【xx】Delphi/C++チュートリアル セッション

「Delphiでつくる！3DVRシミュレーション」
～UC-win/Road SDK　の活用～

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

EMBARCADERO
TECHNOLOGIES.

概要

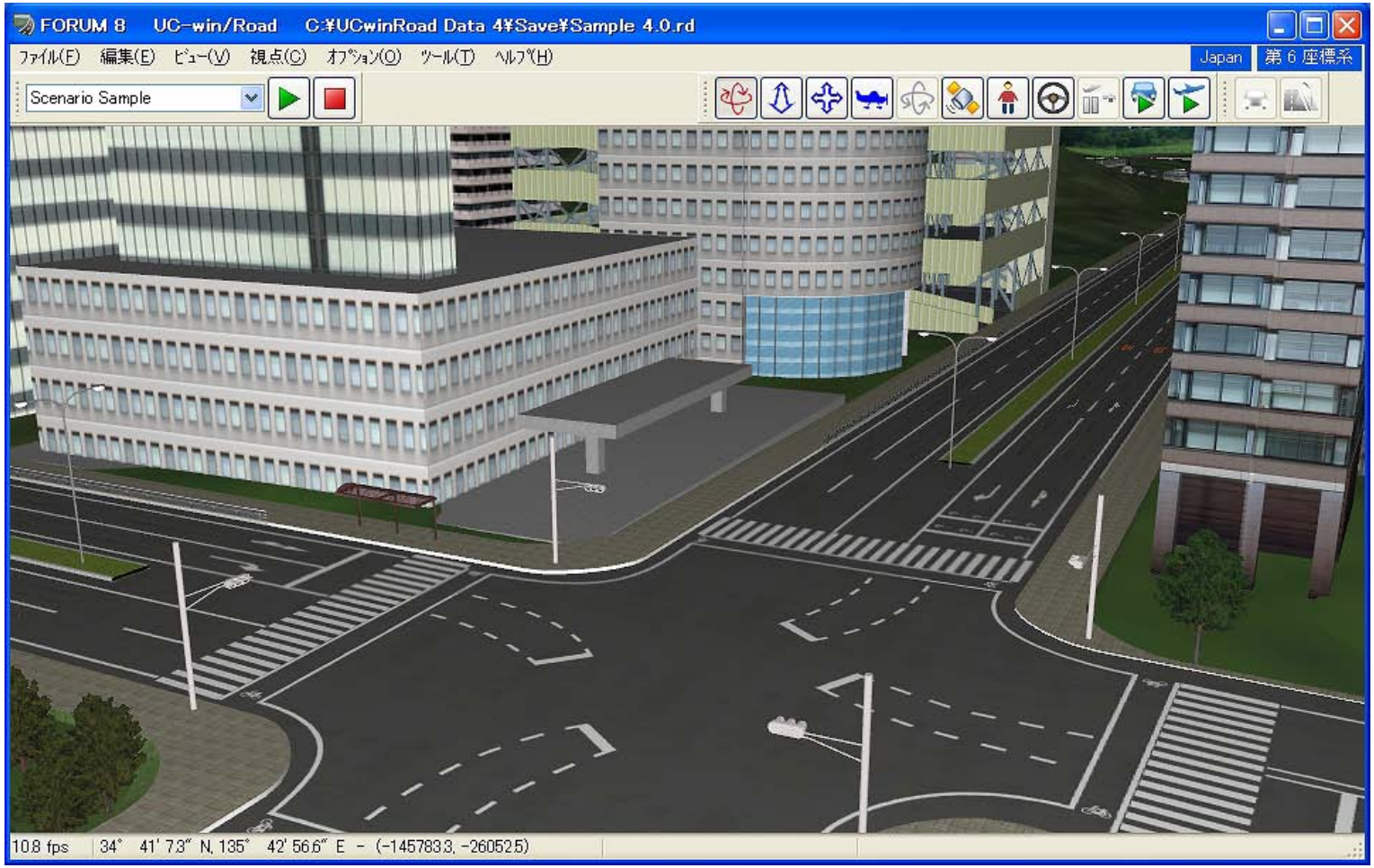
UC-win/Road Ver4.0

- 3次元のバーチャル・リアリティー（VR=仮想現実）を簡単なPC操作で作成できる優れた機能をサポート。
- わかりやすい手順と操作で大規模な3次元空間を驚くくらい短時間に作成できます。
- 走行シミュレーションに加え、日照シミュレーション、交通流シミュレーション、マニュアルドライブシミュレーションなどに対応。

