The Model Stack

Lecture 8

Robb T. Koether

Hampden-Sydney College

Wed, Sep 6, 2017

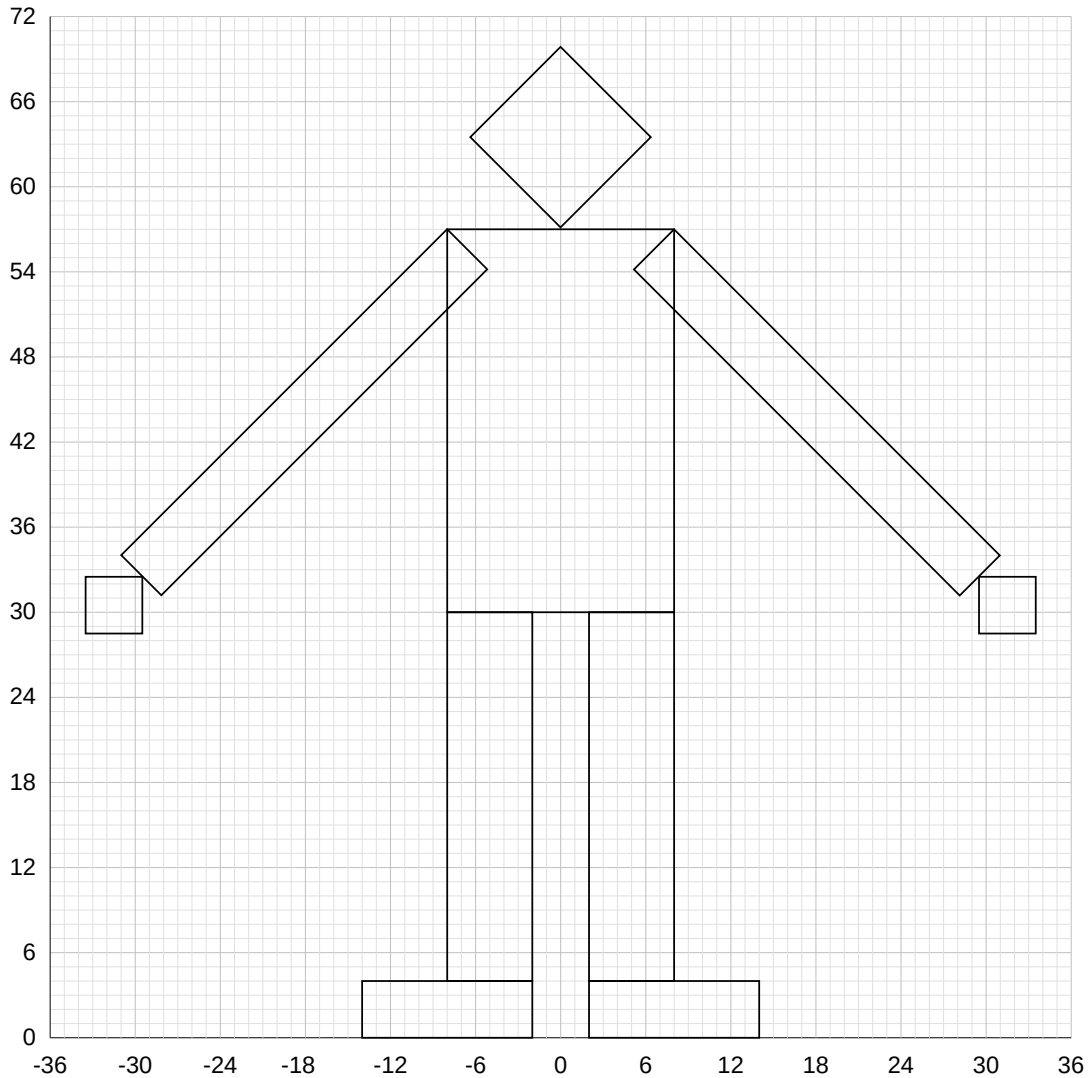
Outline

Outline

Drawing “Rectangle Man”

