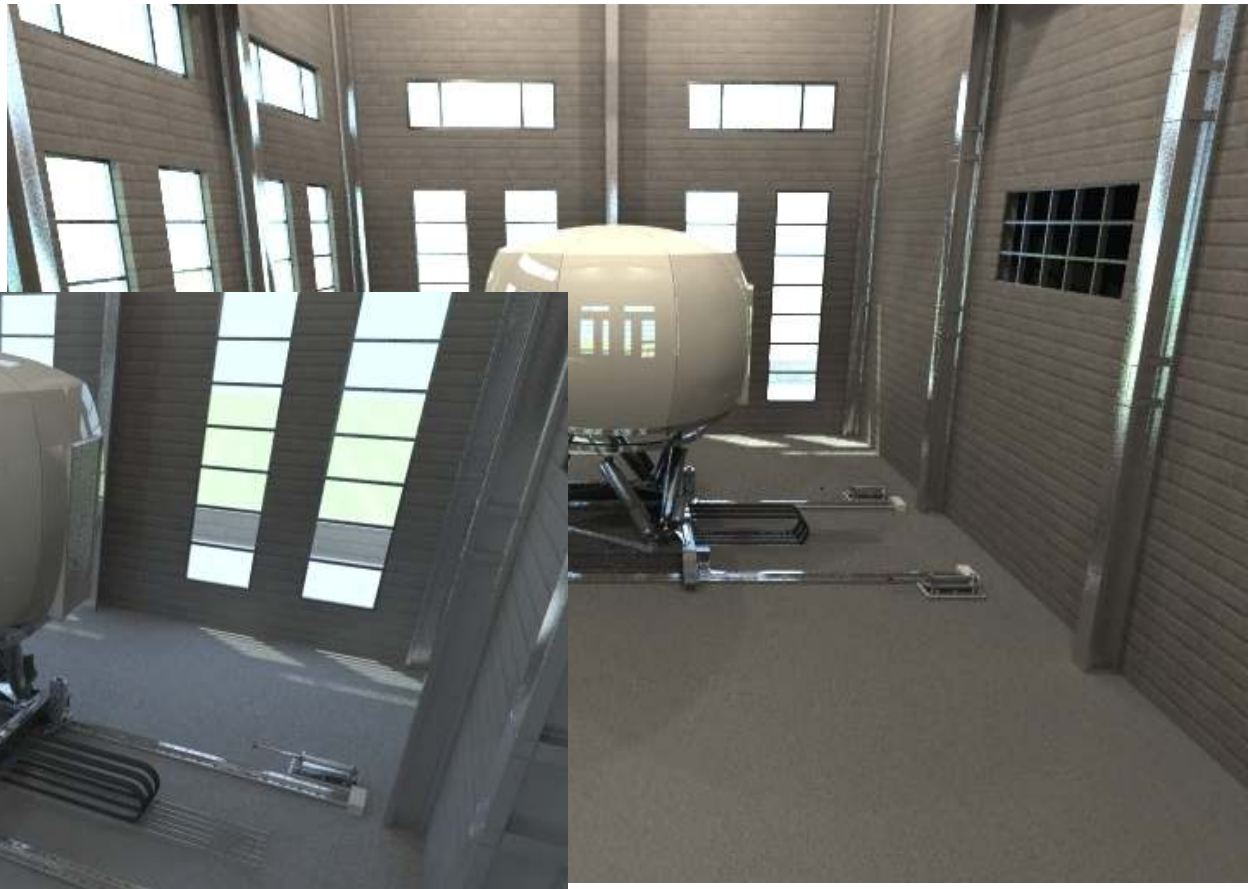
EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

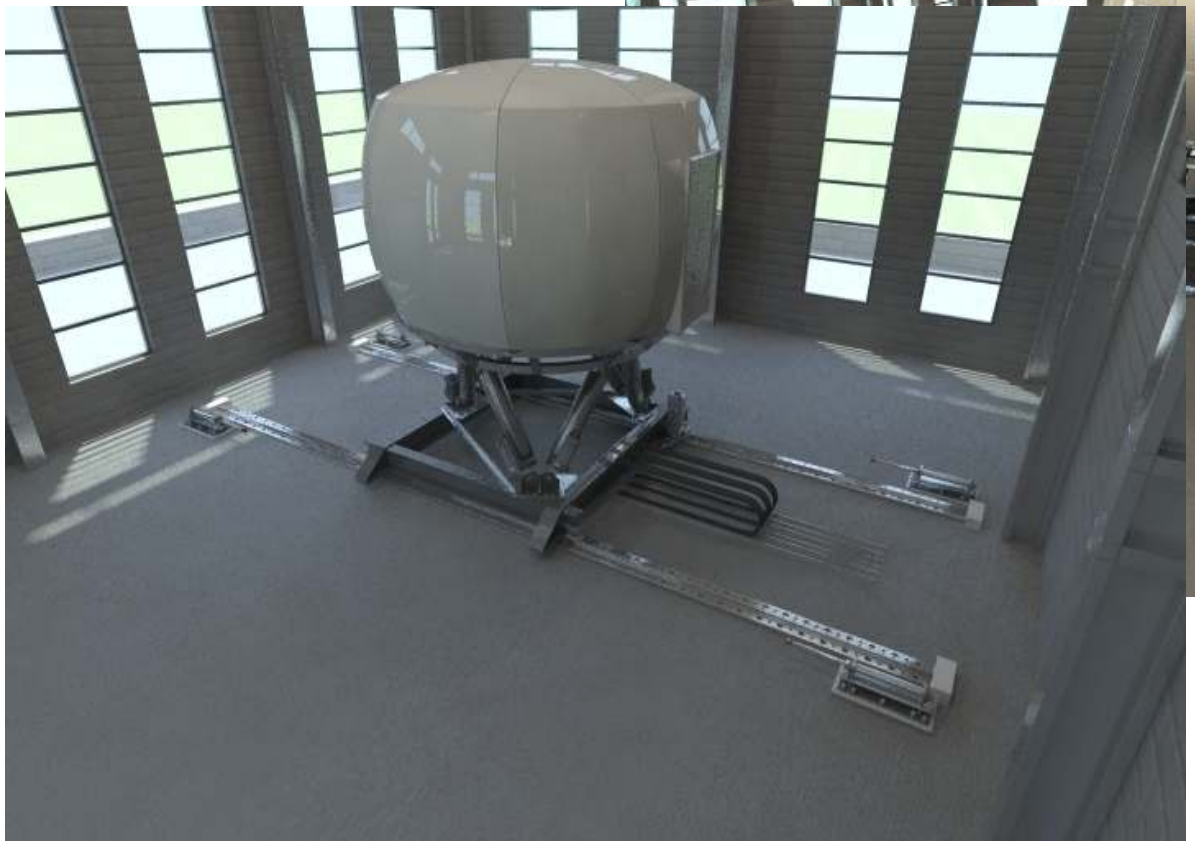


各種シュミレータ

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

モーションイメージAVI





DEMONSTRATION

- さらに詳しい情報は・・・

  http://vr.forum8.co.jp/

3D Stereo View
Multi User
Large Scale
Multiple Realties

Multi Core / CPU
Enhanced Traffic simulation
Advanced Shading Lighting
Multi Modal Editors

