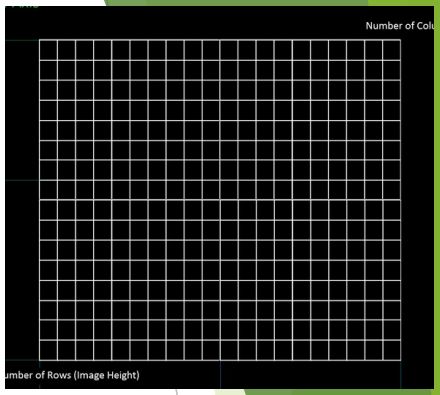
Midterm Project
Rasterization -Line

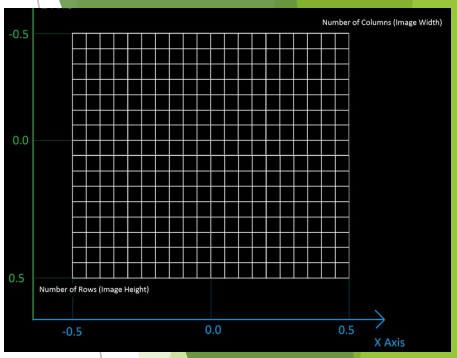
(Previous Lab)
Use the 2D grid you created

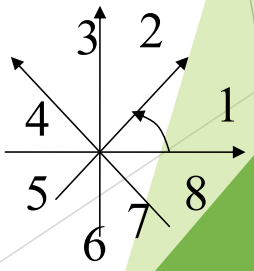


- Clickable 2D Grid
  - Provide a popup menu to select the grid dimensions: (10 or 15 or 20 etc…)
  - Draw a 2D grid based on the selected dimension.
    - The default is 10 → x: (-10 ~ 10), y: (-10 ~ 10)
    - The origin (0,0) is at center
    - When the user select 15, the grid will be re-drawn to: x: (-15 ~ 15), y: (-15 ~ 15)
  - When the user click on one of the cell
    - draw/fill the cell
      - You will need to implement a function to convert coordinates
      - Print out the coordinate (x, y) of this cell on the console window

(Previous Lab)
Draw Line: Midpoint algorithm

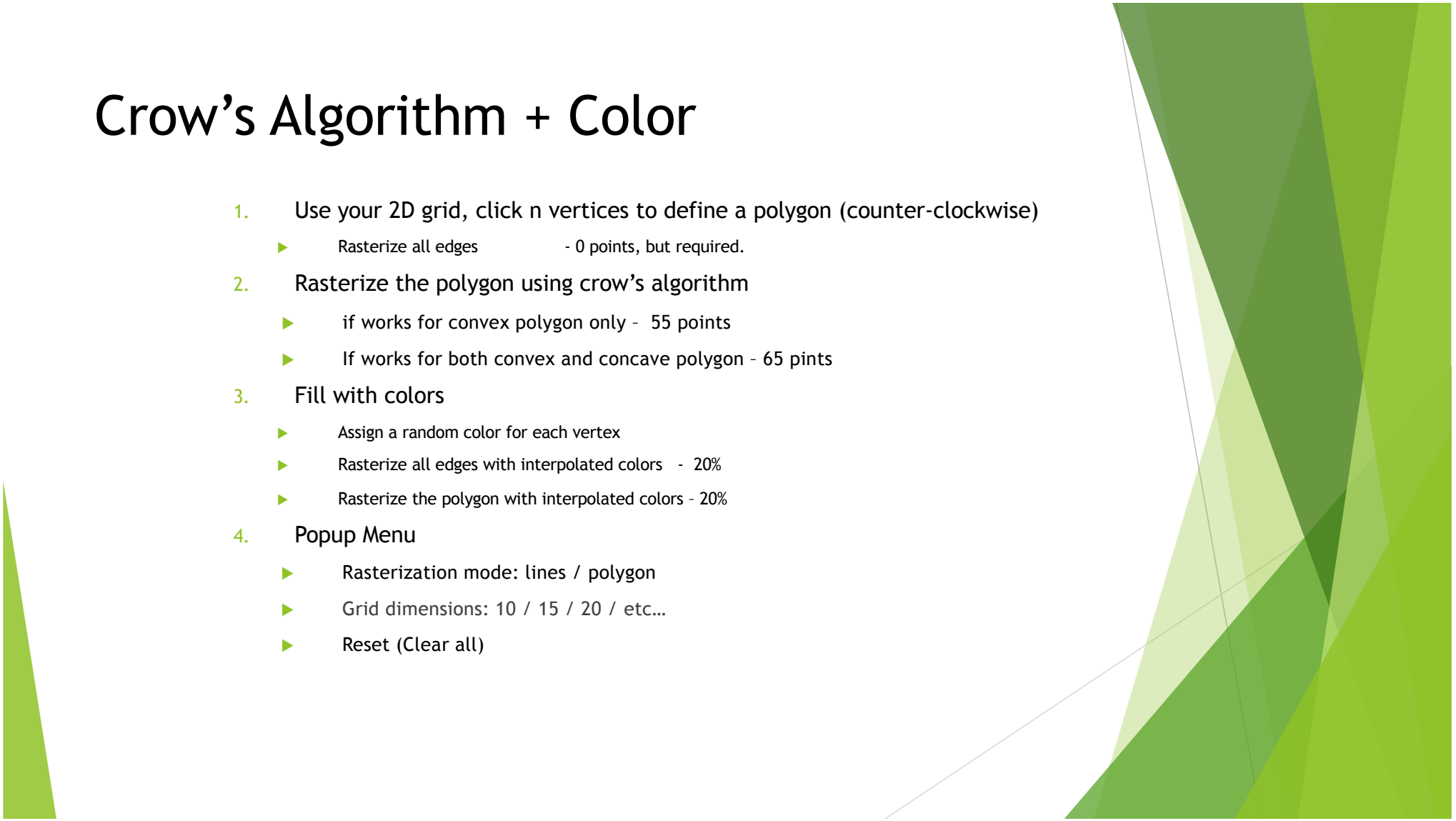
- Select two endpoints
- Use midpoint algorithm to draw the pixels along the line
  - Draw and print out all the pixels represent the line
  - Print out the coordinate (x, y) OF EACH PIXELS
  - Considering all regions
    - (First 2 region for 30%, the rest regions total 20%)
- anti-aliasing algorithm
  - A popup menu to switch between midpoint/anti-aliasing algorithm





Crow's Algorithm + Color

1. Use your 2D grid, click n vertices to define a polygon (counter-clockwise)
   - Rasterize all edges - 0 points, but required.
2. Rasterize the polygon using crow's algorithm
   - if works for convex polygon only - 55 points
   - If works for both convex and concave polygon - 65 pints
3. Fill with colors
   - Assign a random color for each vertex
   - Rasterize all edges with interpolated colors - 20%
   - Rasterize the polygon with interpolated colors - 20%
4. Popup Menu
   - Rasterization mode: lines / polygon
   - Grid dimensions: 10 / 15 / 20 / etc…
   - Reset (Clear all)



Requirement

- Do not use other libraries. Only OpenGL API (gl, glu, glut) is allowed
- Write comments in your code
- Turn in your code, document and demo video.
  - You should explain how to use your program with some screen shots in your document.
- Due date: 5/21 midnight (11:00pm)