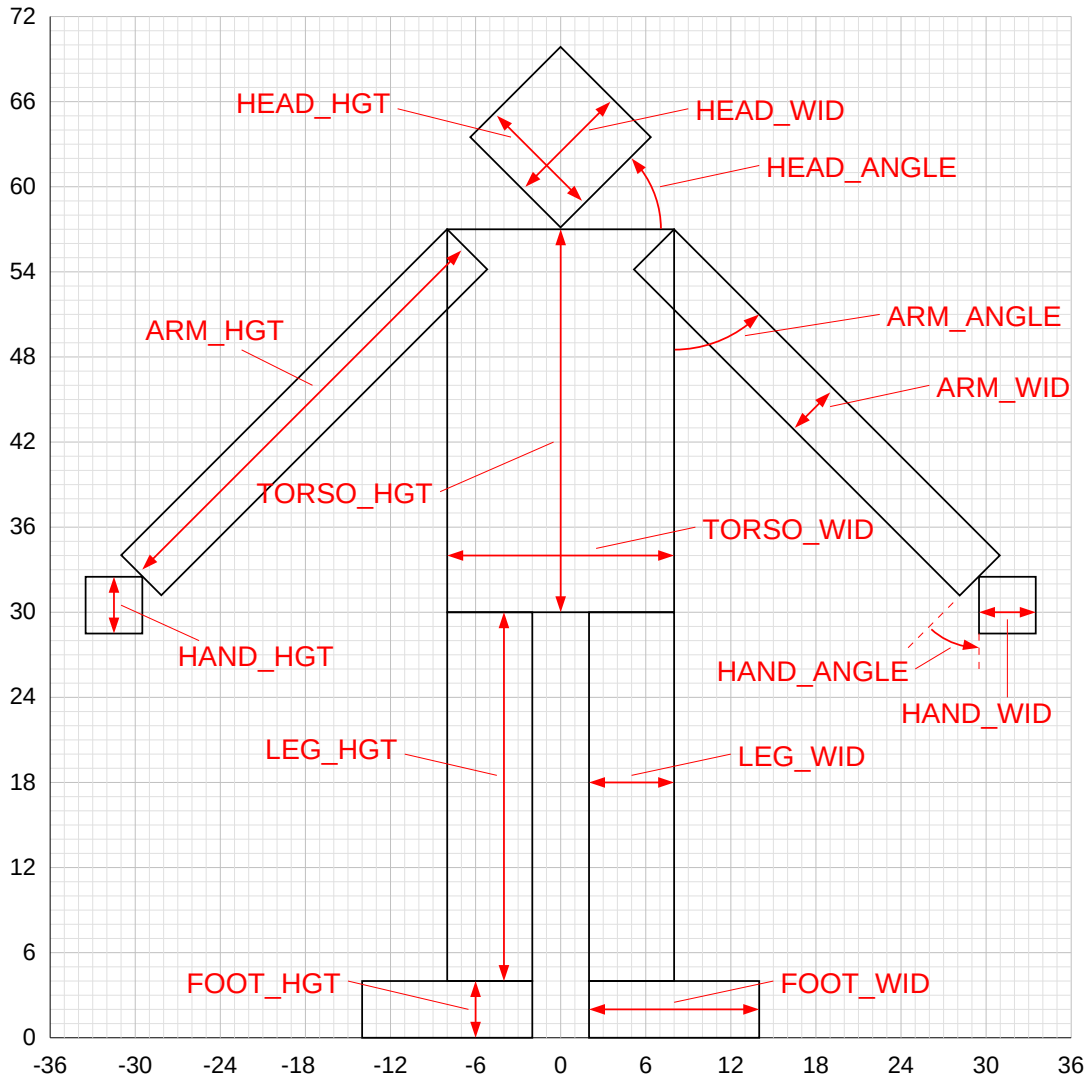
- We will draw an object (Rectangle Man) that is created entirely from our basic rectangle (unit square), using transformations.