UC-win/Road SDK

～3DVRシミュレーションをつくる～

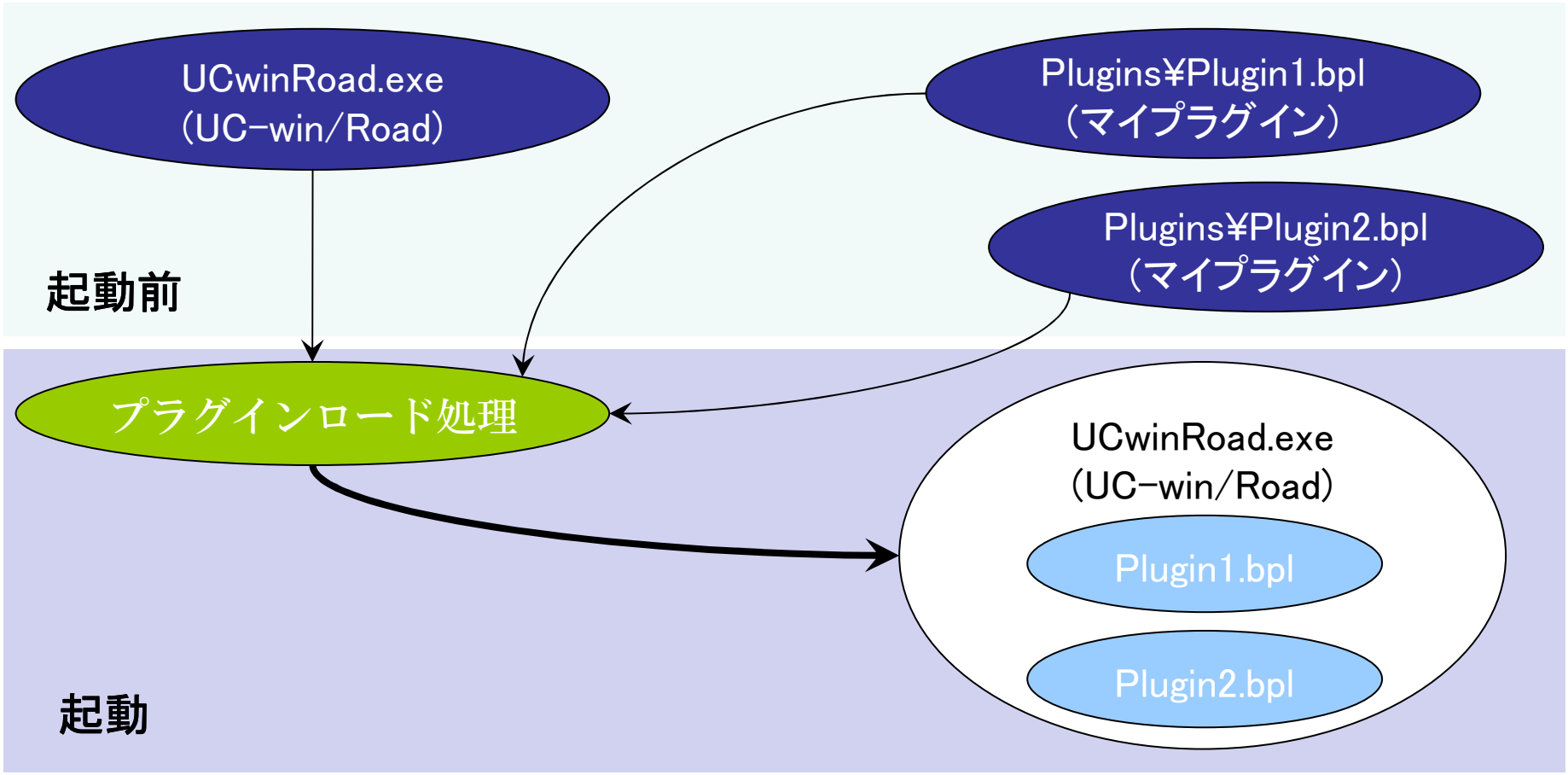
概要

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- UC-win/Road SDKとは
  - UC-win/Roadオプション開発環境の提供
  - UC-win/Roadの拡張、制御
  - DelphiのInterfaceを利用しAPIを実装

- 目標
  - 自由なカスタマイズ
  - UC-win/Roadの活用
  - 開発者からのフィードバック

プログラムとSDKの関係



※　UC-win/Roadプラグインマネージャから手動でロード・アンロードが可能。

UC-win/Road SDK の構成

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- UC-win/Road 3.4 SDKの構成：
  - APIライブラリファイル（Delphi 2007™専用）
  - API添付資料
    - APIの使用説明書（入門説明、プログラマーズガイド）
    - リファレンスガイド（公開クラスの仕様説明書）
    - サンプルプログラム

UC-win/Road API のサポートする内容1

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- 編集機能：静的なデータの読み取り、書き込み、編集
  - 地形、航空写真、道路、交差点、交通、3Dモデル
- GUI機能：ユーザインタフェースのカスタマイズ
  - メイン画面にコントロールの追加、既存コントロールの制御
- インタラクション機能：シミュレーション状況の取得
  - キーボード、マウス、ゲームコントローラ操作によるコールバック関数の呼び出し
  - 視点状況
  - ログ出力

UC-win/Road API のサポートする内容2

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- シミュレーション：VR空間のリアルタイム制御
  - モデルやキャラクタのリアルタイム制御
  - メイン画面の視点制御
  - ドライビングシミュレーション運転開始・制御

