Resizing the Window

Lecture 6

Robb T. Koether

Hampden-Sydney College

Fri, Sep 6, 2019

Outline

Outline

Resizing the Window

- When the window is resized by the user, the framebuffer size (or window size) callback function is automatically called.
- It is the responsibility of the framebuffer size callback function to create a projection matrix, based on the new dimensions of the window, and to pass it to the vertex shader.

The `ortho2D()` Function

The `ortho2D()` Function

```
mat4 ortho2D(int xmin, int xmax, int ymin, int ymax);
```

- The `ortho2D()` function will create the projection matrix if we provide it with the left, right, bottom, and top boundaries (in world coordinates) of the window, which we are naming `xmin`, `xmax`, `ymin`, and `ymax`.

Outline

The FramebufferSize Callback Function

The FramebufferSize Callback Function

```
void framebufferSizeCB(GLFWwindow* window, int width,
     int height)
{
//   Use width and height to compute the new
//   window boundaries

     xmin = ···
     xmax = ···
     ymin = ···
     ymax = ···
          ⋮
     mat4 proj = ortho2D(xmin, xmax, ymin, ymax);
     glUniformMatrix4fv(proj_loc, 1, GL_TRUE, proj);
          ⋮
}
```

- We will use the global variables `xmin`, `xmax`, `ymin`, and `ymax` for

Outline

The `setProj()` Function

The `setProj()` Function

```
void setProj()
{
    proj = ortho2D(xmin, xmax, ymin, ymax);
    glUniformMatrix4fv(proj_loc, 1, GL_TRUE, proj);
}
```

- It is handy to place into a separate function the statements that create and transfer the projection matrix.
- It is ok to make `proj`, `proj_loc`, `xmin`, `xmax`, `ymin`, and `ymax` global variables.

The `setProj()` Function

The `setProj()` Function

```
void setProj()
{
    proj = ortho2D(xmin, xmax, ymin, ymax);
    glUniformMatrix4fv(proj_loc, 1, GL_TRUE, proj);
}
```

- It is handy to place into a separate function the statements that create and transfer the projection matrix.
- It is ok to make `proj`, `proj_loc`, `xmin`, `xmax`, `ymin`, and `ymax` global variables.
- Why is it ok?

The FramebufferSize Callback Functionndow

- The question is, how to recompute *xmin*, *xmax*, *ymin*, and *ymax* when the window is resized?

The FramebufferSize Callback Functionndow

- The question is, how to recompute *xmin*, *xmax*, *ymin*, and *ymax* when the window is resized?
- Choose a point that is to be fixed.
  - Upper-left corner
  - Lower-left corner
  - Center
  - Etc.