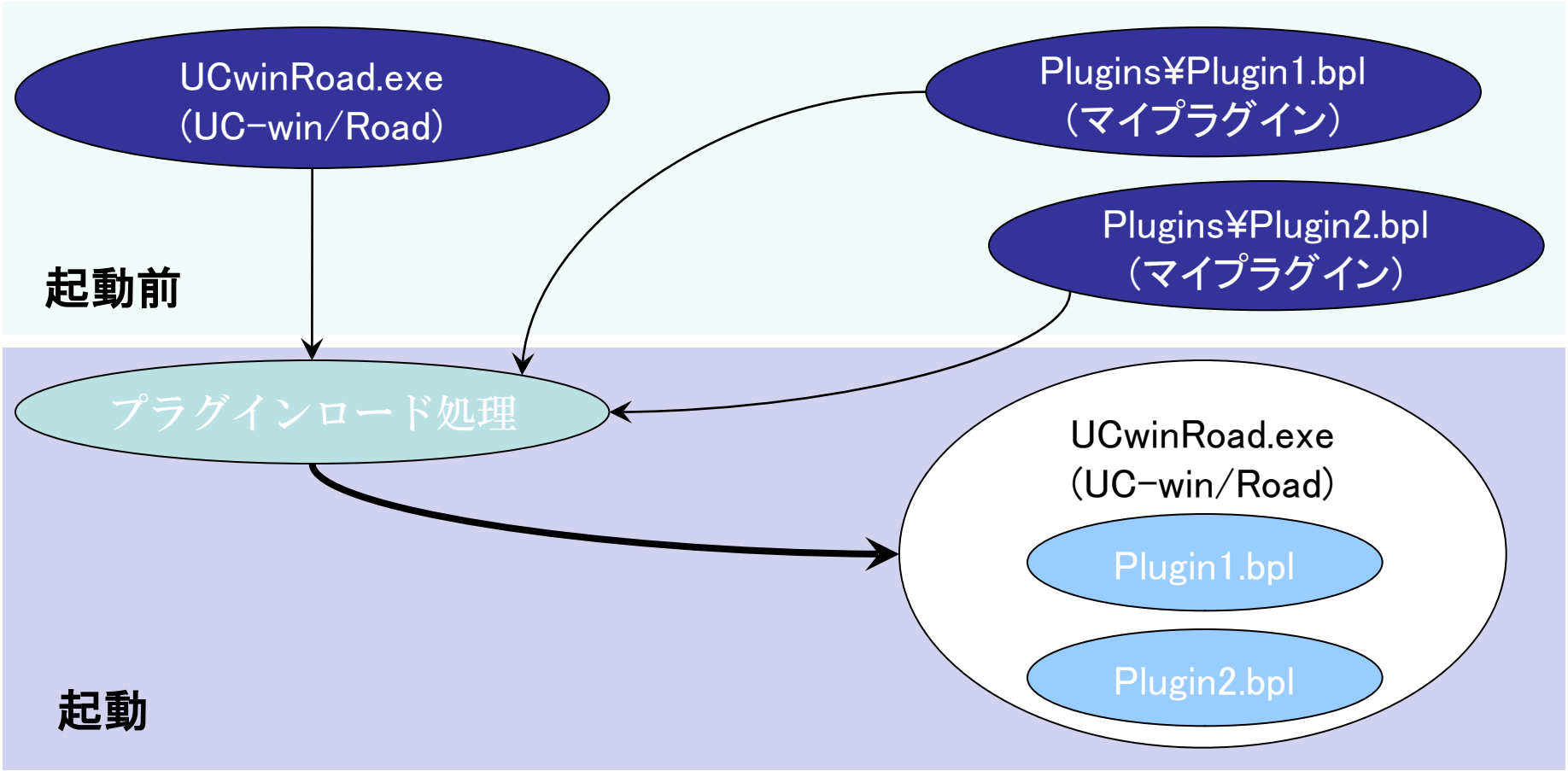
UC-win/Road SDK

- 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

UC-win/Road



UC-win/Road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 API のサポート内容

