A Proposal for a

Procedural Terrain Modelling Framework

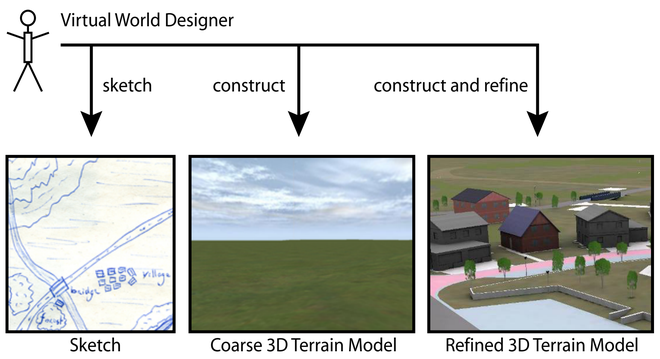
R.M. Smelik, T. Tutenel, K.J. de Kraker and R. Bidarra

Motivation

Virtual worlds for games and simulations are increasing in size, detail and visual realism. Therefore, *manual* virtual world design requires more *effort, time, artistic and 3D modelling skills, money,* etc. There is a clear need for **higher productivity** for virtual world designers. We propose to **automate** virtual world modelling in a **framework** that supports **declarative terrain modelling**.

Current Terrain Modelling Workflow

Manual virtual world modelling is a process of **iterative refinement**:



Starting with a **sketch,** the designer manually builds the terrain model by **experimentation**.

Procedural Content Generation Methods

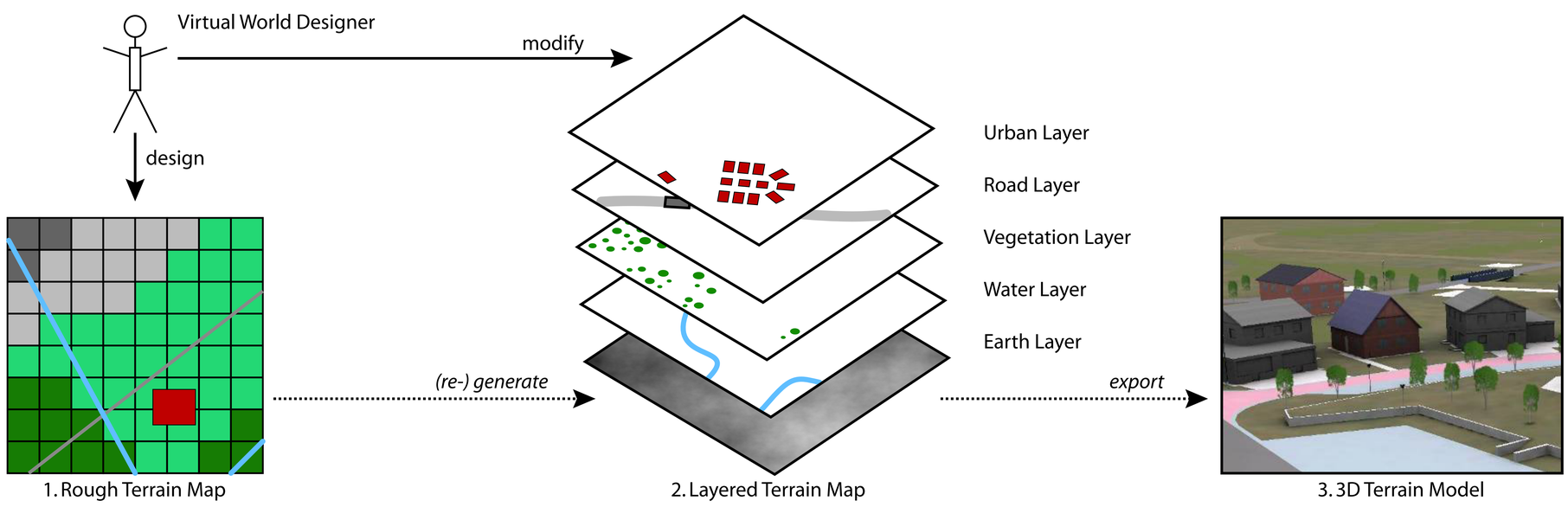
- Methods that generate content through rules, algorithms and random numbers;
- Numerous procedural methods have been developed, e.g.:
  - Height map generation algorithms;
  - Automatic distribution of vegetation and plant models generation;
  - Generation of urban environments (road networks, buildings, facades, and interiors).
- Therefore, these methods are a promising approach to automatic virtual world construction.

Current limitations:

1 No integrating *framework* exists that combines methods in a *usable* way;

2 In-depth *knowledge* of method required to predict effect of method parameters on outcome;

3 User is unable to declare *intentions* and has little *control* over process: **trial and error**



High-Level Framework Requirements

1 **Terrain Sketching** The framework requires high-level user input in the form of a *sketch* of the terrain and important terrain *features*.

2 **Usability** User input consists of *intuitive*, *result-oriented* parameters, mapped to procedural method parameters.

3 **Terrain Generation** After the designer has declared his terrain, the framework is to generate a terrain that matches the user specification at large, but has, on a small scale, a high level of detail and variations.

4 **Manual Editing** The workflow is to support further *manual* editing and fine-tuning, as well as regenerating areas of the terrain.

5 **Visualisation** The framework should provide a clear 3D view of the terrain.

6 **Results** The terrain model should be exportable to different relevant formats.

Procedural Terrain Modelling Workflow

1 Sketch of terrain by designer, using rough terrain map and global terrain parameters;

2 Procedural generation of detailed terrain map; further automatic and manual refinement; a *layer mechanism* is used to improve the *flexibility* and *adaptability* of the terrain model:
- **Urban Layer** cities, towns, farms,
- **Road Layer** highways, local roads, bridges,
- **Vegetation Layer** bushes, trees, shrubs,
- **Water Layer** rivers, canals, lakes, oceans,
- **Earth Layer** elevation and soil data.

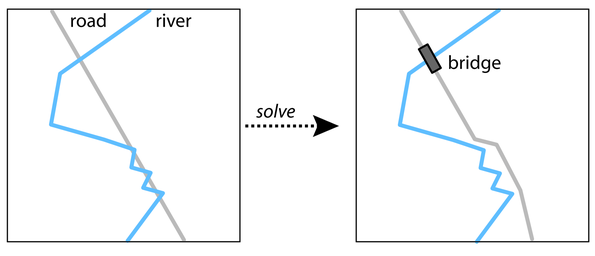
3 Automatic export to, among other things, a 3D terrain model, GIS vector data, or navigation maps for Computer Generated Forces.

A *procedural terrain modelling framework* fits an *iterative design process* and combines existing procedural methods in a new, coherent, and useful way.

Ongoing Work

Important design challenges:

- Automatic solving techniques for *interactions* and *conflicts* between terrain layers;
- Mapping of user input parameters to procedural method parameters.



Increasing demands in virtual world modelling require a shift from **manual construction** towards **declarative modelling**. We expect that a **procedural terrain modelling framework** will contribute to this.

TNO
GATE
http://gate.gameresearch.nl/
TU Delft