Resizing the Window

Lecture 6

Robb T. Koether

Hampden-Sydney College

Fri, Sep 1, 2017

Outline

Outline

Resizing the Window

- When the window is resized by the user, the framebuffer 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 left, int right, int bottom, int top);
```

- 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.

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 bou

    left = ···
    right = ···
    bottom = ···
    top = ···
        ⋮
    mat4 proj = ortho2D(left, right, bottom, top);
    glUniformMatrix4fv(proj_loc, 1, GL_TRUE, proj);
        ⋮
}
```

Outline

The `setView()` Function

The `setView()` Function

```
void setView()
{
    proj = ortho2D(left, right, bottom, top);
    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`, `left`, `right`, `bottom`, and `top` global variables.

The `setView()` Function

The `setView()` Function

```
void setView()
{
    proj = ortho2D(left, right, bottom, top);
    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`, `left`, `right`, `bottom`, and `top` global variables.
- Why is it ok?

The FramebufferSize Callback Functionndow

- The question is, how to recompute *left*, *right*, *bottom*, and *top* when the window is resized?

The FramebufferSize Callback Functionndow

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

Outline

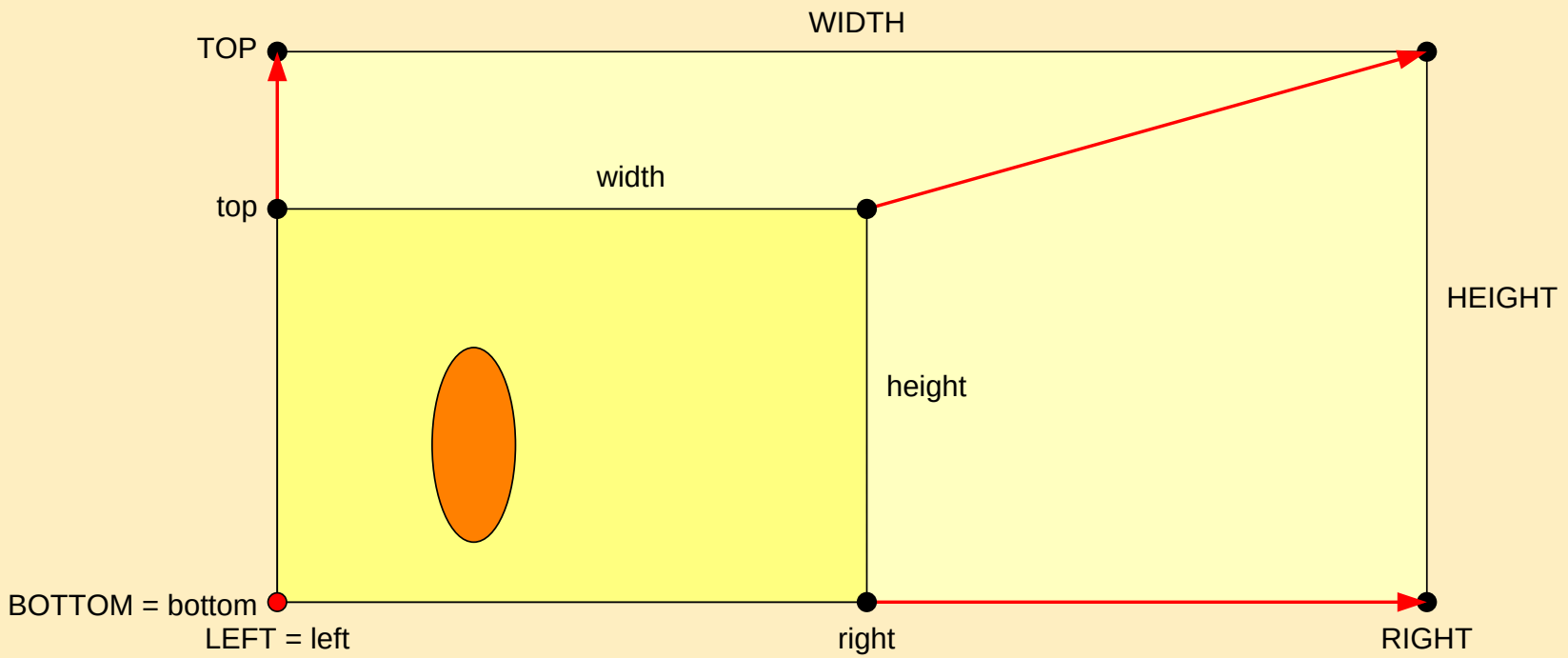
Fixing a Point

- We will label the new values *LEFT*, *RIGHT*, *BOTTOM*, and *TOP*.
- 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, left, right, bottom, top, width, and height are the "old" values and LEFT, RIGHT, BOTTOM, TOP, WIDTH, and HEIGHT are the "new" values.

Lower-Left Corner Fixed

- *left*, *right*, *bottom*, and *top* are in world coordinates.
- *width* and *height* are in screen coordinates.
- Be careful!

Lower-Left Corner Fixed

- *left*, *right*, *bottom*, and *top* are in world coordinates.
- *width* and *height* are in screen coordinates.
- Be careful!
- In the following formulas, lowercase letters ℓ, r, b, t, w, h represent the old values and uppercase letters L, R, B, T, W, H represent the new values.