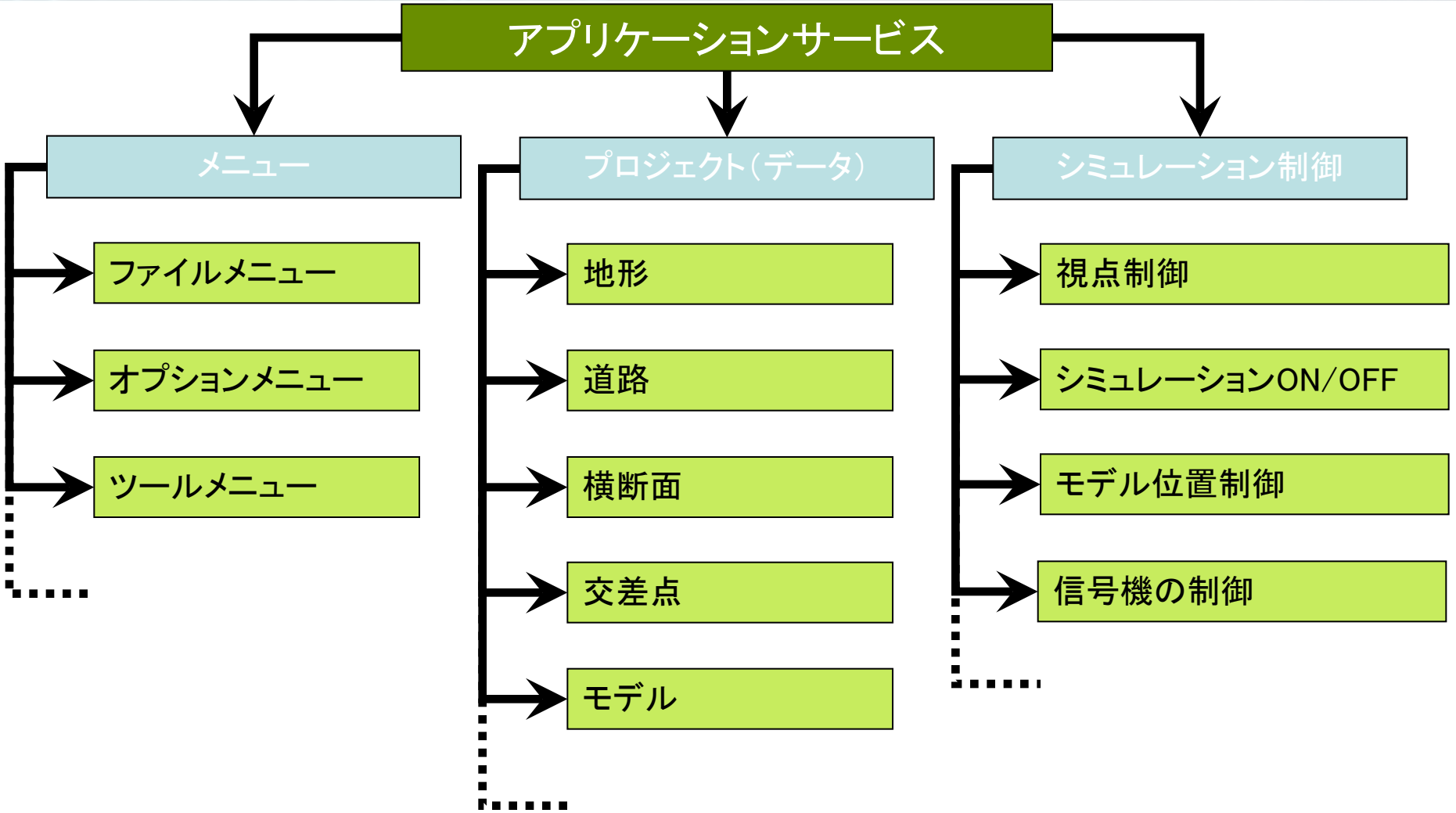
- 編集機能：静的なデータの読み取り、書き込み、編集
  - 地形、航空写真、道路、交差点、交通、3Dモデル
- GUI機能：ユーザインタフェースのカスタマイズ
  - メイン画面にコントロールの追加、既存コントロールの制御
- インタラクション機能：シミュレーション状況の取得
  - キーボード、マウス、ゲームコントローラ操作によるコールバック関数の呼び出し
  - 視点状況
  - ログ出力