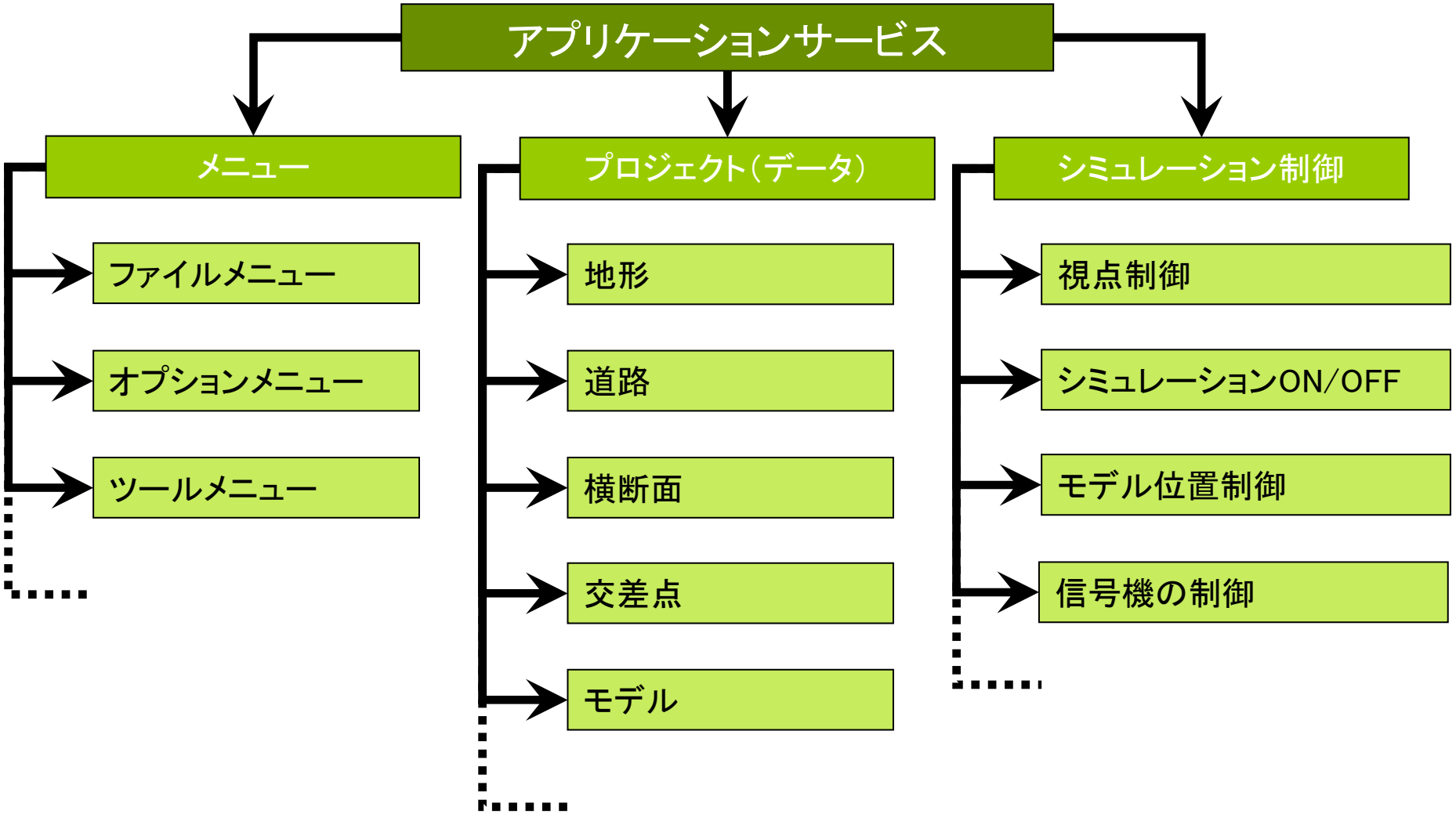
- 基本機能のカスタマイズ
  - 運転シミュレーションにおける車輌運動モデルのカスタマイズ
  - OpenGLコントロールの自由な描画

UC-win/Road API の構造

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- UC-win/Roadオブジェクトとは
  - プログラミングで言う一般的なオブジェクト
- 1つの要素に1つのオブジェクトの割り当て
  - 地形、道路、横断面、3Dモデル、カメラ、テクスチャ等
- 本APIでは一部のオブジェクトの中から一部の関数を公開

UC-win/Road API の構造



UC-win/Road SDK サンプルプログラム

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

- 線形IP情報出力ツール
- 道路データ作成
- 地形処理
- 道路の交通設定
- 人間キャラクタの制御　★
- 3Dモデル：メッシュの作成　★
- ログ取得機能　★、ユーザクリックイベント
- カメラ・視点の制御機能
- 環境設定
- 車両動作プロファイル編集
- 景観3Dモデルの出力

