

Creating Spherical Worlds (sap_0251)

Kate Compton James Grieve Ed Goldman Ocean Quigley Christian Stratton Eric Todd Andrew Willmott*
Maxis, Electronic Arts

1 Introduction

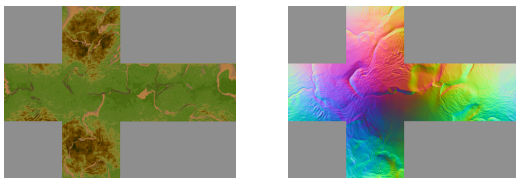
Our goal was to deliver procedurally created spherical planets for the game *Spore*, a considerable departure from our usual flat-world terrain approach. This presented three major problems: how to parameterize the planet's surface, how to create the surface itself, and how to texture the resulting geometry.

2 Parameterization

Some of the many possible approaches to parameterizing a deformed sphere include angle-based mappings, such as a longitude/latitude, gnomonic projections, and tiling the surface with charts, e.g., a regular tiling such as a dodecahedron. Two possible projections for such charts are planar (project into the tangential plane), and perspective (divide by the local z value).

Our choice had to minimise distortion as much as possible. We also had to consider the operations our system would need, such as heightfield storage, transforming from 3D locations to map coordinates and back, wrapping between map boundaries, efficient normal map generation from heightfield data, and splatting of rectangular areas into map space. Most of our existing gameplay and graphics approaches to terrain are grid-based, so we preferred the chart-based approach. In particular, perspective chart projections have the advantage that straight lines in such charts map to great circles on the sphere.

We chose a cubemap/projective parameterization as the best trade-off between distortion and our desired operations. The heightfield and other attribute maps are stored as six face grids. Further, by choosing face mappings to be permutations of the corresponding axes, it is possible to formulate efficient algorithms for wrapping between faces, and projecting a 3D point into a chart. Normal maps are generated in a similar way to the standard 2D texture-based approach, with the addition of a Jacobian term that takes into account spherical distortion.



3 Creating Heightfields

We create the heightfields that represent a planet's surface with a procedural brush system that operates on the sphere. Brushes are simple 2D textured rectangles that conditionally raise or lower the existing height values. Their footprint on the sphere is projected onto each face by culling to that face's frustum, and the GPU then splats the resulting textured triangles onto the face's heightfield; a pixel shader implements the brush operation.

We use our effects system, Swarm, to run these brushes across the planet surface, controlled by particle systems with various randomized parameter ranges. These systems can perform random walks, respond to terrain forces, and also hierarchically contain or spawn

other systems. The result is that artists can assemble a set of effects that represent planet features such as rivers, mesas, canyons, oceans, and plateaus.

4 Texturing

Texturing is accomplished similarly to other terrain-based games: detail and colour maps are combined according to a control texture. However in our case the control texture must be derived directly from the height field, rather than being pre-authored. It is generated by a combination of filtering options on the height field: we take the heightfield thresholded against water level, the gradient, and curvature, and combine them according to a technical-artist-supplied formula.

A given planet type can choose between various texture sets and adjustable parameters to the terrain shaders to produce planets with a certain thematic look. Planets also have atmosphere and temperature settings that control both colour ramps and parameters to our atmosphere and fogging model. Planets start particular values, but these can be modified later by terraforming gameplay.

We have developed an authoring system to allow many planet types to be assembled from terrain feature scripts by an artist. For any particular planet type, many different planet instances can be generated by varying the initial random seed.

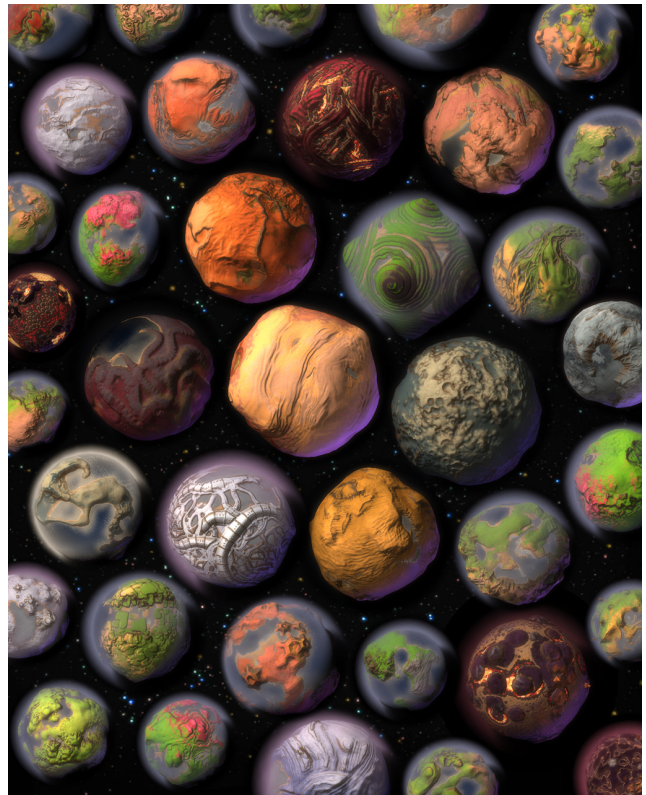


Figure 1: Some of the four billion planets possible with our system

*e-mail: awillmott@maxis.com