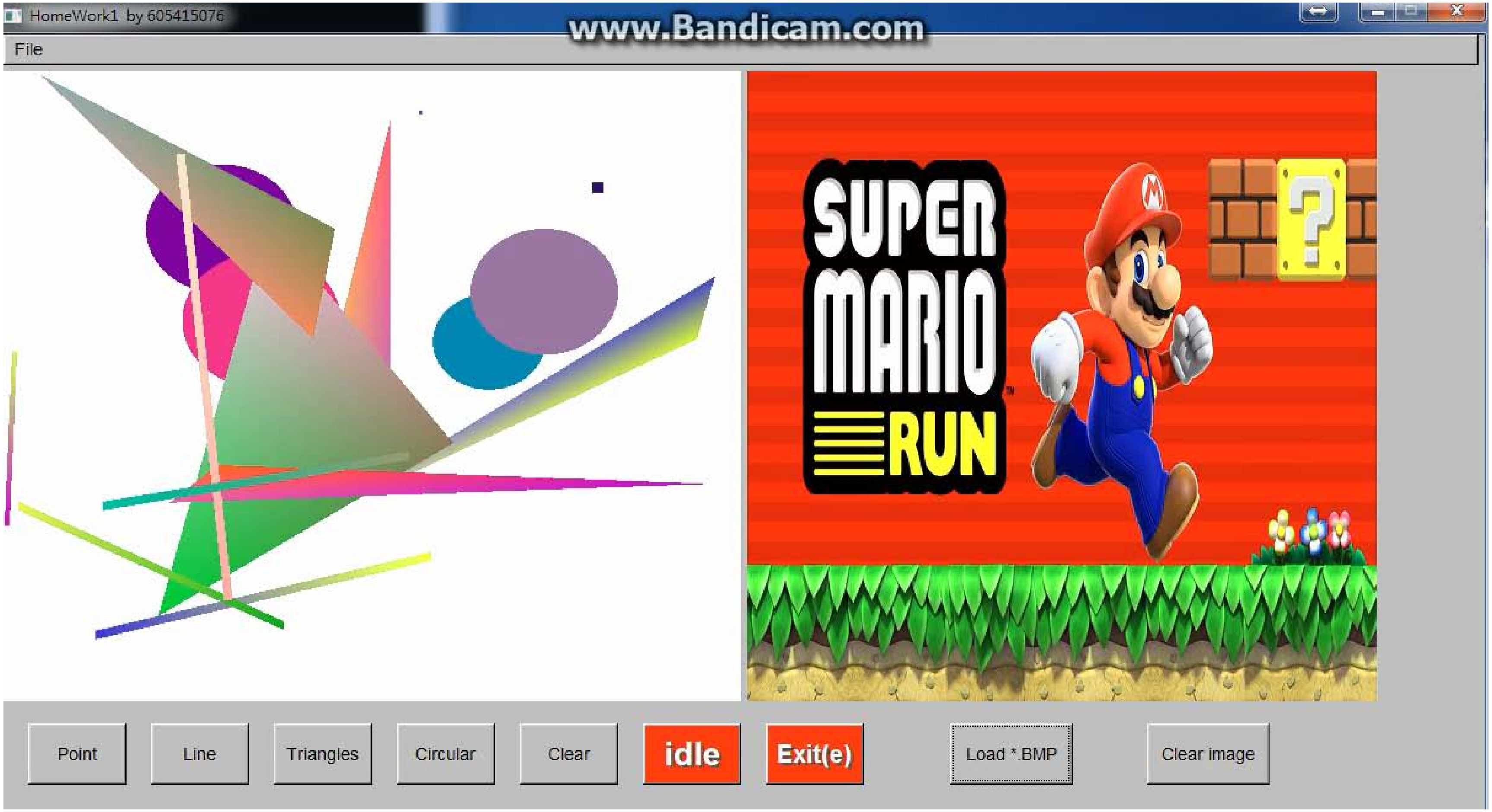
Laboratory #1 (2D Drawing)

- The purpose of this assignment is to familiarize you with the basic OpenGL drawing commands.
- You will implement a program that mimics an old screen saver by filling the screen with random two-dimensional geometry.

- The program will start out with a blank screen. The user interacts with the program by pressing buttons to change the behavior of the program.



Movie in action

Laboratory #1 (2)

- The program has four modes: idle, points, lines, and triangles. The behavior in each mode is as follows.

- In idle mode (will restart the program), the program displays a blank screen. Idle mode is set by pressing the button called idle.

Laboratory #1 (3)

- In points mode, the program displays points on the screen. Each point should have a random location (within the program’s window) and a random color. Points mode is set by pressing the button called points.
- In lines mode, the program displays line segments on the screen. The endpoints of the lines should be placed and colored randomly. Lines mode is set by pressing the button called lines.

Laboratory #1 (4)

- In triangles mode, the program display triangles on the screen. The vertices of the triangles should be placed and colored randomly. Triangles mode is set by pressing the button called triangles.
- Except during the idle mode, new shapes should be added to the screen on each frame. That is, every time the renderFunc() is called, your program should add another point, line, or triangle to the visible image.

Laboratory #1 (5)

- After enough shapes have been drawn to fill the window, the screen should be cleared; drawing should continue. The screen should also be cleared whenever the mode is changed.

- The program should terminate when the ‘Quit’ button is pressed.

Laboratory #1 (6)

- **Remember that it clears the screen between each call to renderFunc(). Therefore, you have to draw all the geometry you want to display each time this function is called – the screen will not remember what you have previously drawn. You might store previously generated vertices in a list so they can all be drawn for each frame.**

Laboratory #1 (7)

- To get random numbers, you can call the C library function rand(), which returns an integer between 0 and RAND_MAX.

- In a separate window, we also provide is: supermario.bmp and READ_BMP.h
- Please use the void drawPoint(x, y, r, g, b) function to color your pixels; x and y are ints while r, g, b are floats.