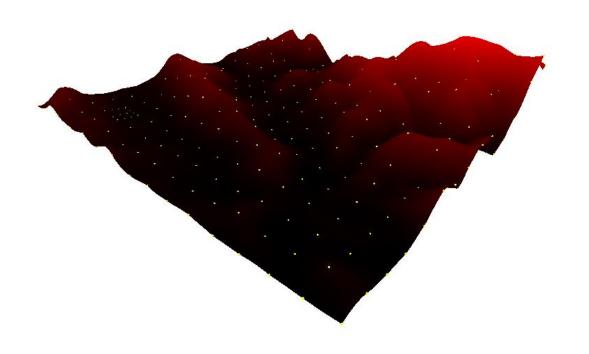
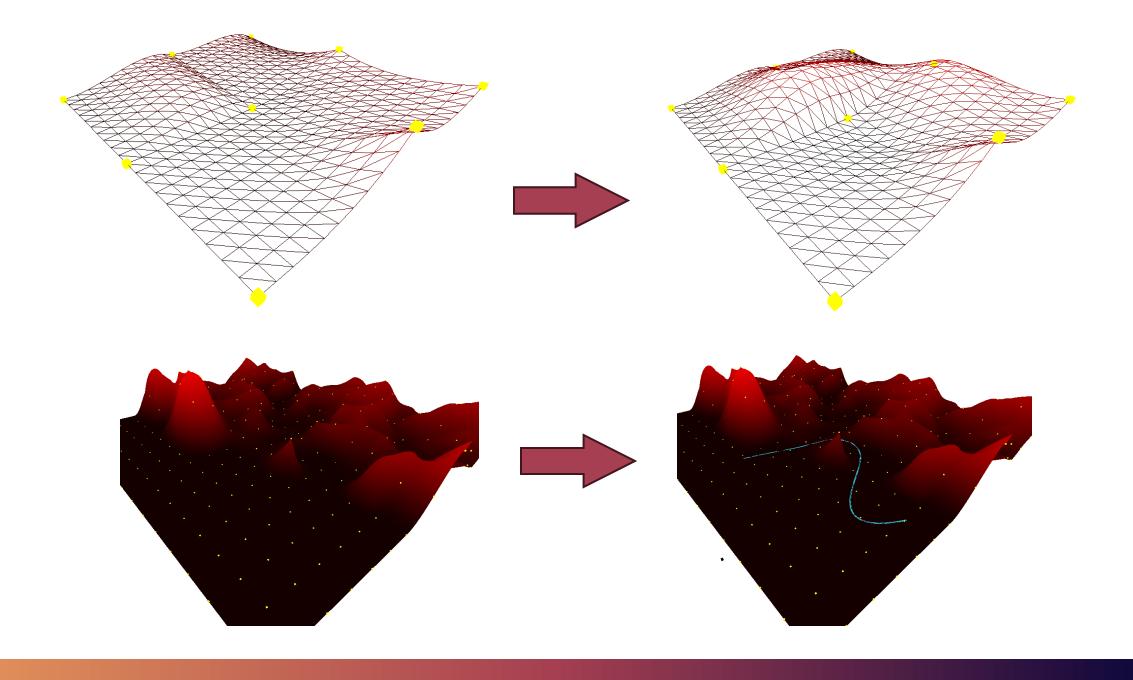
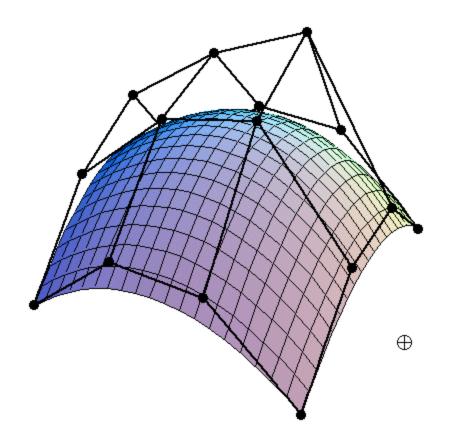
# MAT 300 – TERRAIN GENERATION WITH BEZIER PATCHES



# INTRODUCTION TO THE PROJECT

- TERRAIN MESH GENERATION
   USING BEZIER PATCHES
- HEIGHT MAP GENERATION WITH ADJUSTABLE SMOOTHNESS
- POSSIBILITY OF MANUAL EDITING
   AND ADDITION OF RIVERS