Lower-Left Corner Fixed

- Clearly, $L = \ell$ and $B = b$.
- Also, we have the relation

$$\frac{R - L}{r - \ell} = \frac{W}{w}.$$

Lower-Left Corner Fixed

- Clearly, $L = \ell$ and $B = b$.
- Also, we have the relation

$$\frac{R - L}{r - \ell} = \frac{W}{w}.$$

- Therefore,

$$R = L + \left(\frac{W}{w}\right)(r - \ell).$$

Lower-Left Corner Fixed

- Clearly, $L = \ell$ and $B = b$.
- Also, we have the relation

$$\frac{R - L}{r - \ell} = \frac{W}{w}.$$

- Therefore,

$$R = L + \left(\frac{W}{w}\right)(r - \ell).$$

- Similarly,

$$T = B + \left(\frac{H}{h}\right)(t - b).$$

The framebufferSizeCB() Function

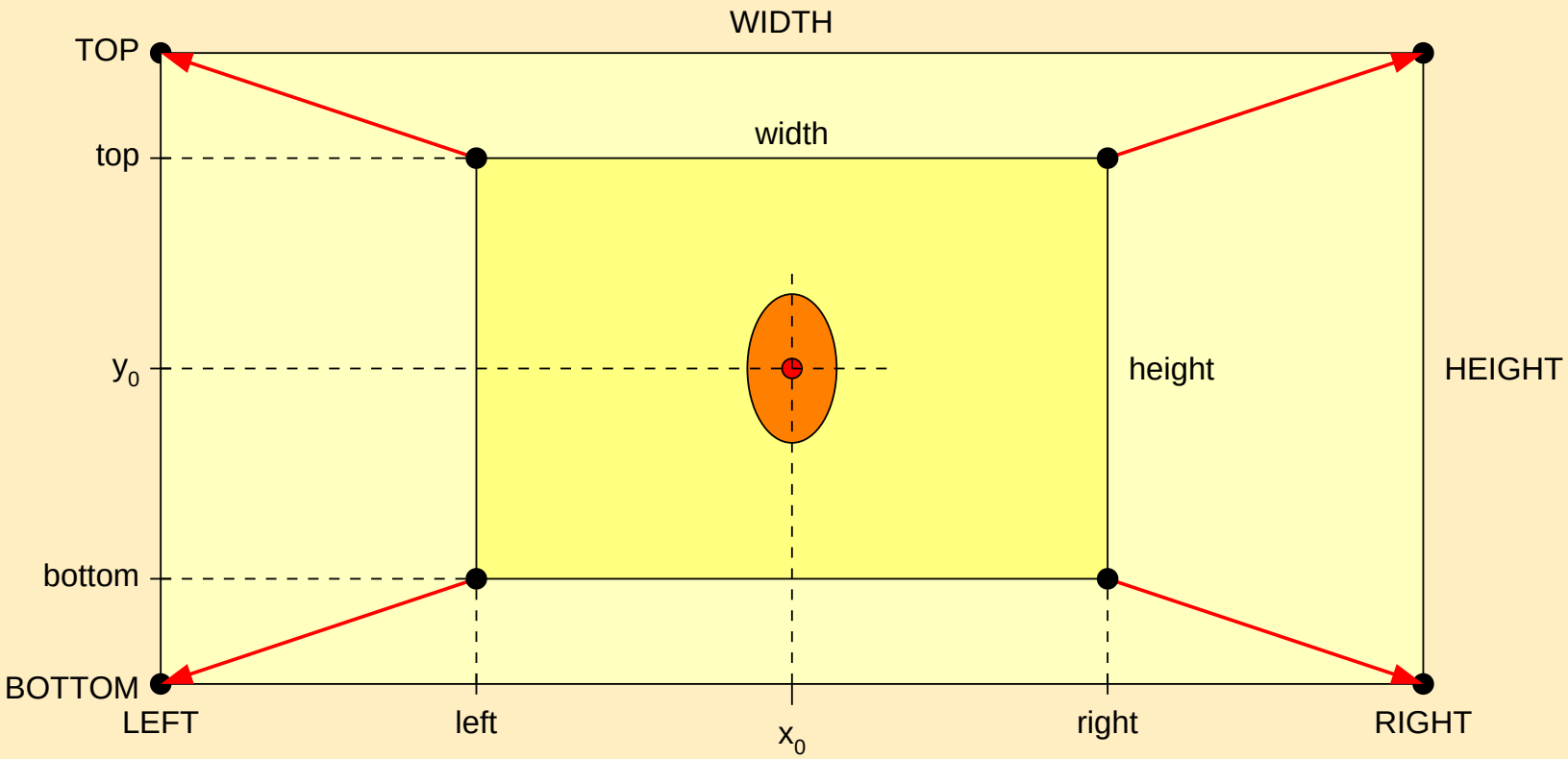
The framebufferSizeCB() Function

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

    right = left + (float)width/fb_width*(right - left);
    top = bottom + (float)height/fb_height*(top - bottom);

    setView();
        ⋮
}
```

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

    right = 0.5f*((1.0f + (float)width)/fb_width)*right +
        (1.0f - (float)width/fb_width*left);
    left = ···
    top = ···
    bottom = ···

    setView();
        ⋮
}
```