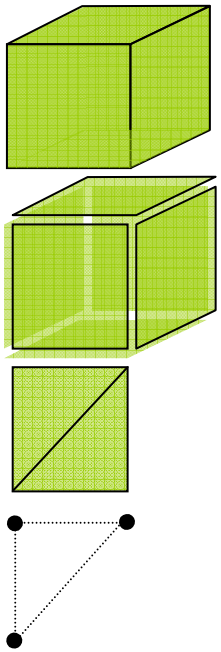
★…実演予定

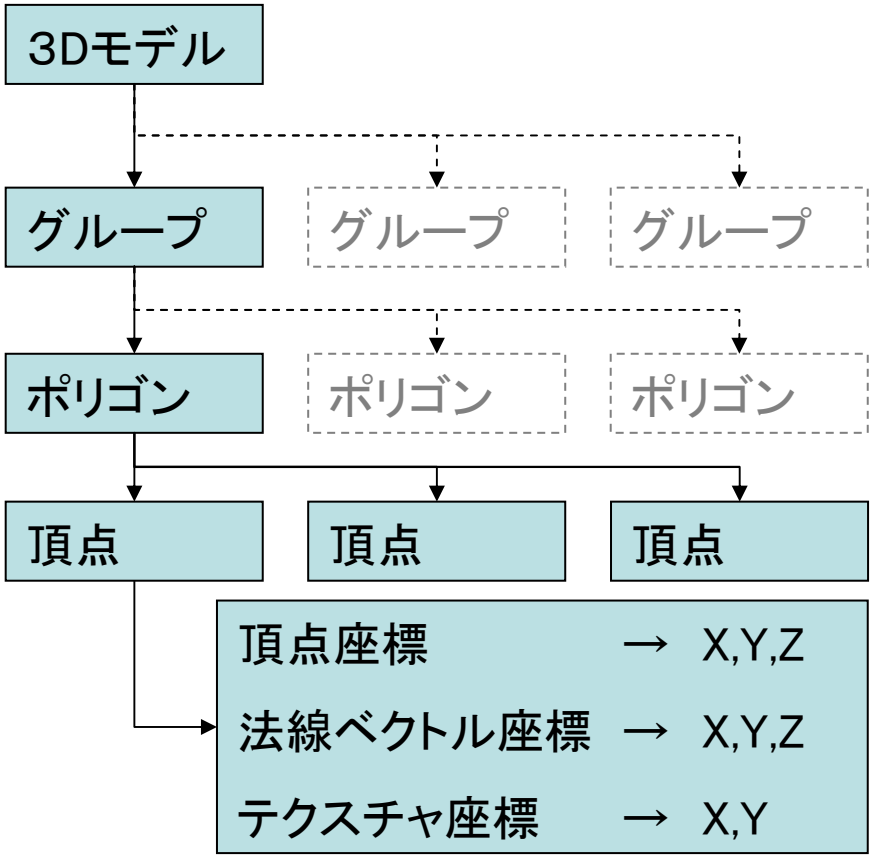
実演1：3Dモデル読込みツール

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

このプラグインでテキストファイルから3Dモデルのメッシュデータを読み込む。
メッシュデータによりUC-win/Roadに新たなモデルを作成し登録する。
UC-win/RoadのVR空間の中央にモデルを配置し、視点をモデルの上に移動させる。

3Dモデルの構造

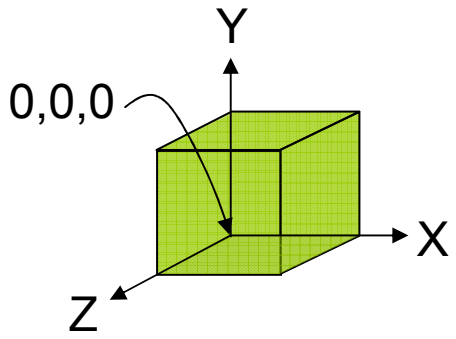




EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

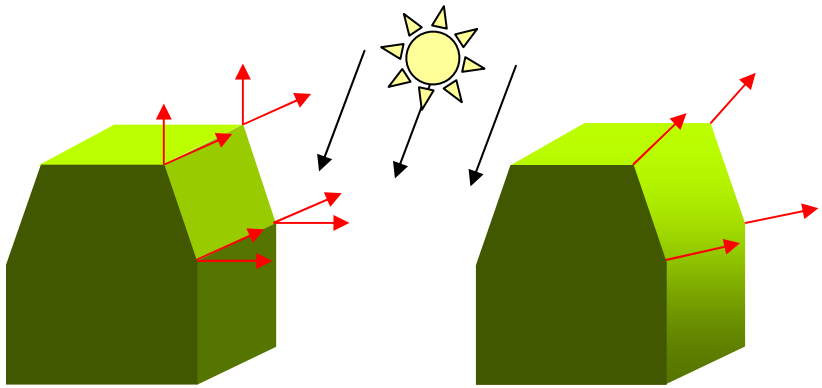
頂点情報について

3D座標 → X,Y,Z



法線ベクトル → X,Y,Z

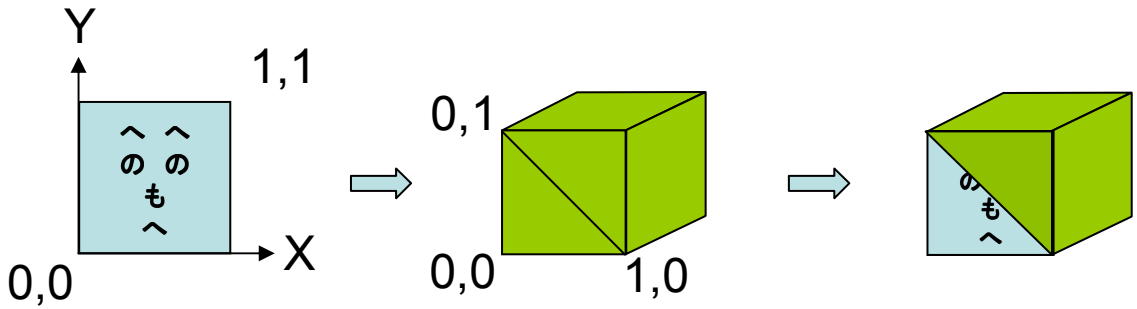
照明計算



それぞれのﾎﾟﾘｺﾞﾝに異なる法線ﾍﾞｸﾄﾙを利用すると照明の効果が違う。

それぞれのﾎﾟﾘｺﾞﾝに同じ法線ﾍﾞｸﾄﾙを利用すると照明が滑らかに擦り付ける。

テクスチャ座標 → X,Y



実演1：3Dモデル読込みツール（仕様1）

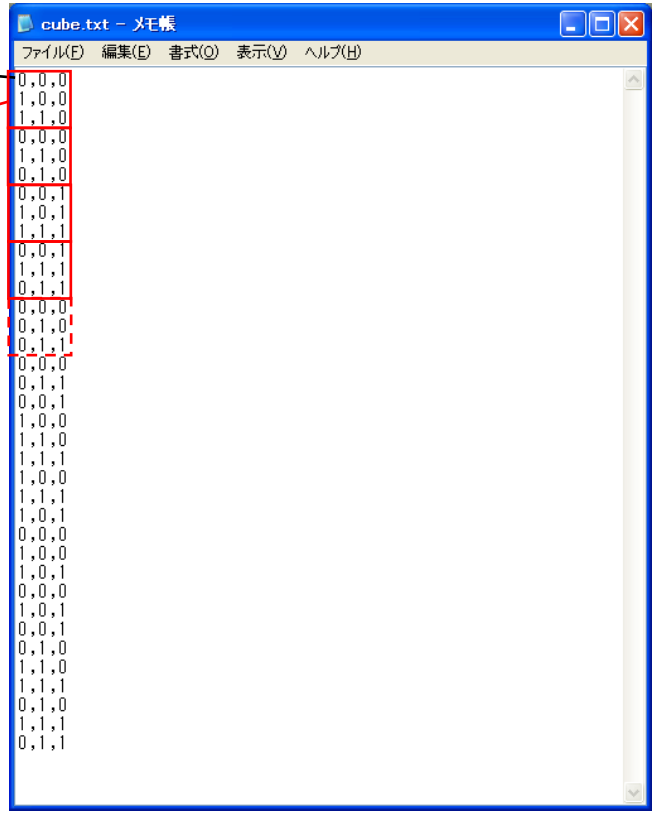
EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