UC-win/Road API のサポート内容2

- 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



UC-win/Road SDK サンプルプログラム

- 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
- 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

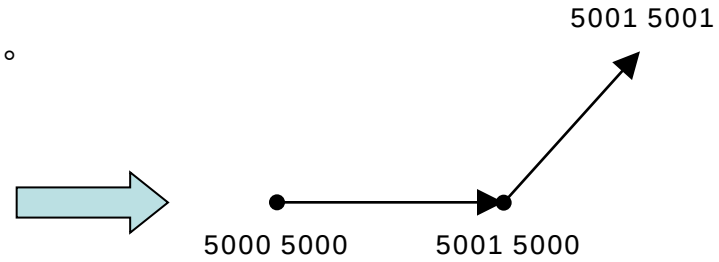
実演：人間キャラクターの制御

VR空間内の歩行者をテキストファイルの経路に沿ってリアルタイムに制御する。

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。

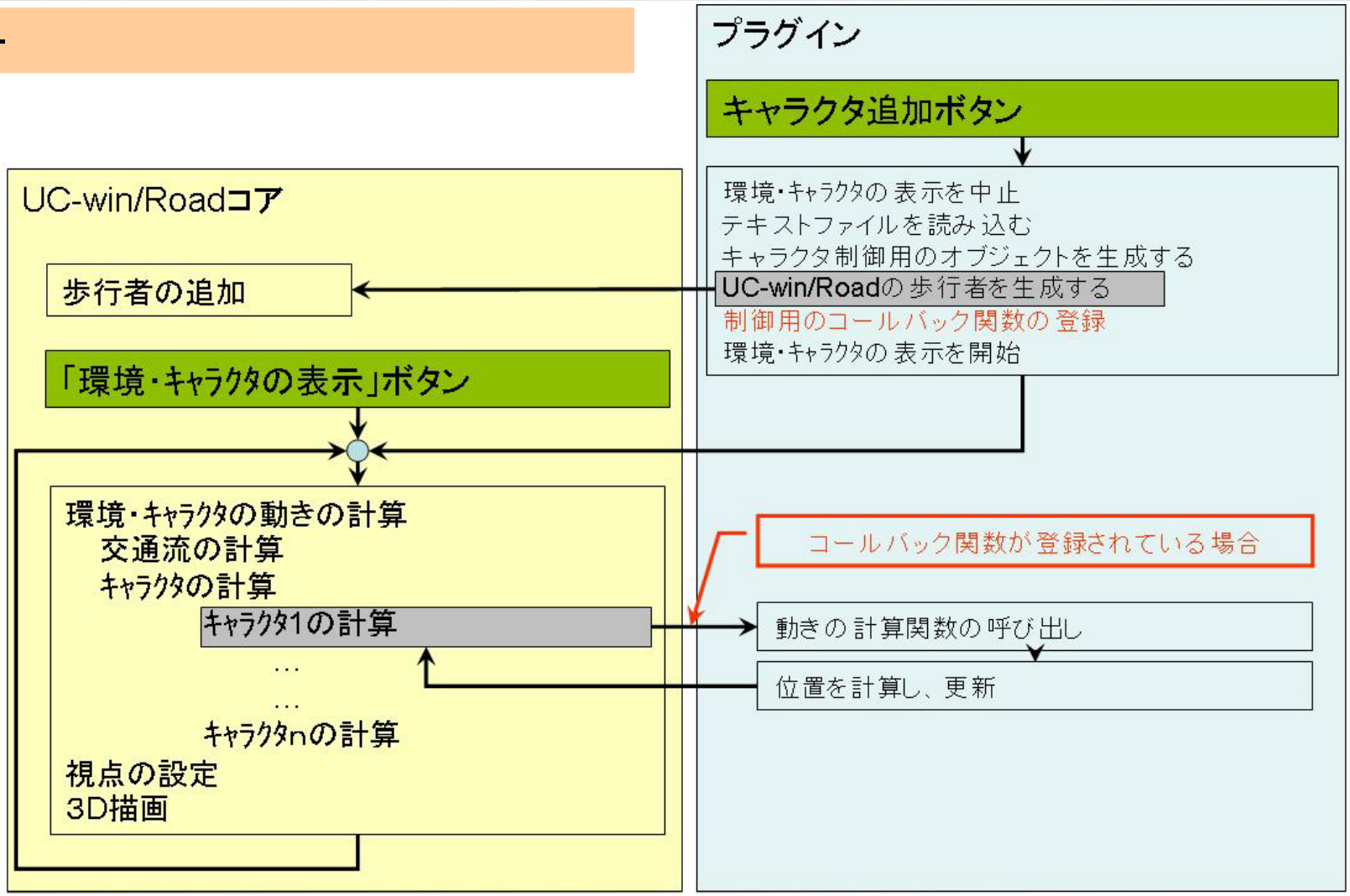
X	Y	Z
5000	5000	20
5001	5000	20
5001	5001	20



- 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：人間キャラクターの制御

フロー



