# WorldForge

Continent Scale Persistent Online Worlds

## WorldForge Overview

- Born out of desire for free MMORPG
- Many often conflicting goals
- Too many different ideas for one game
- Build an engine for many games
- Open Source and Open Content
- Source Code in the hands of the Enemy

## Many Games -> Lots of World Data

- Shipping huge datasets over download is hard
- No big downloads
- No offline patching
- No long connection times

## Terrain

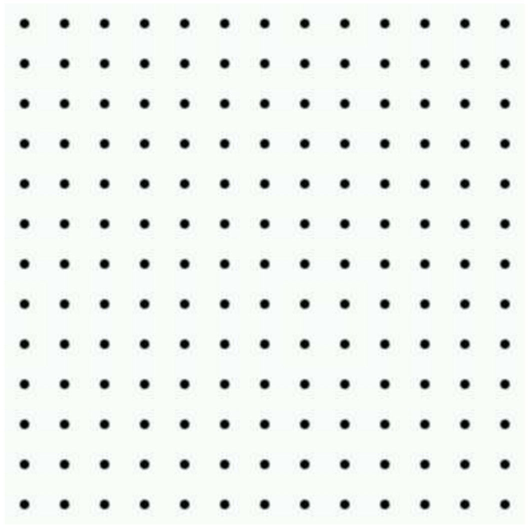
- Heightfields chosen early on
  - Easy to edit
  - Easy to load and render
- It takes alot of heightfield to store a world
  - Too much for the client to have it all
- Various approaches to handling data
  - Download tile data from server
  - Ship tiles as image files (content)
  - Generate tiles at the client

## Terrain generation algorithm properties

- Repeatable and Portable
- Translocatable
- Seamless
- Fast

## Quasi Random Mid-point Displacement

- Diamond Square Interpolation



## The Mathematics

- Key calculation is the displacement value
- Random number generated using Mersenne Twister
- Seeded from the heights of the control points

$$Displacement = \frac{rand * heightDifference * roughness}{(1 + depth^{falloff})}$$

- roughness and falloff provide a degree of control

## Problems with Diamond Square

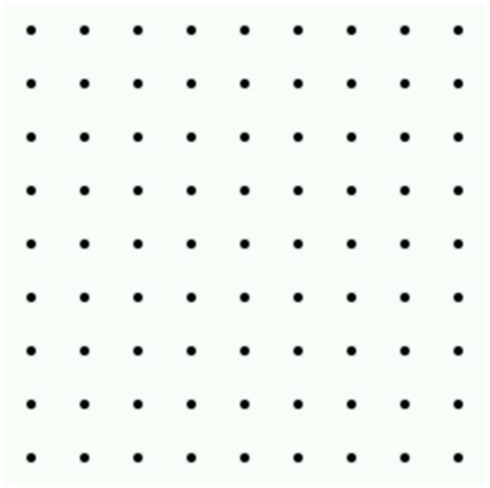
- Generating one tile requires 9 control points

  and calculating redundant points
- Changing 1 control point affects 16 tiles

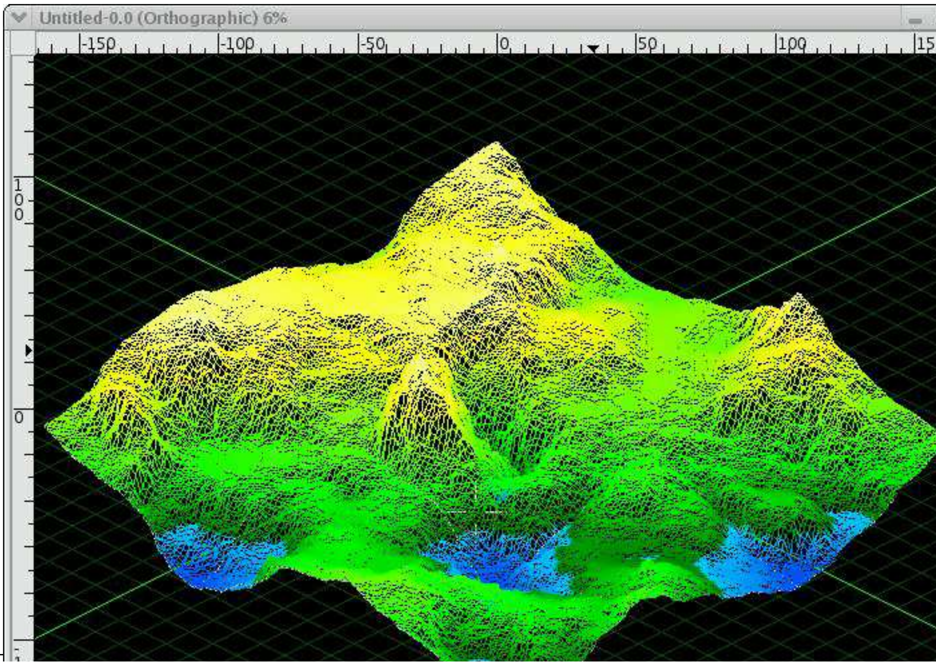
  If not, seams won’t match
- Tends to result in peak at control point