Outline

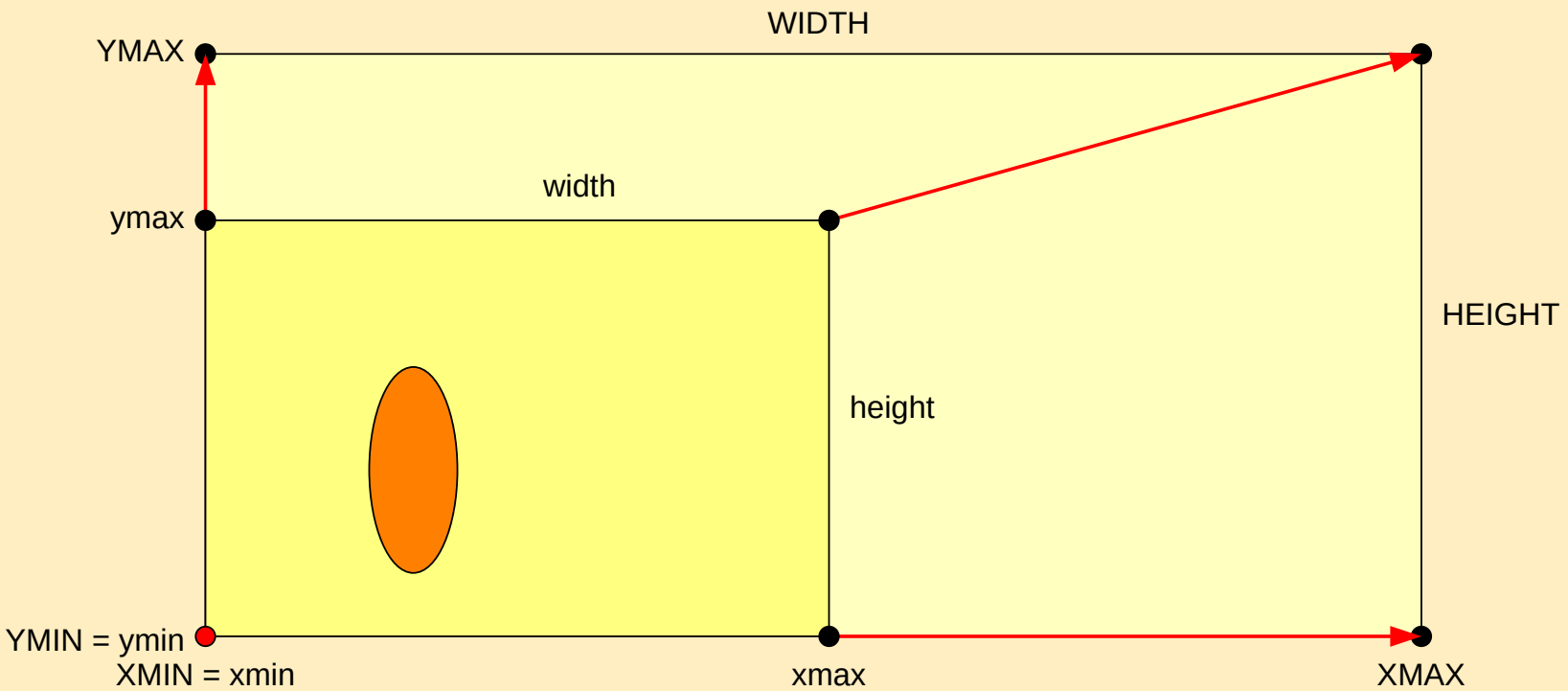
Fixing a Point

- We will label the new values *XMIN*, *XMAX*, *YMIN*, and *YMAX*.
- Suppose we keep the lower-left corner fixed.
- If the window is expanded, then the expansion will reveal more of the scene to the right and above.

Outline

Lower-Left Corner Fixed

Lower-Left Corner Fixed



- In the diagram, xmin, xmax, ymin, ymax, width, and height are the "old" values and XMIN, XMAX, YMIN, YMAX, WIDTH, and HEIGHT are the "new" values.

Lower-Left Corner Fixed

- *xmin*, *xmax*, *ymin*, and *ymax* are in world coordinates.
- *width* and *height* are in screen coordinates.
- Be careful.

Lower-Left Corner Fixed

- *xmin*, *xmax*, *ymin*, and *ymax* are in world coordinates.
- *width* and *height* are in screen coordinates.
- Be careful.
- Be Careful!

Lower-Left Corner Fixed

- *xmin*, *xmax*, *ymin*, and *ymax* are in world coordinates.
- *width* and *height* are in screen coordinates.
- Be careful.
- Be Careful!
- BE CAREFUL!!!