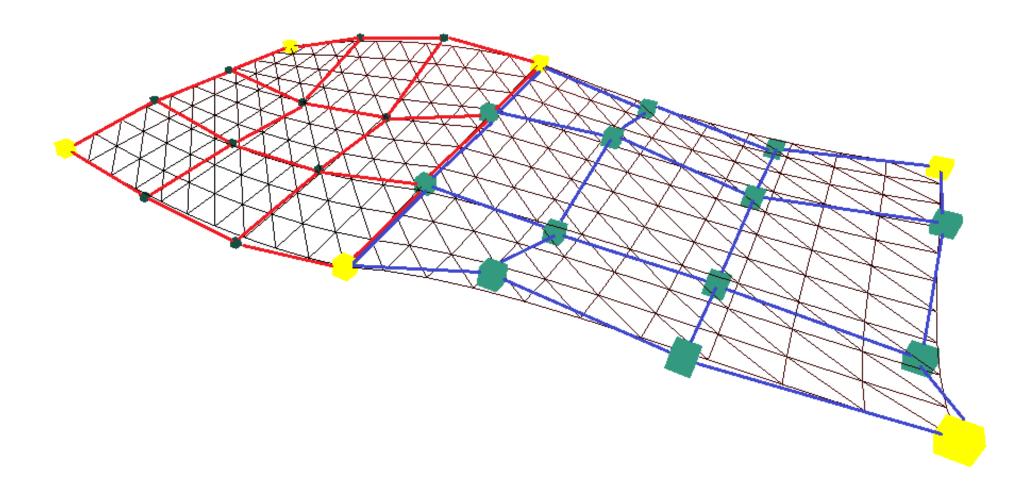


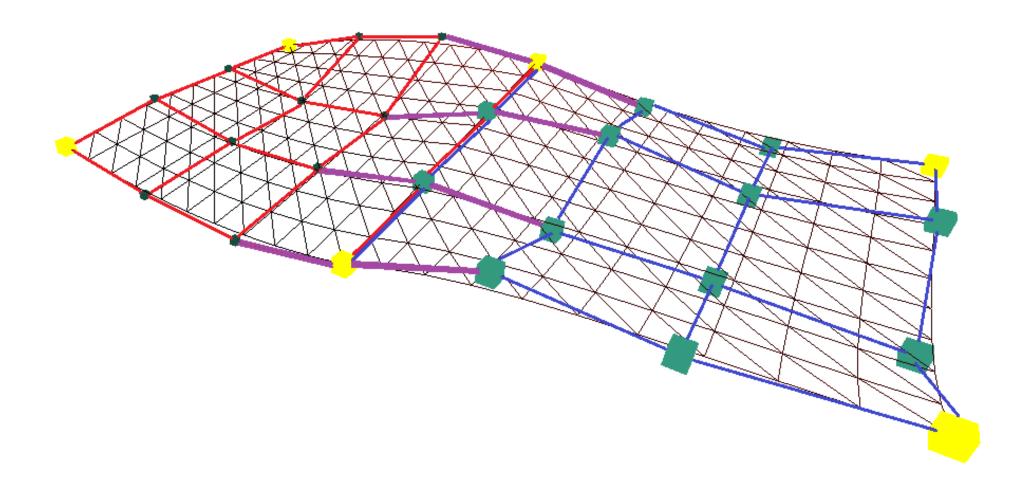
### MATHEMATICAL FOUNDATION

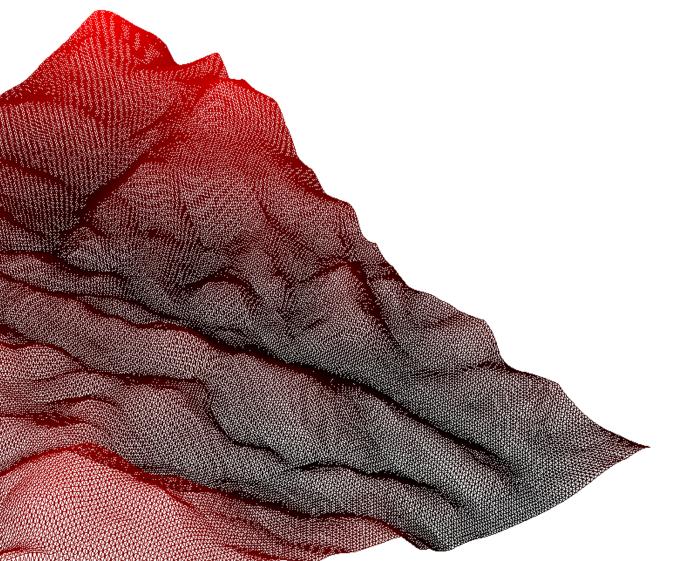
- WE USE CUBIC BEZIER PATCHES TO CREATE THE TERRAIN MESH
- $P(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i}^{n}(u) B_{j}^{m}(v) * k_{i,j}$ WHERE  $k_{i,j}$  IS A CONTROL POINT
- SINCE WE USE CUBIC BEZIER
   PATCHES WE USE BERNSTEIN, SINCE
   WE CAN ALREADY HAVE THE
   BINOMIAL COEFFICIENTS
   CALCULATED