実演：人間キャラクターの制御

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：人間キャラクターの制御

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

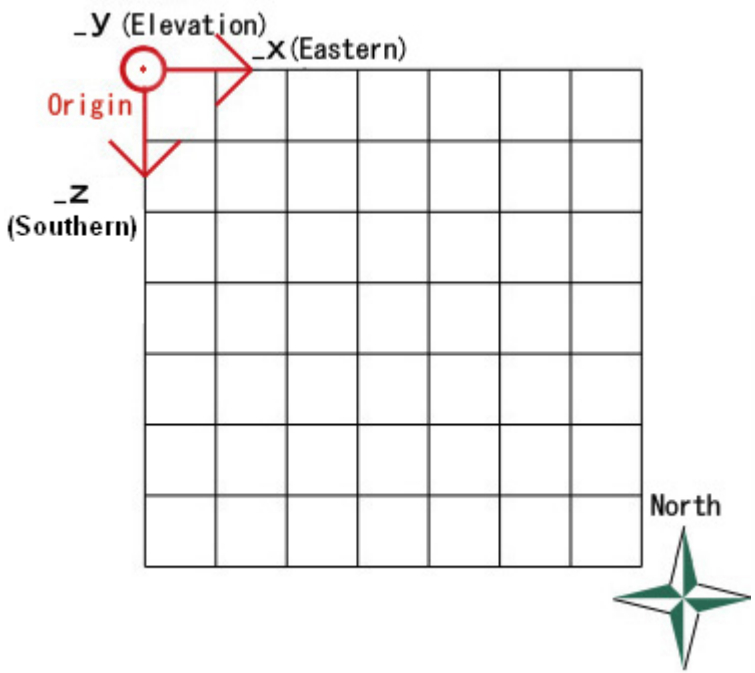
+ property instancePosition : GLPointType; → 位置情報（OpenGL座標系）
+ property onMove : OnMoveEventProc; → 位置計算時に呼び出す関数をここに設定する。（コールバック関数）
+ property onDestroy : OnDestroyEventProc; → キャラクタ削除時に呼び出す関数をここに設定する。（コールバック関数）
+ property yawAngle : double; → ヨー角
+ property pitchAngle : double; → ピッチ角
+ property rollAngle : 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