Lower-Left Corner Fixed

- Clearly, $XMIN = xmin$ and $YMIN = ymin$.
- Also, we have the relation

$$\frac{YMAX - XMIN}{xmax - xmin} = \frac{WIDTH}{width}.$$

Lower-Left Corner Fixed

- Clearly, $XMIN = xmin$ and $YMIN = ymin$.
- Also, we have the relation

$$\frac{YMAX - XMIN}{xmax - xmin} = \frac{WIDTH}{width}.$$

- Therefore,

$$YMAX = XMIN + \left(\frac{WIDTH}{width}\right)(xmax - xmin).$$

Lower-Left Corner Fixed

- Clearly, $XMIN = xmin$ and $YMIN = ymin$.
- Also, we have the relation

$$\frac{YMAX - XMIN}{xmax - xmin} = \frac{WIDTH}{width}.$$

- Therefore,

$$YMAX = XMIN + \left(\frac{WIDTH}{width}\right)(xmax - xmin).$$

- Similarly,

$$YMAX = YMIN + \left(\frac{HEIGHT}{height}\right)(ymax - ymin).$$

The framebufferSizeCB() Function

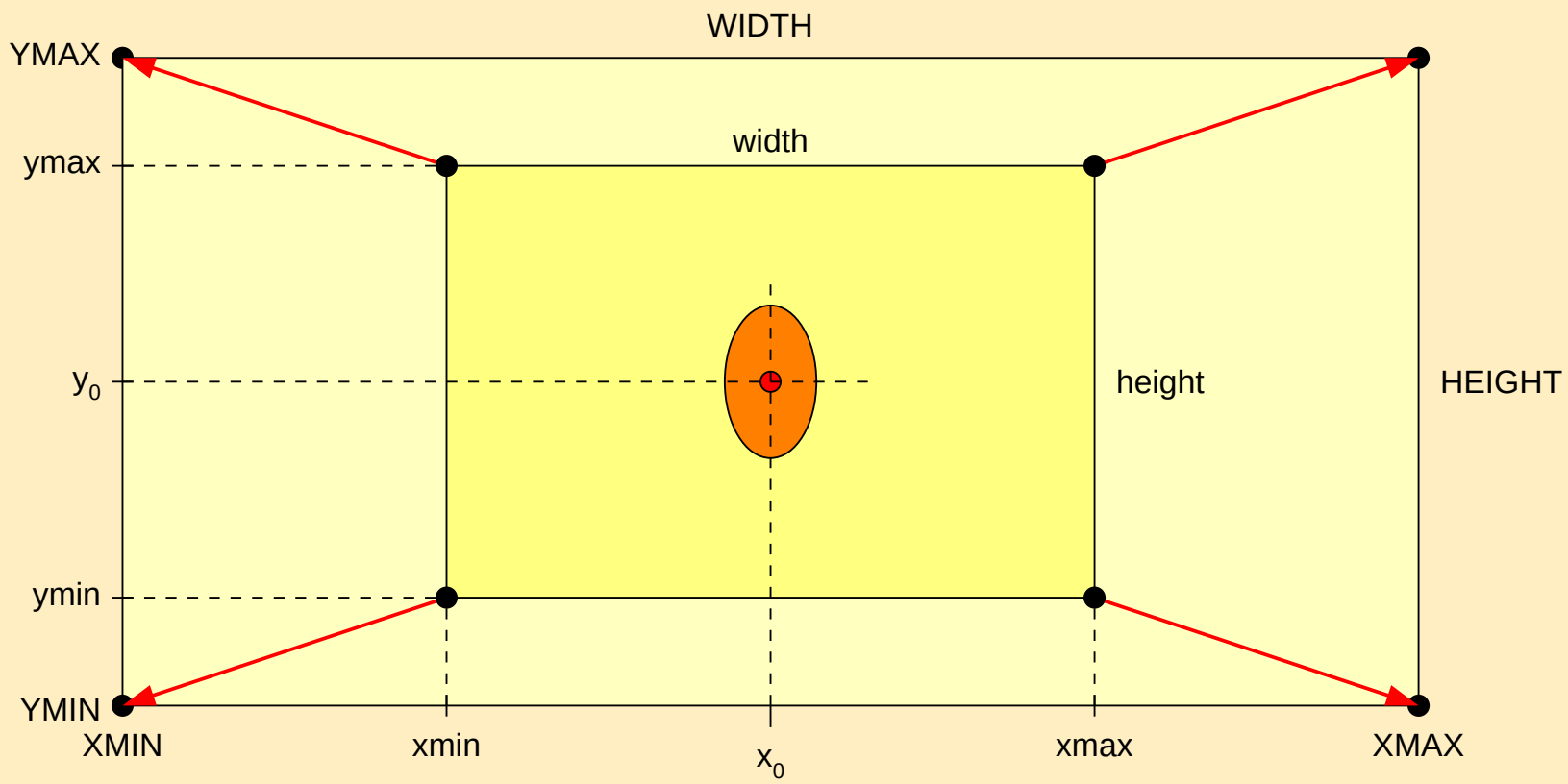
The framebufferSizeCB() Function

```
void framebufferSizeCB(int width, int height)
{
//  Compute new window boundaries

    xmax = xmin + (float)width/fb_width*(xmax - xmin);
    ymax = ymin + (float)height/fb_height*(ymax - ymin);

    setProj();
        ⋮
}
```

