```
void Application::window(float window_width, float window_height, float window_depth,
    float window_x_position, float window_y_position, float window_z_position,
    float window_x_rotation, float window_y_rotation, float window_z_rotation,
    float window_x_rotation_2, float window_y_rotation_2, float window_z_rotation_2,
    float outer_trim_width, float outer_trim_depth,
    float inner_trim_width, float inner_trim_depth) {

    // draw glass
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(window_width - (window_outer_trim_width), window_height - (window_outer_trim_width), window_depth,
        window_x_position, window_y_position, window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.36, 0.43, 0.58);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object

    // ----- draw 4 outer trim -----------
    // bottom outer trim
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(window_width, window_outer_trim_width, window_outer_trim_depth,
        window_x_position, window_y_position - (window_height * 0.5), window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.1, 0.1, 0.1);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object

    // top outer trim
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(window_width, window_outer_trim_width, window_outer_trim_depth,
        window_x_position, window_y_position + (window_height * 0.5), window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.1, 0.1, 0.1);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object

    // right outer trim
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(outer_trim_width, window_height + outer_trim_width, outer_trim_depth,
        window_x_position + (window_width * 0.5), window_y_position, window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.1, 0.1, 0.1);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object

    // left outer trim
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(outer_trim_width, window_height + outer_trim_width, outer_trim_depth,
        window_x_position - (window_width * 0.5), window_y_position, window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.1, 0.1, 0.1);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object

    // draw inner trim
    // vertical inner trim
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(inner_trim_width, window_height - (outer_trim_width), inner_trim_depth,
        window_x_position, window_y_position, window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.1, 0.1, 0.1);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object

    // horizontal inner trim
    bm_window_object_ptr = new basic_model;
    bm_window_object_ptr->mesh = rectangular_prism(window_width, inner_trim_width, inner_trim_depth,
        window_x_position, window_y_position, window_z_position,
        window_x_rotation, window_y_rotation, window_z_rotation,
        window_x_rotation_2, window_y_rotation_2, window_z_rotation_2, window_uniform_scale);
    bm_window_object_ptr->color = vec3(0.1, 0.1, 0.1);    // assign color to object
    bm_window_object_ptr->shader = just_shader;
    m_window_objects.push_back(bm_window_object_ptr);      // append current basic_model object


}
```