# SMOOTHNESS IN A BEZIER PATCH

SAME AS IN A BEZIER CURVE, IN A
BEZIER PATCH THE CONTROL POINTS
OF TWO PATCHES NEED TO BE
COLINEAR ON THE EDGES







#### OUR FRAMEWORK

- C++
- OPENGL
- GLM-GLFW
- IMGUI/GUIZMO



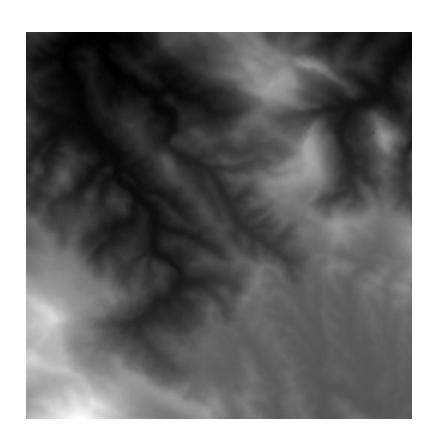
#### OUR SYSTEM

- SIMPLE GRAPHICS PIPELINE
- TERRAIN SYSTEM
- EDITOR



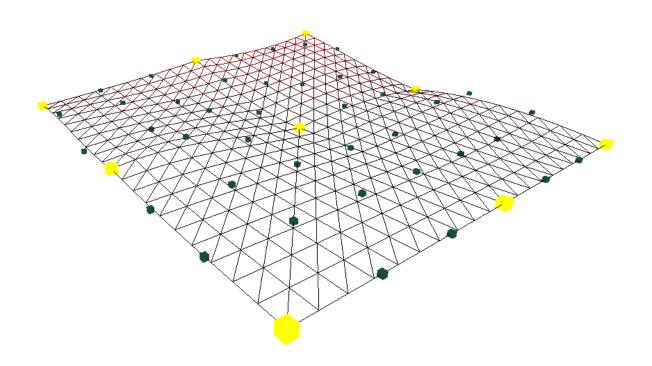
#### FEATURES

- CONTROL POINTS EDITOR
- SMOOTH/ROUGH EDITING
- CAMERA DISTANCE BASED
   PATCH RESOLUTION



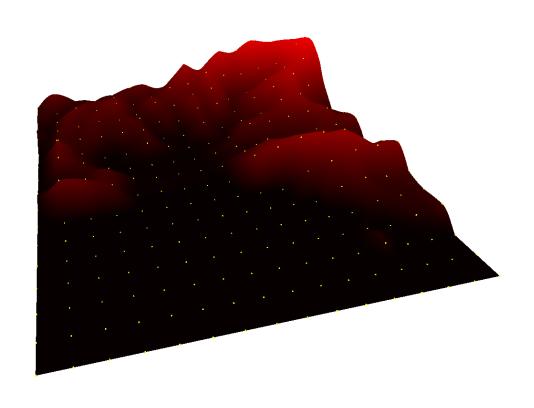
#### FEATURES

- 3D BEZIER PATCHES
- RIVER 2D BEZIER CURVE



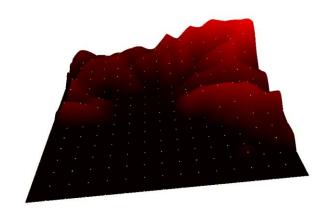
1- Rendering framework, we focused on creating a framework where we could test our advances

2-Create Bezier patches.