Outline

Center Fixed

Center Fixed



- We may keep the center fixed.

The framebufferSizeCB() Function

The framebufferSizeCB() Function

```
void framebufferSizeCB(int width, int height)
{
//  Compute new window boundaries

    xmin = ···
    xmax = 0.5*((xmax  + xmin)
           + (float)(width/fb_width)*(xmax - xmin));
    ymin = ···
    ymax = ···

    setProj();
        ⋮
}
```

- This may produce the wrong results. How so?

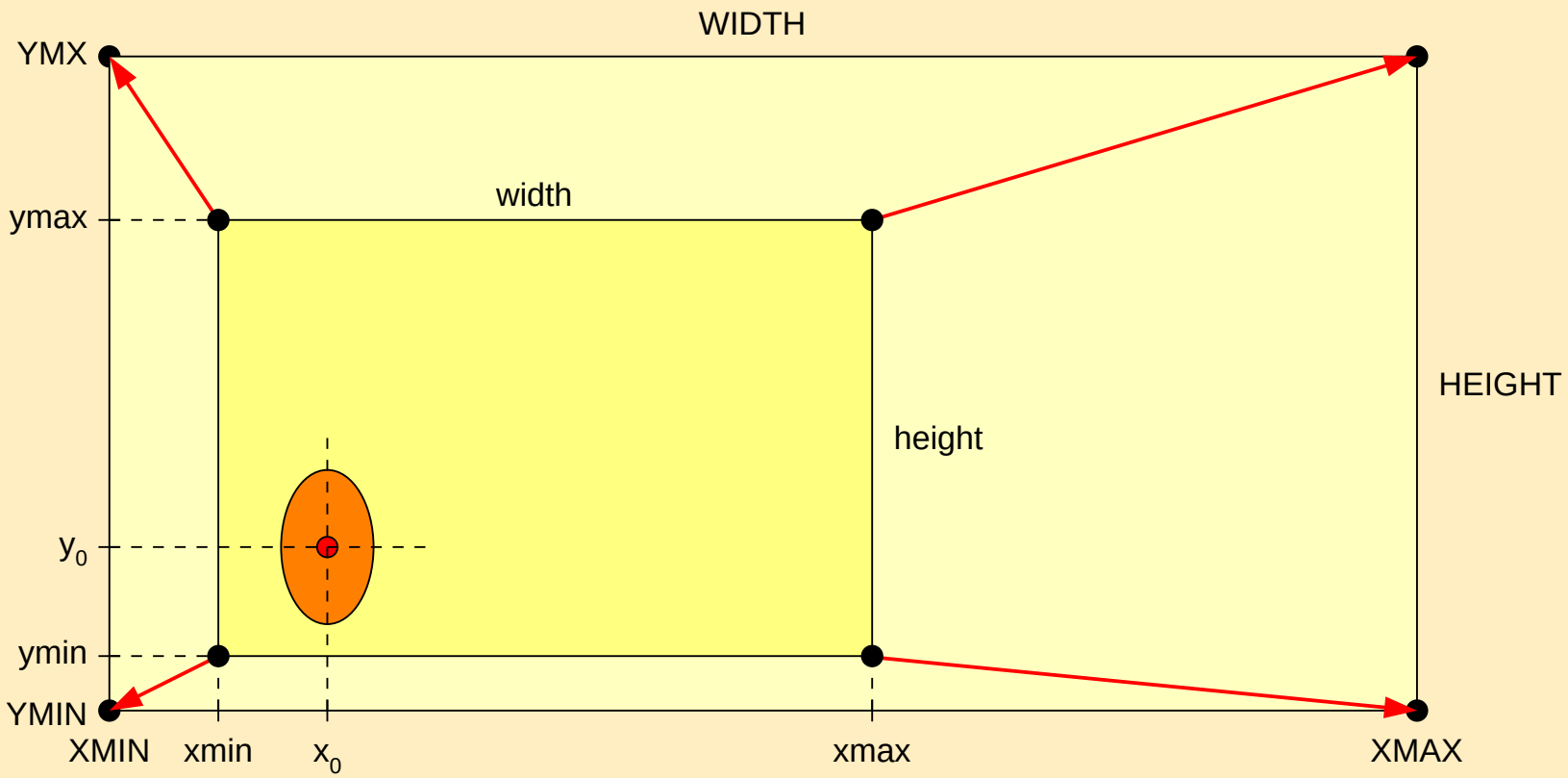
Outline

Fixing an Arbitrary Point

- We may keep any point fixed.
- If the window is expanded, then the expansion will reveal area to the left and right, and the bottom and top, in proportion to the fixed point's location.
- For example, if the point is $\frac{1}{6}$ of the way from xmin to xmax, then $\frac{1}{6}$ of the revealed area will be to the left and $\frac{5}{6}$ will be to the xmax.
- The same holds in the vertical direction.

Fixing an Arbitrary Point

Fixing an Arbitrary Point



Fixing an Arbitrary Point

- To keep the point (x_0, y_0) fixed, we must have the relation

$$\frac{x_0 - XMIN}{x_0 - xmin} = \frac{WIDTH}{width}.$$

- Therefore,