仕様1では頂点の3D座標のみを読込み、3Dモデルの作成を行う。
UC-win/RoadのVR空間の中央にモデルを配置し、視点をモデルの上に移動させる。

ファイル形式

頂点の座標 X,Y,Z

1つのポリゴン（三角形）



ソースコード

ファイル名：CreateModel.pas

```
procedure AddModel(project : IF8ProjectForRoad; filename : WideString);
```

project ： UC-win/Roadの現在のプロジェクト
filename ： 読み込むテキストファイルのフルパス

実演1:3Dモデル読込みツール(仕様1)

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

利用するSDK及びDelphiの関数

TStringList (文字列処理用のクラス)

+ function LoadFromFile → テキストファイルの読み込み
+ property Text : String → 全体のテキスト
+ property CommaText : String → カンマの区切りを改行に変換
+ property Lines[index : integer] : String → 行毎のテキスト
+ property Count : Integer → 行数

IF8Project (プロジェクトのインタフェース)

+ function CreateThreeDModel : IF8ThreeDeeStudio → 新規モデル作成

  新規モデルにはグループが1つ存在する。

+ function MakeModel (currentModel : IF8ThreeDeeStudio ; xx , yy : double) : IF8ModelInstance

  currentModel : 配置するモデルのインタフェース
  xx : 平面「_x」座標
  yy : 平面「_z」座標

+ property refreshModels : boolean → 画面をリフレッシュするときにモデルの更新を行うように「True」にする必要がある。

実演1:3Dモデル読込みツール(仕様1)

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

IF8ThreeDeeStudio (3Dモデルのインタフェース)

+ procedure AddTrianglesToGroup (groupIndex : integer; triangles : F8TriangleArrayType) → 指定グループにポリゴンを追加する。

groupIndex → グループ番号

Triangles → ポリゴン情報

+ procedure Center → モデルの重心の計算を行う。

+ property name : WideString → モデルの名称

F8TriangleArrayType = array of F8TriangleType → ポリゴンのリスト

F8TriangleType = array [1 .. 3] of GLPointType → 3点のリスト

GLPointType = array [1 .. 4] of GLfloat → 頂点

実演1：3Dモデル読込みツール（仕様2）

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

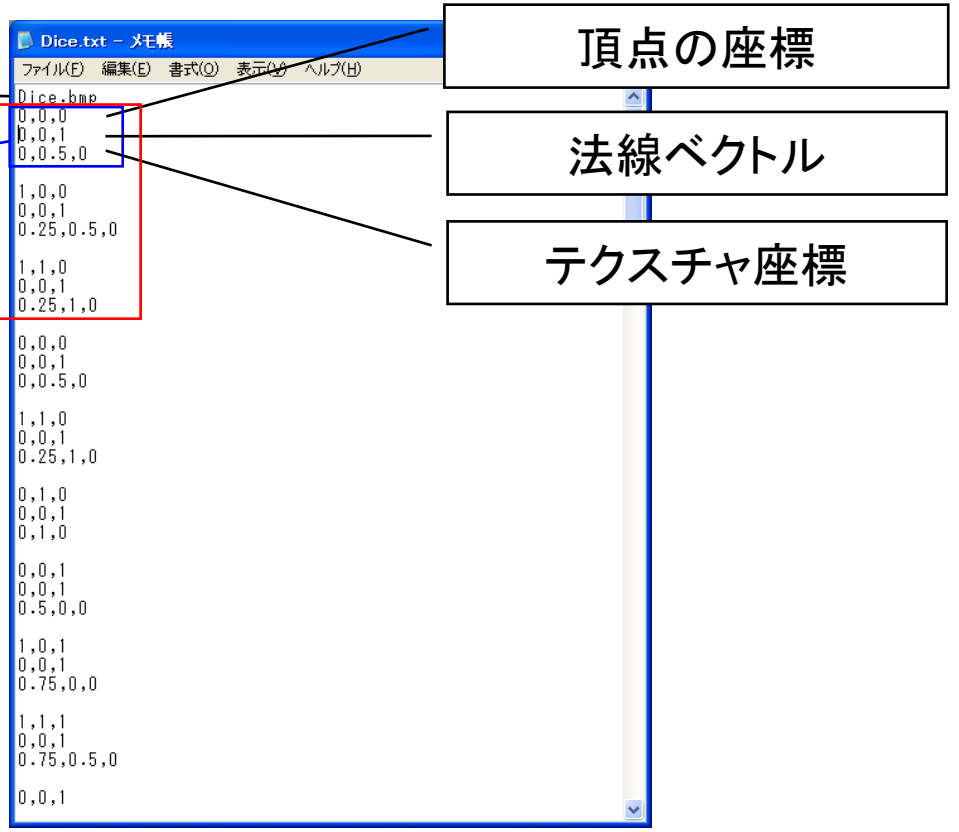
仕様2ではメッシュの座標及び法線ベクトルとテクスチャの座標を読込み、3Dモデルの作成を行う。
UC-win/RoadのVR空間の中央にモデルを配置し、視点をモデルの上に移動させる。

ファイル形式

テクスチャファイルの名前

頂点の群

1つのポリゴン（三角形）



頂点の座標

法線ベクトル

テクスチャ座標

ソースコード

ファイル名：CreateModel.pas

```
procedure AddModelWithTexture(project : IF8ProjectForRoad; filename : WideString);
```

project ： UC-win/Roadの現在のプロジェクト
filename ： 読み込むテキストファイルのフルパス

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

利用するSDK及びDelphiの関数

IF8ThreeDeeStudio （3Dモデルのインタフェース）

+ procedure AddTexturedTriangleToGroup (groupIndex : integer ; triangles : F8TexturedTriangleArrayType) → 指定グループにポリゴンを追加する。頂点、法線ベクトル、テクスチャ座標が設定される。

groupIndex → グループ番号

Triangles → ポリゴン情報

+ property groupHas3DSTextures [index : integer] : boolean → テクスチャの設定を行った場合「True」にする必要がある。