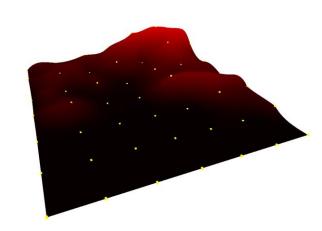


3- Load height maps and create Bezier patches based on them, still no editable mesh

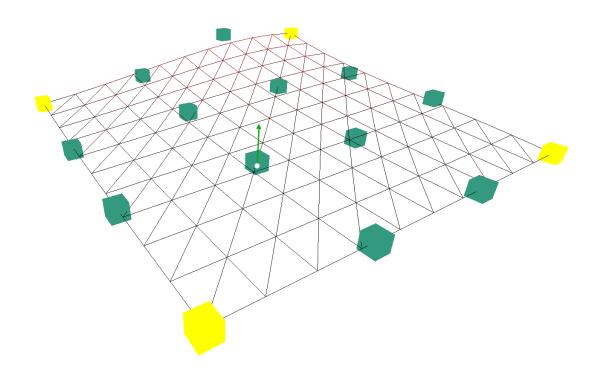






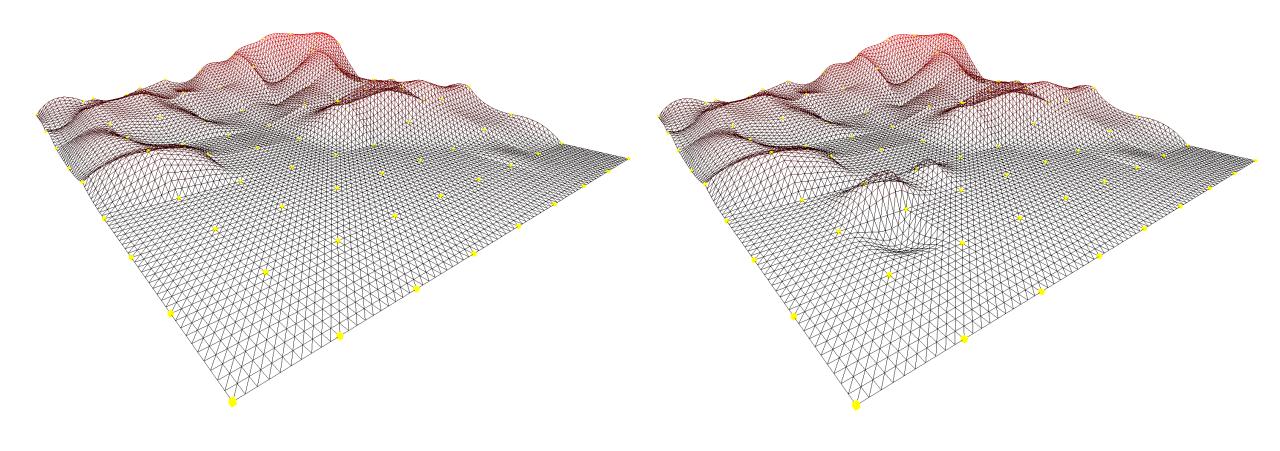


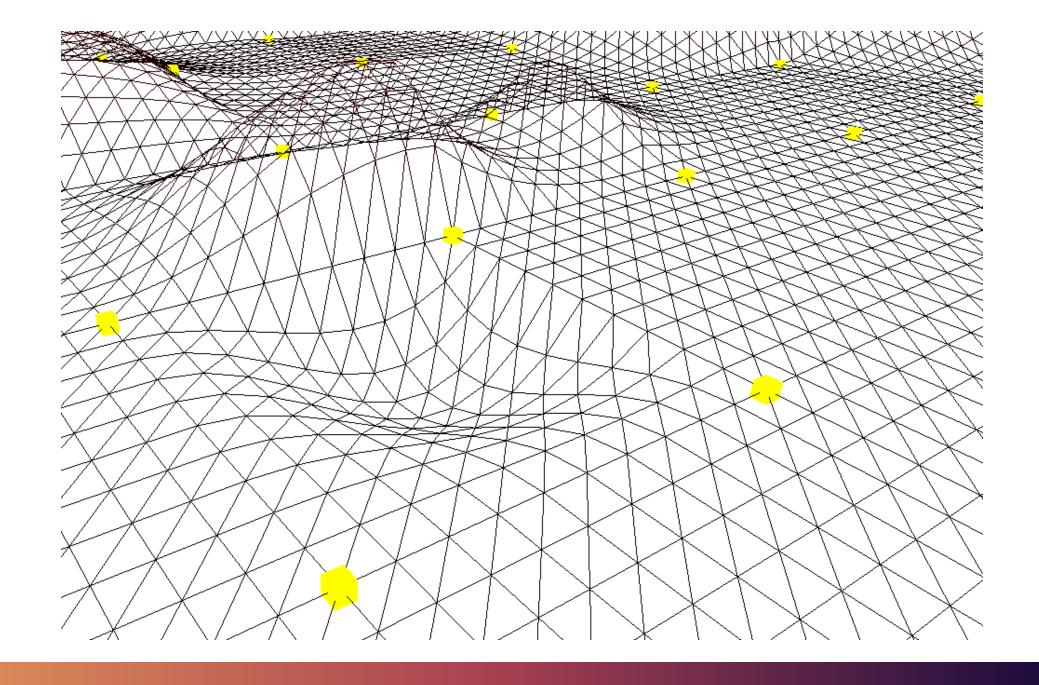
4-Add ImGui to edit basic things such us amount of patches