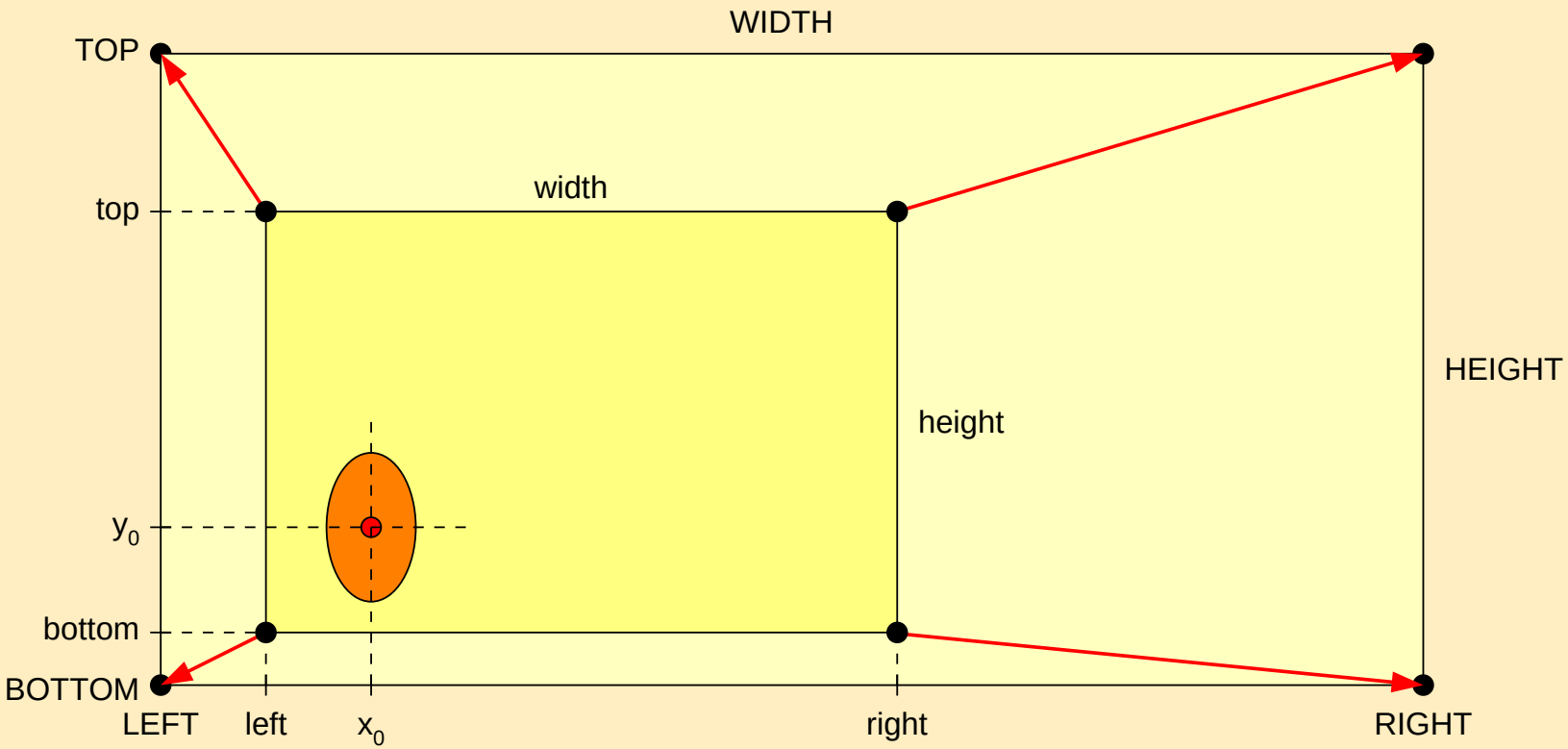
Outline

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 bottom, in proportion to the fixed point's location.
- For example, if the point is $\frac{1}{6}$ of the way from left to right, then $\frac{1}{6}$ of the revealed area will be to the left and $\frac{5}{6}$ will be to the right.
- 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

$$\frac{x_0 - L}{x_0 - \ell} = \frac{W}{w}.$$

- Therefore,

$$L = x_0 - \left(\frac{W}{w}\right)(x_0 - \ell).$$

Fixing an Arbitrary Point

- The full set of equations is

$$L = x_0 - \left(\frac{W}{w}\right)(x_0 - \ell),$$

$$R = x_0 + \left(\frac{W}{w}\right)(r - x_0),$$

$$B = y_0 - \left(\frac{H}{h}\right)(y_0 - b),$$

$$T = y_0 + \left(\frac{H}{h}\right)(t - y_0).$$

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;

    left = x_0 - ratio_w*(x_0 - left);
    right = x_0 + ratio_w*(right - x_0);
    bottom = y_0 - ratio_h*(y_0 - bottom);
    top = y_0 + ratio_h*(top - y_0);

    setView();
        ⋮
}
```

Fixing an Arbitrary Point

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

$$L = r - \left(\frac{W}{w}\right)(r - \ell),$$
$$R = r,$$
$$B = t - \left(\frac{H}{w}\right)(t - b),$$
$$T = t.$$

Outline

Assignment

Assignment

- Assignment 6