+ property groupUse3DSTextureCoord [index : integer] : boolean → テクスチャの設定を利用したい場合「True」にする必要がある。

F8TexturedTriangleArrayType = array of F8TexturedTriangle → ポリゴンのリスト

F8TexturedTriangle = record
vertex : F8TriangleType; → 頂点座標
normal : F8TriangleType; → 法線ベクトル座標
textureUV : F8TriangleType; → テクスチャ座標
end ;

実演2：人間キャラクターの制御

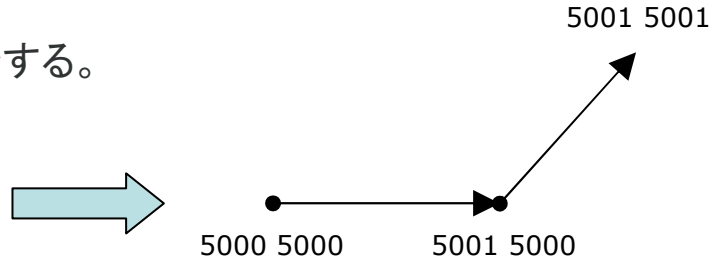
EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

UC-win/Roadの空間に歩行者を追加する。
追加した歩行者をリアルタイムに制御する。
歩行者の経路はテキストファイルから読み込む。

プログラムの仕様

- 3Dポイントをテキストファイルから読込む→経路
  - 1行毎に1点
  - X,Y,Z座標の区切りを「TAB」とする。

X	Y	Z
5000	5000	20
5001	5000	20
5001	5001	20

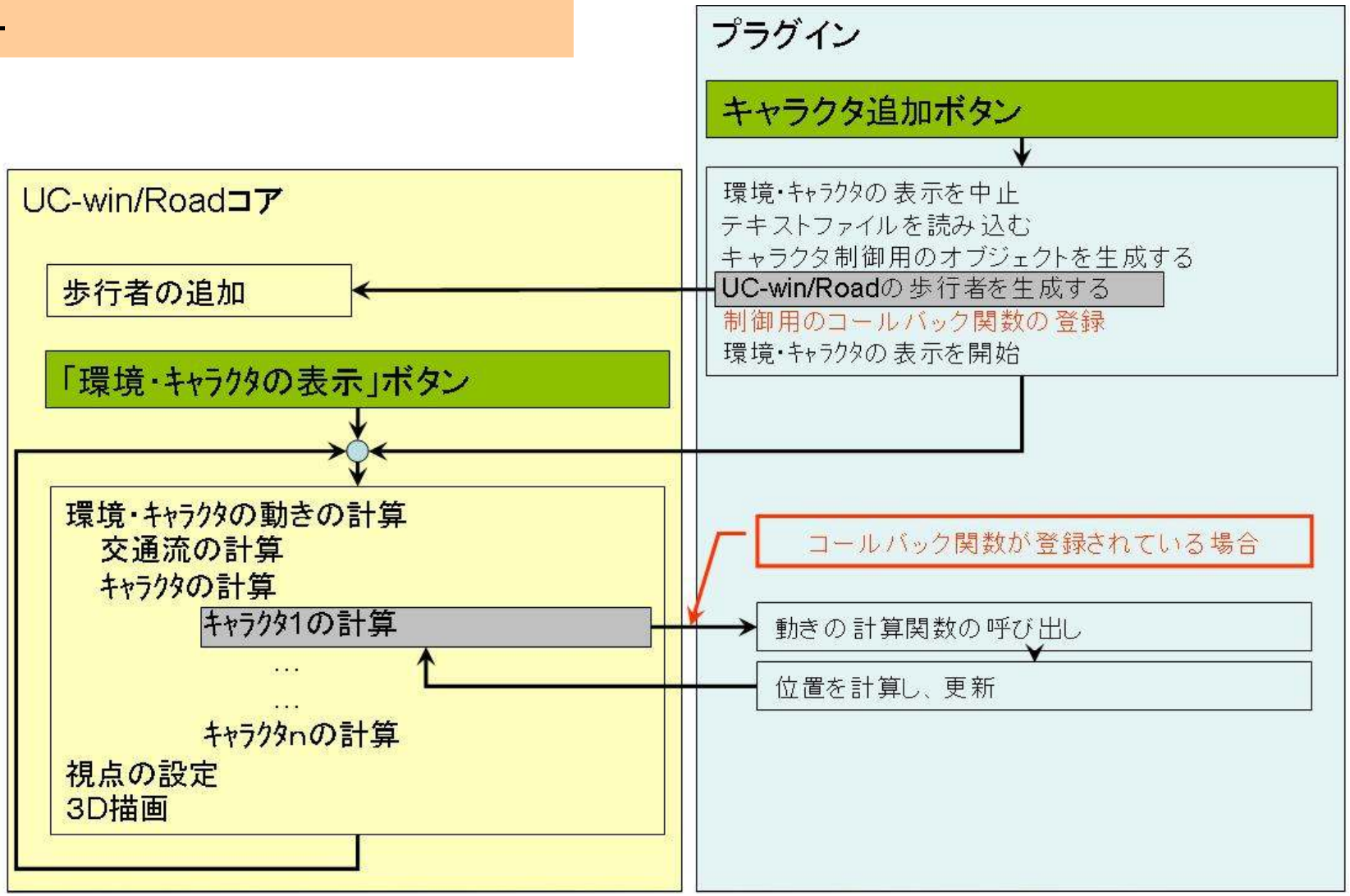


- テキストファイル毎に1人の歩行者を追加し経路に沿って動きと位置座標をリアルタイムに更新する。
  - 移動の速度：ファイルに定義された点の間は0.5秒で歩く。
  - キャラクタの向き:ファイルに定義された点を狙う。
  - ループ