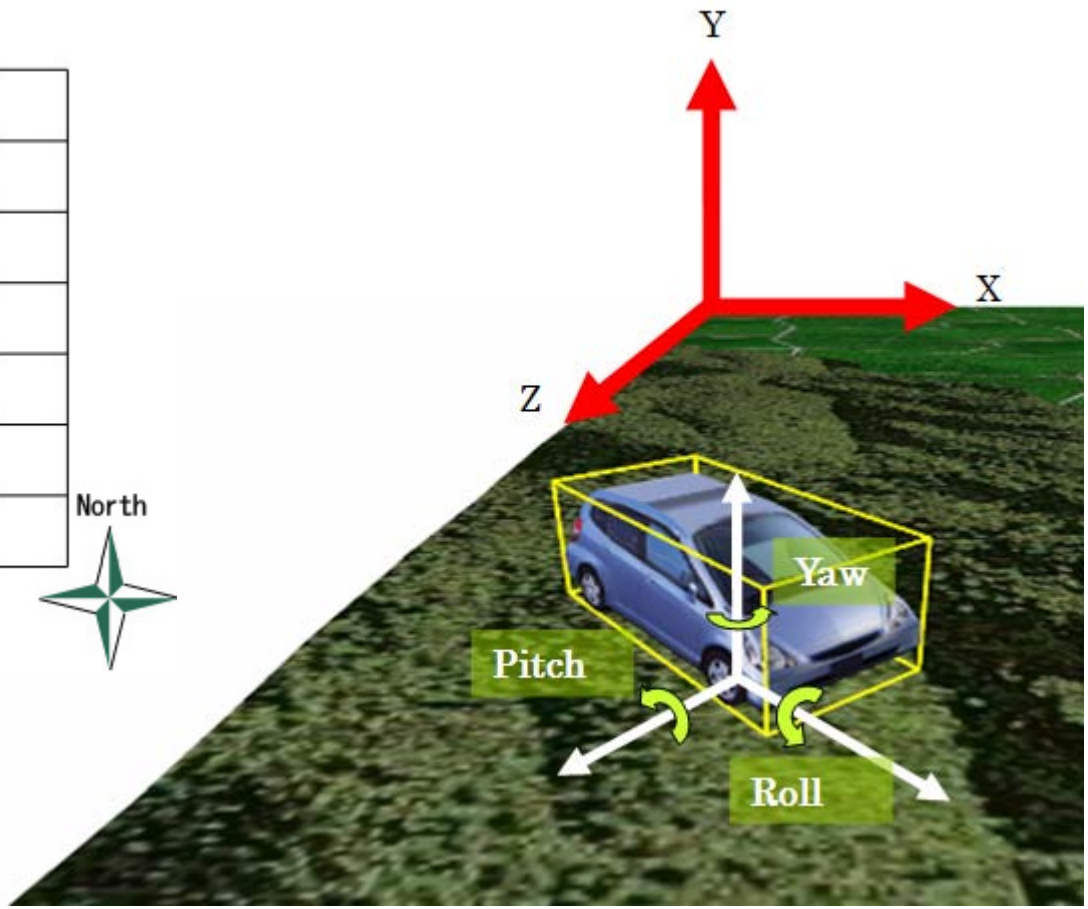
コールバック関数のヘッダ

+ 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：人間キャラクターの制御

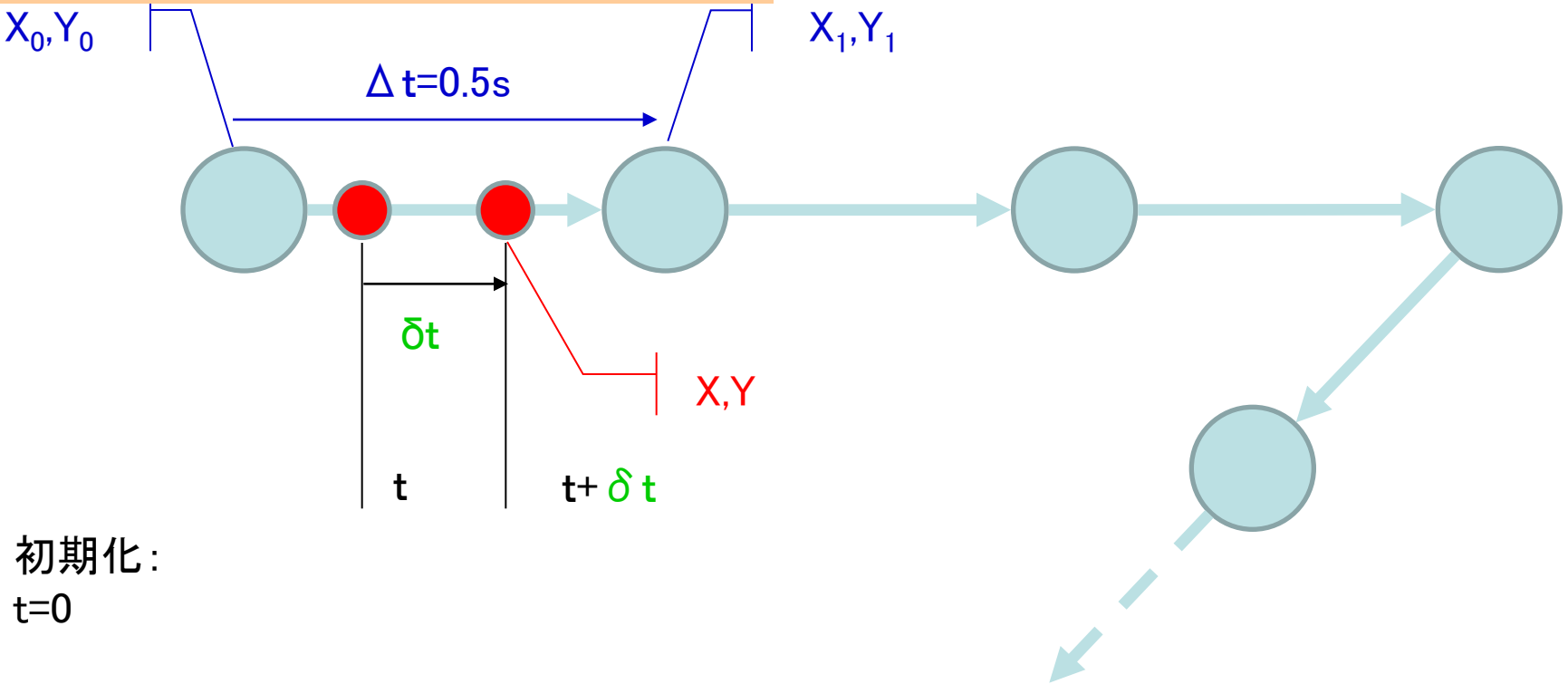
OpenGL座標系





実演：人間キャラクターの制御

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
$X=X_0+(X_1-X_0)*(t+\delta t)/\Delta t$
$Y=Y_0+(Y_1-Y_0)*(t+\delta t)/\Delta t$
$t=t+\delta t$