Drawing “Rectangle Man”



Rectangle Man

Drawing “Rectangle Man”



Rectangle Man's Dimensions

The “Rectangle Man” Parameters

The “Rectangle Man” Parameters

```
//  The foot
    const float FOOT_WID = 12.0f;
    const float FOOT_HGT = 4.0f;
//  The leg
    const float LEG_WID = 6.0f;
    const float LEG_HGT = 22.0f;
    const float LEG_GAP = 4.0f;
        ⋮
```

- Assign a symbolic name to every parameter.

The “Rectangle Man” Parameters

The “Rectangle Man” Parameters

```
//  The foot
    const float FOOT_WID = 12.0f;
    const float FOOT_HGT = 4.0f;
//  The leg
    const float LEG_WID = 6.0f;
    const float LEG_HGT = 22.0f;
    const float LEG_GAP = 4.0f;
        ⋮
```

- Assign a symbolic name to every parameter.
- Don’t argue. Just do it.

Outline

The `ModelStack` Class

- I have created a class named `ModelStack`.
- As an object, it is a stack of 4×4 matrices.
- The matrix on top of the stack is the current matrix.
- The function `push()` will push a copy of the current matrix onto the stack, thereby duplicating it.
- The function `pop()` will pop the top matrix off the stack.

The `ModelStack` Class

The `ModelStack` Class

```
ModelStack();
```

- `ModelStack()` – Constructs an `ModelStack` object with an empty stack.

The `ModelStack` Class

The `ModelStack` Class

```
void init();
void push();
void pop();
```

- `init()` – Initializes the stack to the identity matrix.
- `push()` – Pushes a copy of the top matrix onto the stack, thereby duplicating it.
- `pop()` – Pops the top matrix off the stack.

The `ModelStack` Class

The `ModelStack` Class

```
void setModelLoc(GLuint m_loc);
void setNormalLoc(GLuint n_loc);
```

- `setModelLoc()` – Stores the shader location of the model matrix in the `ModelStack` object.
- `setNormalLoc()` – Stores the location of the normal matrix in the `ModelStack` object.

The `ModelStack` Class

The `ModelStack` Class

```
void mult(mat4 m);
void toShader(GLuint loc);
```

- `mult(mat4 m)` – Replaces top matrix with itself *right multiplied* by `m`.
- `toShader()` – Copies the model matrix and, if `normals` is true, the normal matrix to the shaders.

The Modelview Stack

- Pushing and popping are used to “remember” previous transformations.
- The basic pattern is
  - Push the current matrix onto the stack (to remember it).
  - Perform a series of geometric transformations and draw an object.
  - Pop the current matrix off the stack, thereby restoring the former “current matrix.”

Outline

Manipulating the Stack

Initialize the stack

```
ModelStack model_stack(...);          // Global
model_stack.setModelLoc(model_loc); // In init();
model_stack.init();                    // In display()
```

- Declare the stack globally.
- After defining the variable `model_loc` in the `init()` function, store its value in the `ModelStack` object.
- Initialize it in the `display()` function before any drawing is done.

Manipulating the Stack

Drawing an Object

```
model_stack.push();
{
    model_stack.mult(translate(2.0f, 0.0f, 0.0f));
    model_stack.mult(rotate(90.0f, 0.0f, 0.0f, 1.0f));
    model_stack.mult(scale(1.0f, 4.0f, 1.0f));
    model_stack.toShader();
    drawRectangle();
}
model_stack.pop();
```

- Transform and draw an object without losing the previous transformation.
- The effect is $\mathbf{v}' = \mathbf{CM} * \mathbf{T} * \mathbf{R} * \mathbf{S} * \mathbf{v}$, where **CM** is the current model matrix.

- Use the `ModelStack` class to create Rectangle Man.

Outline

Assignment

Assignment

- Assignment 8.