実演2：人間キャラクターの制御

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

フロー



EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

利用するSDK及びDelphiの関数

IF8ProjectForRoad （プロジェクトのインタフェース）

+ property numberOfCharacters : Integer; → 登録されているmd3キャラクタの個数
+ property character [i : integer] : IF8QuakeIII; → 登録されているmd3キャラクタにアクセス
+ function CreateCharacterInstance (model : IF8QuakeIII; const transient : boolean) : IF8CharacterInstance; → 指定md3モデルを歩行者として追加する。
  model → md3モデル
  transient → 一時オブジェクトとするかどうか（Trueの場合は交通のリセットで削除され、データに保存されない）
+ procedure DeleteCharacterInstance (character : IF8CharacterInstance); → 指定歩行者を削除する（ポインタで指定）
  character → 削除する歩行者のポインタ
+ procedure DeleteCharacterInstance (i : Integer); 指定方向者を削除する（番号で指定）
  i → 削除する歩行者の番号

実演2：人間キャラクターの制御

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

IF8CharacterInstance （キャラクタインスタンスのインタフェース）

+ property instancePosition : GLPointType; → 位置情報（OpenGL座標系）
+ property onMove : OnMoveEventProc; → 位置計算時に呼び出す関数をここに設定する。（コールバック関数）
+ property onDestroy : OnDestroyEventProc; → キャラクタ削除時に呼び出す関数をここに設定する。（コールバック関数）
+ property yawAngle : double; → ヨー角
+ property pitchAngle : double; → ピッチ角
+ property rollAngle : double; → ロール角
+ property walkForward : boolean → md3モデルのアニメーションの再生を制御する：Trueの場合は通常の再生、Falseの場合は逆再生

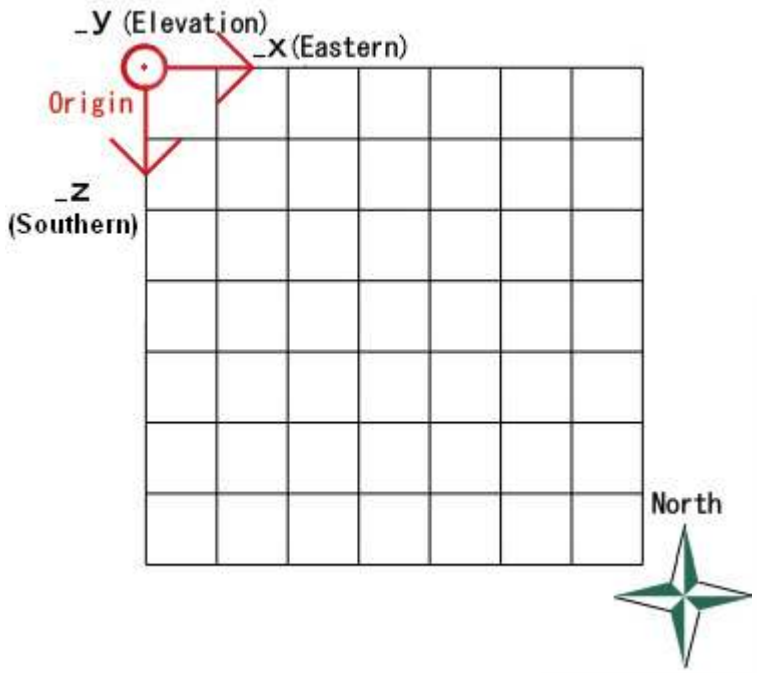
コールバック関数のヘッダ

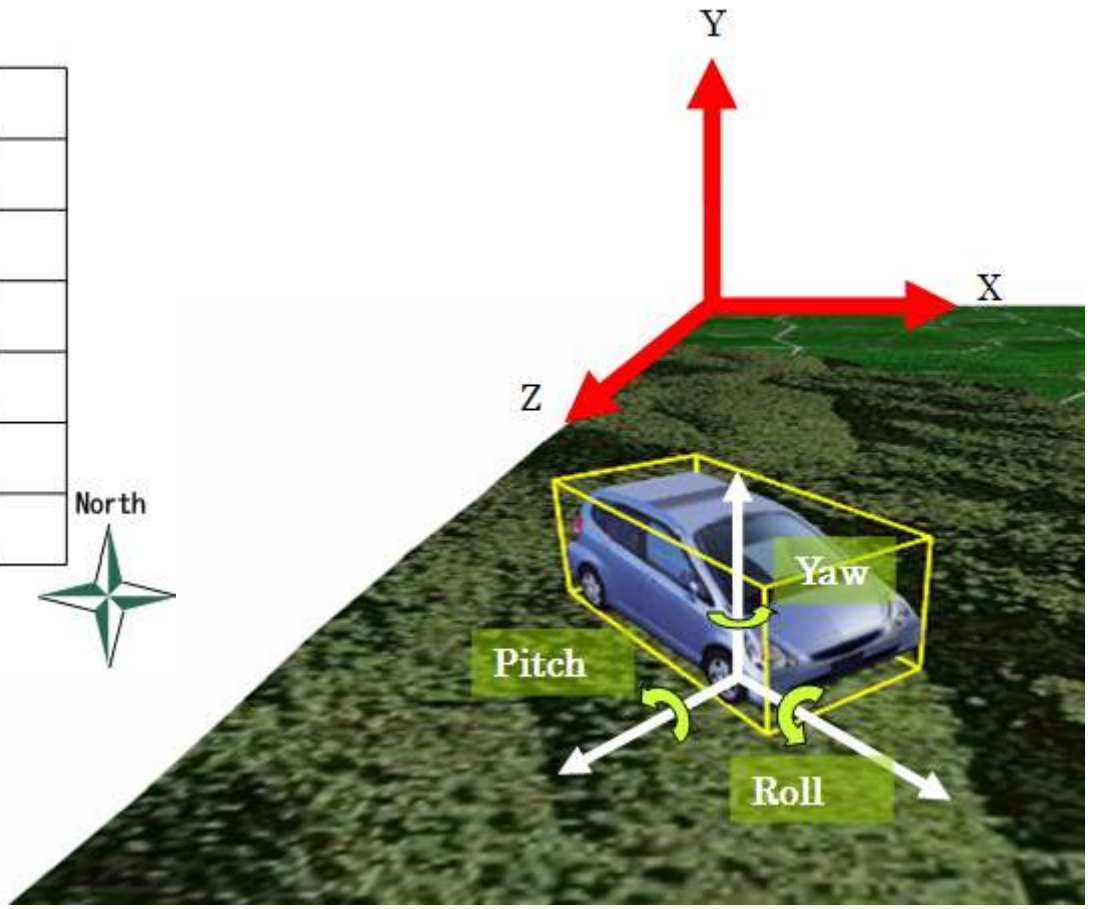
+ OnMoveEventProc = procedure (dTimeInSeconds : double; Instance : IF8MovingObjectInstance) of Object; → モデル及びキャラクタの位置計算を行う際、呼ぶ関数の形
  dTimeInSecond → 前の表示したフレームからの時間差
  Instance → 一計算が要求されているオブジェクトのポインタ
+ OnDestroyEventProc = procedure (Instance : IF8MovingObjectInstance) of Object; → オブジェクトが削除される際、呼ぶ関数の形
  Instance → 削除されるオブジェクトのポインタ