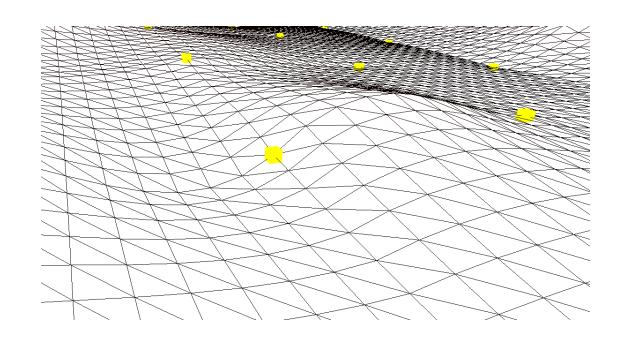


6-Make selectable control points

7- Move the control points without the smoothing option, just recalculate the mesh







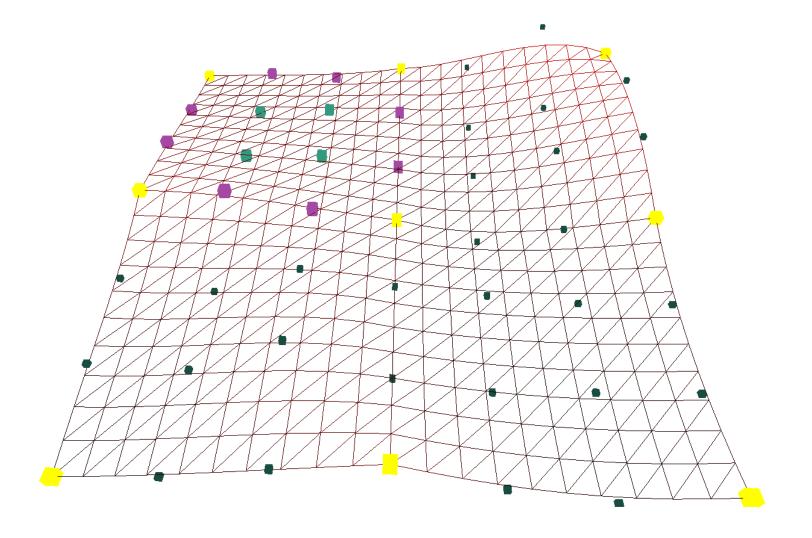
8- Smoothing

Three types of Control points:

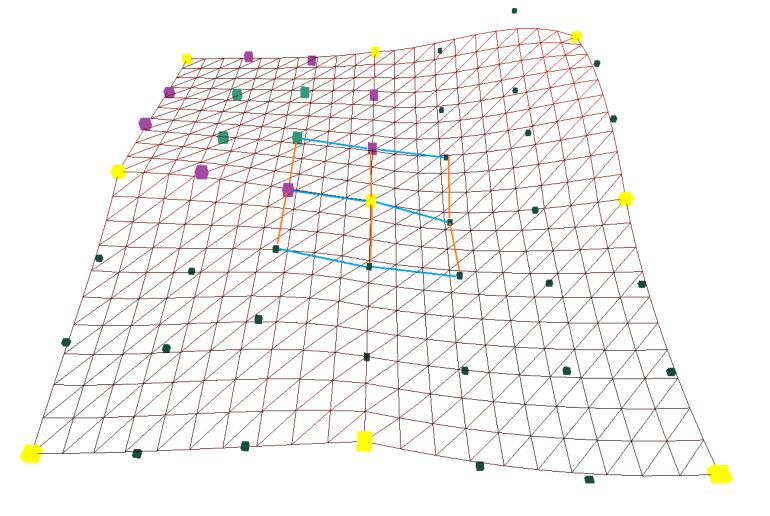
1. Yellow: Shared by 4 patches

2. Purple: Shared by 2 patches

3. Blue: Part of just 1 patch

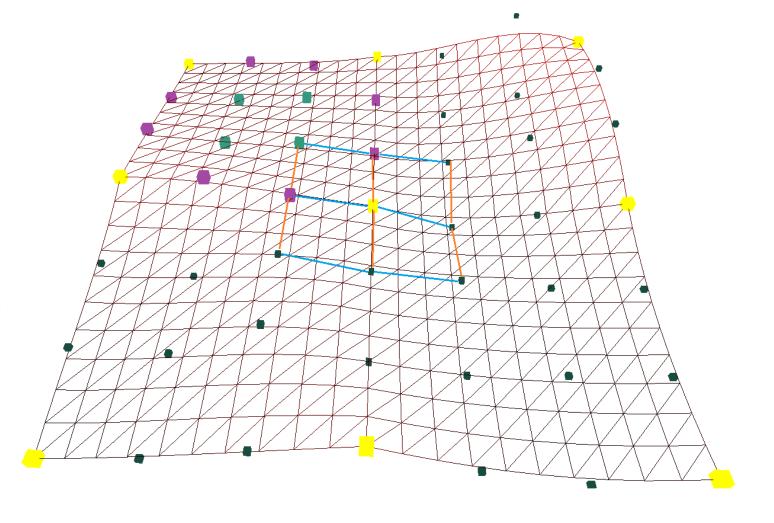


For a perfectly smooth terrain we would need for all these control points to be smooth:



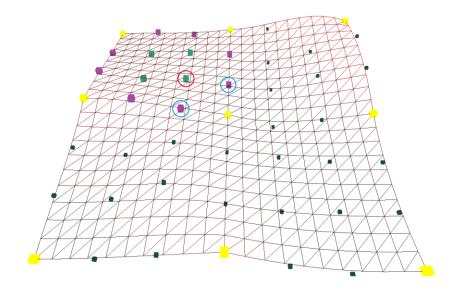
### Two problems:

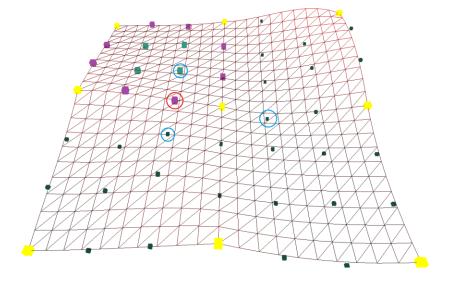
- Too much of the terrain is modified when one control point is changed, not our intention
- 2. Infinite amount of solutions