$$XMIN = x_0 - \left(\frac{WIDTH}{width}\right)(x_0 - xmin).$$

Fixing an Arbitrary Point

- The full set of equations is

$$XMIN = x_0 - \left(\frac{WIDTH}{width}\right)(x_0 - xmin),$$

$$XMAX = x_0 + \left(\frac{WIDTH}{width}\right)(xmax - x_0),$$

$$YMIN = y_0 - \left(\frac{HEIGHT}{height}\right)(y_0 - ymin),$$

$$YMAX = y_0 + \left(\frac{HEIGHT}{height}\right)(ymax - y_0).$$

The framebufferSizeCB() Function

The framebufferSizeCB() Function

```
void framebufferSizeCB(int width, int height)
{
//  Compute new window boundaries

    float x_0 = ···
    float y_0 = ···
    float ratio_w = (float)width/fb_width;
    float ratio_h = (float)height/fb_height;

    xmin = x_0 - ratio_w*(x_0 - xmin);
    xmax = x_0 + ratio_w*(xmax - x_0);
    ymin = y_0 - ratio_h*(y_0 - ymin);
    ymax = y_0 + ratio_h*(ymax - y_0);

    setProj();
        ⋮
}
```

Fixing an Arbitrary Point

- For example, if we wanted to keep the upper-right corner fixed, then $x_0 = xmax$, $y_0 = ymax$ and the equations become

$$\begin{aligned} XMIN &= xmax - \left(\frac{WIDTH}{width}\right)(xmax - xmin), \\ XMAX &= xmax, \\ YMIN &= ymax - \left(\frac{HEIGHT}{width}\right)(ymax - ymin), \\ YMAX &= ymax. \end{aligned}$$