実演2：人間キャラクターの制御

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

OpenGL座標系

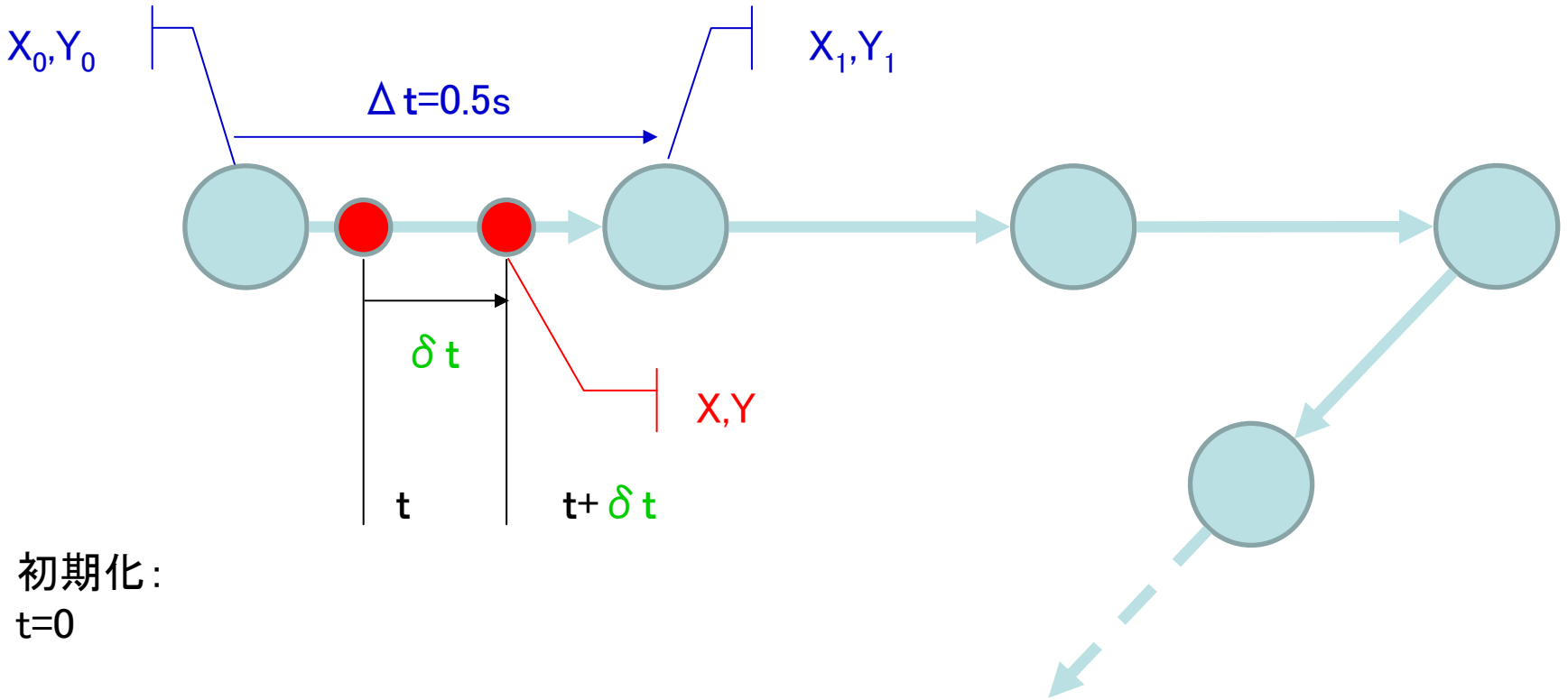




実演2：人間キャラクターの制御

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

計算方法：線形補間



初期化：
t=0

初期化：
X_0,Y_0,X_1,Y_1を求める
$X=X_0+(X_1-X_0) * (t+\delta t)/\Delta t$
$Y=Y_0+(Y_1-Y_0) * (t+\delta t)/\Delta t$
$t= t+\delta t$

実演3:ログ出力

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

利用するSDKとDelphi関数

IF8LogServer (ログサーバ)

+ procedure RegisterOnLogProcedure(method: TMethod); → ログ時に呼ばれるコールバック関数の登録
+ procedure UnRegisterOnLogProcedure(method: TMethod); → 登録したコールバック関数の登録解除
+ procedure StartLogs; → ログ開始
+ procedure StopLogs; → ログ停止

コールバック関数のヘッダ

OnLogServerPushLogProc

= procedure(dTimeInSeconds : double; instance : IF8DBObject; group : TLogExportOption) of Object;

→ ログ時に呼ばれるコールバック関数の型
dTimeInSecond → 前の表示したフレームからの時間差
Instance → 一計算が要求されているオブジェクトのポインタ
TLogExportOption = (→ ログ出力オプション ・・複数設定可
_leoUsersVehicle, //* 自車
_leoDriverInFront, //* 自車と同じ車線を走行する前方車両.
_leoSurroundingMovingObjects, //* カメラ位置から所定の範囲内の車両.
_leoOtherMovingObjects //* _leoSurroundingVehiclesがログサーバのログターゲットに設定された場合は、その範囲を除くシーン内の全ての可動モデル
);

実演3:ログ出力

EMBARCADERO TECHNOLOGIES
DEVELOPER CAMP

function Supports(**const** Instance: IInterface; **const** IID: TGUID; **out** Intf): Boolean;

指定したインターフェースがサポートされているかの確認関数
Instance : チェックされるべきインターフェース
IID : 指定インターフェース
Intf : インターフェースのポインタ
戻り値:指定したインターフェースがサポートされていればTrueが戻り、Intfに所定のポインタが設定される。

IF8VehicleLogs(車両のログ)

IF8InstanceLog(車両以外の可動モデル用ログ)