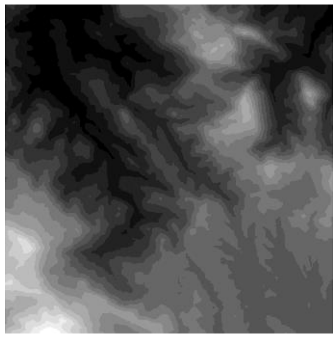
3D Bezier Terrain Editor

Terrain Generation

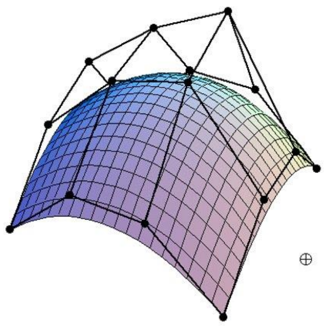
- Height Map Input



The terrain mesh is generated based on a grayscale height map texture. The width and heigh of the texture determines the world position of the further point. So, if a texture is 500x500 the world pos of the most distant point would be (500,500). Moreover, the minimum height is zero and the maximum is ¼ of the width of the texture. Heigh map is assumed to be square. The height is used for the position of the control points.

- Bezier Patch Representation:

The terrain is divided into multiple Bezier patches, allowing smooth surface generation and easy modification. The Bezier Patches are represented by this formula: $P(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} B_i^n(u)B_j^m(v) \cdot k_{i,j}$, where $k_{i,j}$ is a control point. The reason of using the Bernstein algorithm is that we will always have cubic Bezier patches, therefore the binomial coefficients can be precomputed in order to make the patch generation faster.



- Adaptive Resolution:

The resolution of the patches is adjusted based on the distance between the camera and the patch itself. The center of the patch is taken as the location for this one, this position is computed averaging the four corner points of the patch. A drawback of this adaptative resolution is that the hole terrain must be recomputed every time the camera is moving. Another approach that would be better in terms of performance would be to make the patches with more complex terrain features denser and keep the flatter patches with less destiny. This approach would not be recomputed its frame but just when a patch is changed.

Bezier Patch Control for Smoothness

- Editable Control Points:

Except the Corner control points, that are not adjustable to maintain the shape of the initial height-map, the other control points of the patches are adjustable in the Y coordinate, this prevents having cave like terrain which cannot be represented in a height-map.

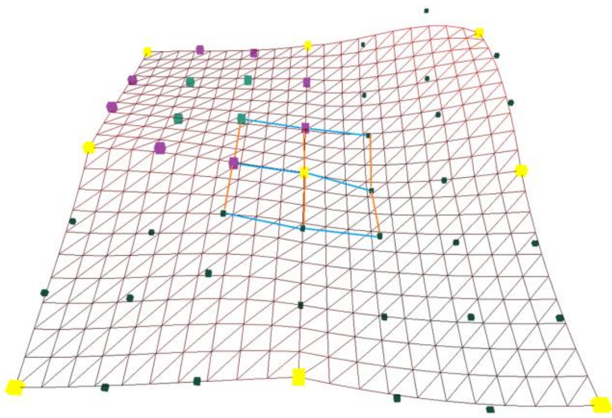
- Continuity Preservation:

The corner and edge Control Points are shared between the adjacent patches and when one of these points is edited, the same control point from the contiguous patch is also updated to ensure continuity in the terrain.

- Subdivision Strategy:

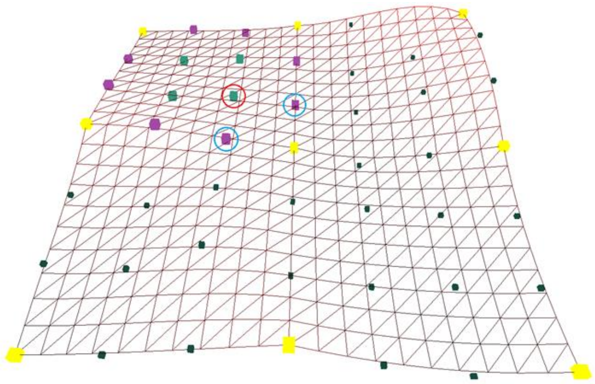
Terrain has a variable in the editor called patches count that update on runtime as the user changes that variable. If the variable gets modified the terrain gets computed again with the new desired precision. The more patches the more precise the terrain will be to the actual height map.

- Smoothness:

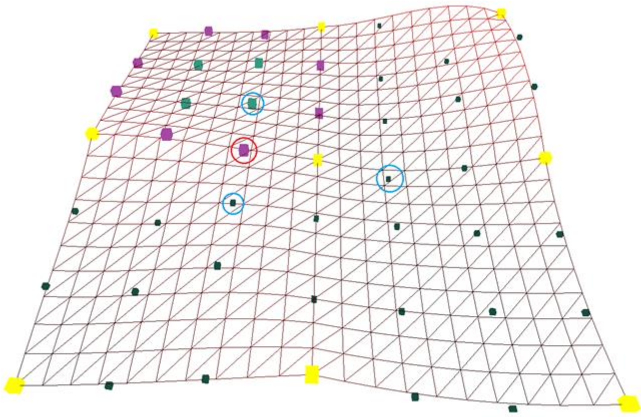


We had a couple of problems with the smoothness of the terrain. The first problem was that to ensure that no sharp edges were to be appeared when editing one control point, another 8 control points were needed to be altered, which in our opinion was altering too much of the terrain and it could be very tedious for the user. The second problem was that there where infinite amount of solutions to have a perfectly smooth surface. Therefore, we decided to take the following approach:

  - When modifying the blue control point, just modify the contiguous control points.



- When modifying the purple control point, modify the adjacent control points by the same delta and make the purple point of the next patches colinear



Editor GUI for Terrain Modification

- Control Point Editing:

By clicking you can select the patch of the terrain to reshape, once that the patch is selected, the control points inside that patch will get bigger in order to select the actual control point that the user wants to modify. Having the actual control point selected, a gizmo will appear letting the user free movement and the terrain will update based on the new control point position. River follows the same logic of selecting the desired control point and modifying its position with a gizmo updating the river's shape.

- Height Adjustment Tools:

Moving a control point inside a patch directly affects the topology and the height, not only of the selected patch but also from its neighbors, in case of using smoothing it will have even greater impact on them smoothing the surface whereas sharp whereas sharp edges just affect the border control points of the neighbors. This tool is essential to adjust the terrain as desired and achieve both sharp changes or smoother ones depending on the user and the goal.

- Visual Feedback:

The terrain updates with every change in real-time showing the current state of the terrain with its modifications. The color of the mess changes with the topology. Points that have height less or equal zero are black while the one with height higher or equal width/4 are red.

Road and River Integration

- Path Definition:

By using shift + left click, if you click on a patch, it selects the first control point of the road/river at the middle of the patch. Then if you shift + left click again in another patch, it will create the last control points. The middle ones will be created in the same line as the line created by the starting and ending points by default.

- Surface Fitting:

For the road/river to be projected onto the terrain what we did was, first we computed a 2D Bezier curve without a height and its normal vector of each point. Then, we created two curves using the normal and a thickness to compute the two curves that will form the road/river. After computing those curves, for every point of the curve, check within which patch is in and once it is found, we use the position of the point as local coordinates to compute the Bezier patch. Lastly, we triangulate the path to create the mesh.