## Alternative Algorithm

- Square Interpolation



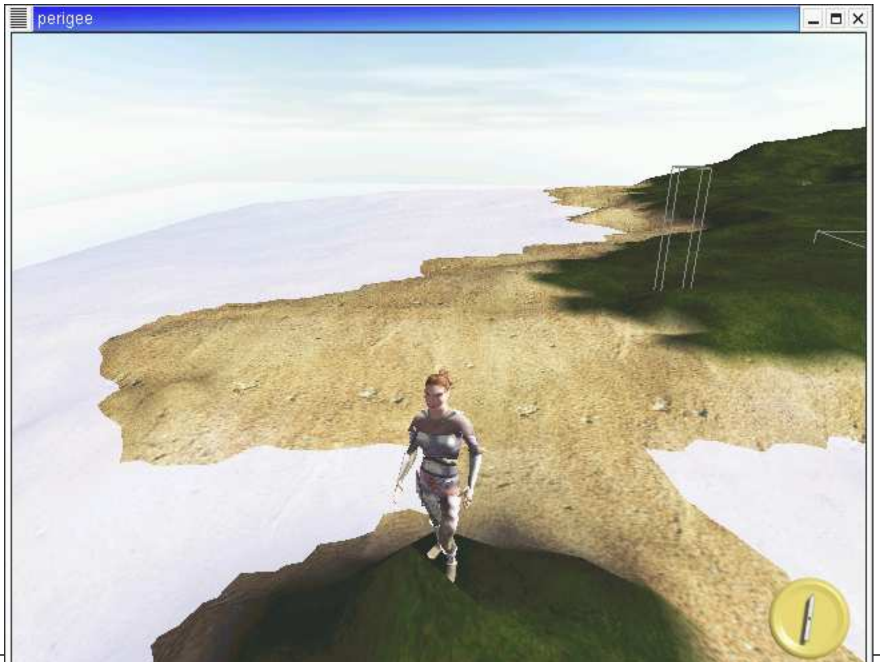
## Resulting heightfields



## Surface texturing

- Surface texture calculated from heightfield
  - Height, slope, climate.
- Traditional two pass texture not really flexible enough
- One pass for each terrain type
  - Rock, Sand, Grass, Snow
- Alpha texture generated from height data

## Resulting surfaces



## What Generated Terrain Won’t Handle

- Builders have to accept limited control
- Flat surfaces for building
- River channels
- Earth works
- Handled using modifiers

## Vegetation

- Good looking landscape requires vegetation
- Just as much data as terrain itself
- Required at the server and client
  - Client for rendering
  - Server for CD and resources
- Also needs to be generated at both ends

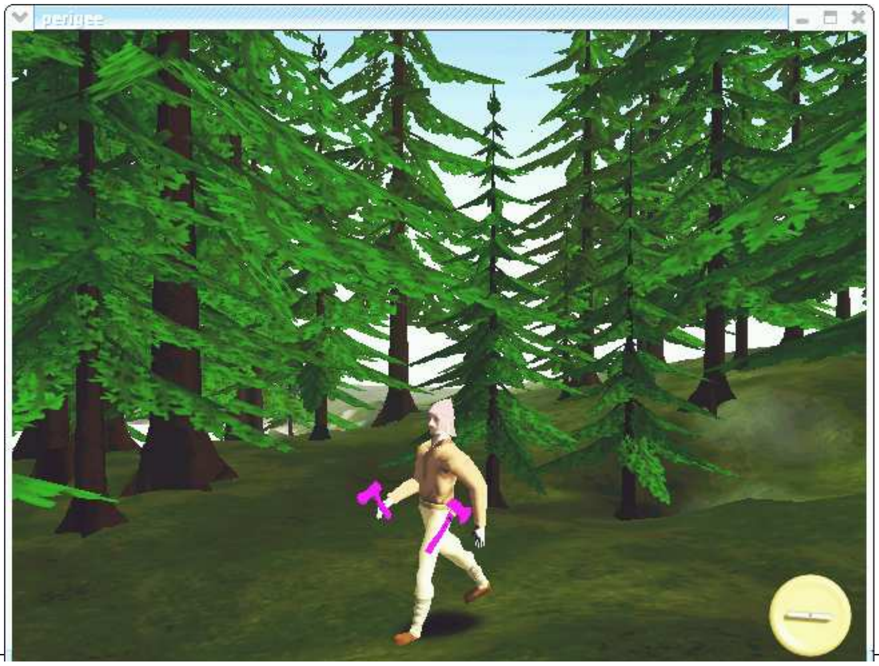
## Area based aproach

- Define areas which contain vegetation of a certain type
- Populate areas with instances procedurally
- Based on discrete grid points
  - Probability of tree at each point
  - Displacement and Orientation

## Key vegetation properties

- Repeatable, even if shape has changed
- Translocatable
- Must handle deleted instances
  - For a fully modifiable world

## Single tree type forest



## Further work

- Multiple species with different probabilities
- Grassland and scrub
- Clouds
- Rainfall, rivers and lakes