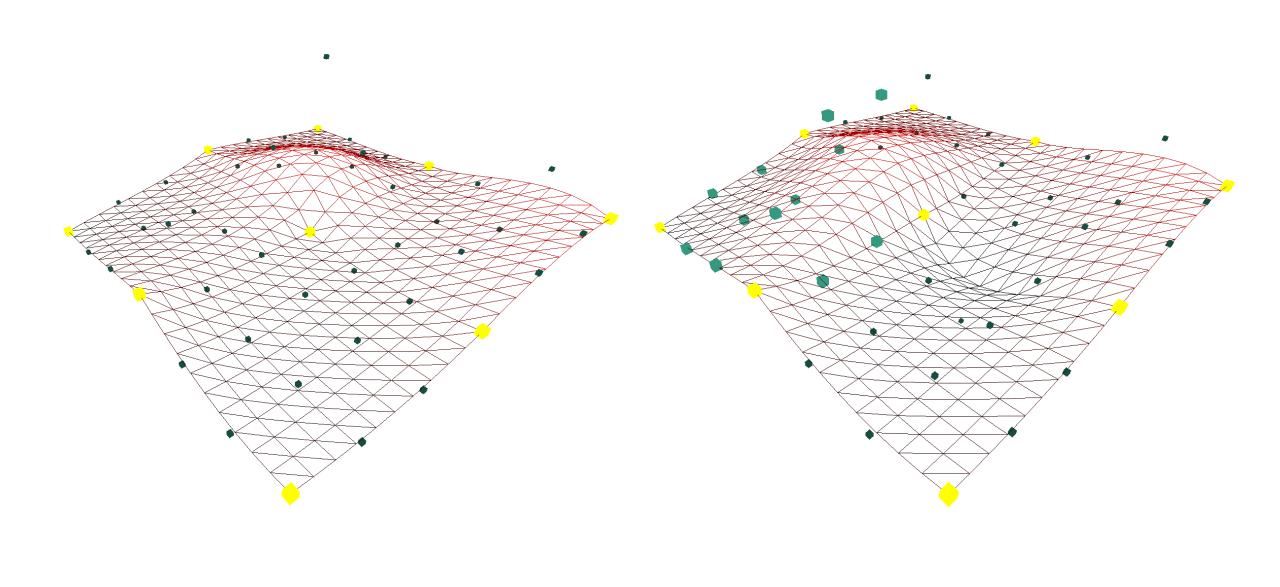


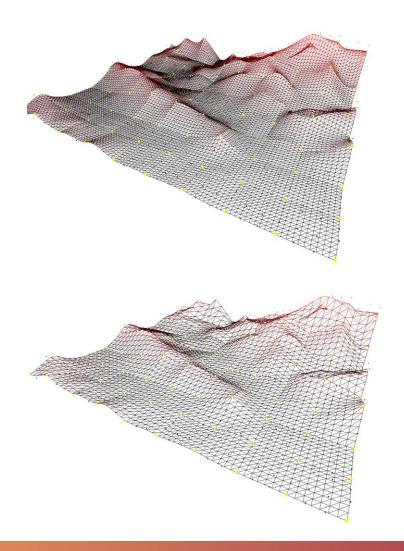
- Our solution:
- 1. When modifying the blue control point, just modify the contiguous control points
- 2. When modifying the purple control point modify the adjacent control points by the same delta and make the purple point of the next patches colinear

- Problem:
- Not fully smooth terrain

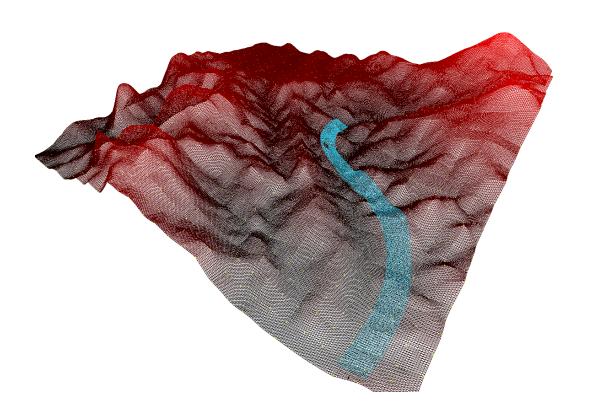






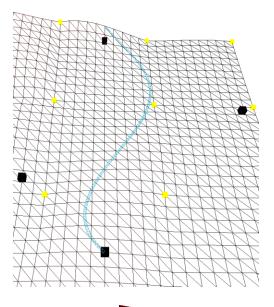


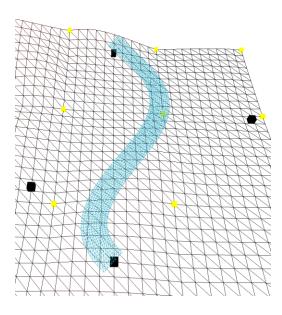
9- Optimizations: dt camera distance dependent

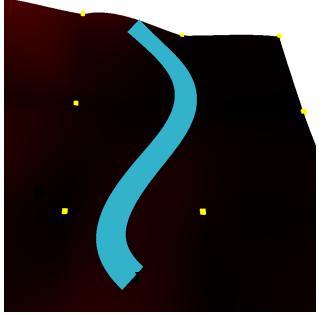


10- River

- Steps to create the river:
- 1. Create a 2D cubic Bezier curve in the XZ axis.
- 2. Offset the Bezier curve to the right and left to give the river some width.
- 3. Project the points in the mesh.
- 4. Triangulate the points.
- 5. Render.

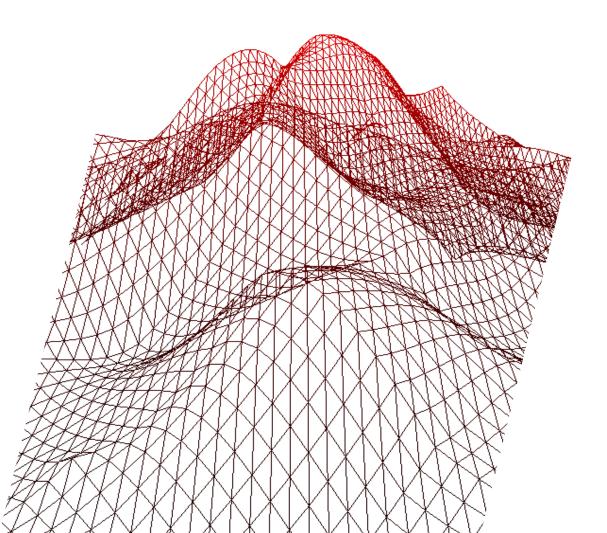






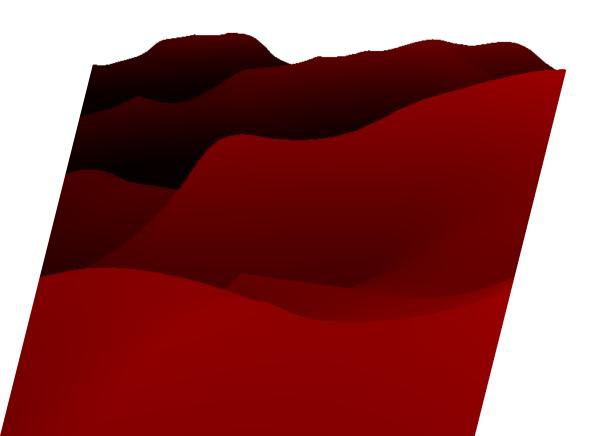
Ander

### DEMO



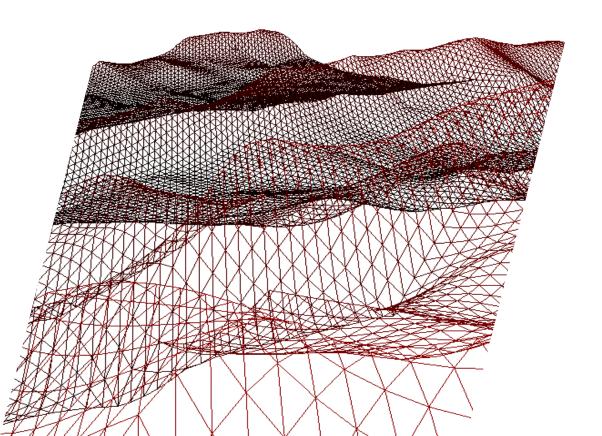
#### WHAT WE HAVE LEARNED

- HOW TO USE AND APPLY 2D AND 3D
   BEZIER CURVES
- SMOOTHING TENDS TO INFINITY
- TERRAIN NOT EXACTLY REAL



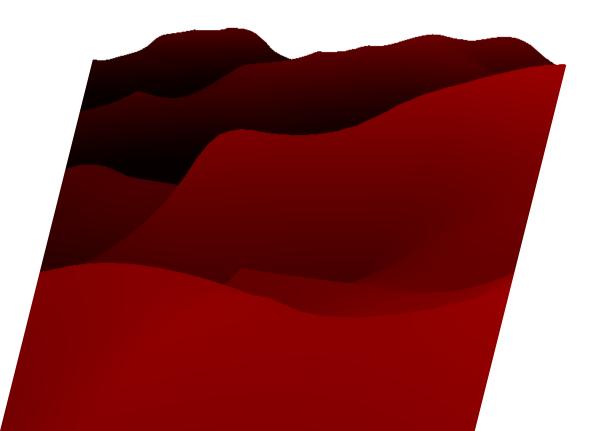
#### WHAT WOULD WE CHANGE

- HOW SMOOTHING WORKS
  - AFFECT MORE NEIGHBORING
     CONTROL POINTS



#### WHAT WOULD WE CHANGE

PATCH RESOLUTION DEPENDENT
 ON ITS COMPLEXITY



#### WHAT WOULD WE ADD

- LIGHT + SHADOWS
  - O NORMAL COMPUTATION

### THANKS

