For my project, I aimed to create a scene like the famous painting, *The Persistence of Memory*:



I chose this scene because there are many 2D and 3D recreations of this painting, it involves interesting shapes, and it is artistically interesting.

To recreate the clock, I chose to use a torus for the gold rim, cones for the hands, a sphere (with very low Z values) for the clock face, and another sphere for the bell on top. While I was not able to achieve a full bending effect like I had planned, I was able to achieve a similar effect with scaling and rotation, and recreate the painting’s feel using background textures and lighting.



For the background wall, I chose a texture based on a painting called “*The disintegration of the persistence of memory*,” to further add to the atmosphere.

**Navigation**:

The user can navigate in 3 dimensions by using the A and D keys to go left or right (X axis), the W and S keys to go forward an backward (Z axis), and the Q and E keys to go up or down (Y axis).

Additionally, there are more nuanced camera controls: the user can pan the mouse to rotate the camera, and I have enabled unlimited panning for it. The user can also scroll the mouse to increase or decrease speed (within reasonably bounded values).

Additionally, the user can press “P” to return to a default projection view, or “O” for an orthographic view, as shown below:





**Coding and custom functions**

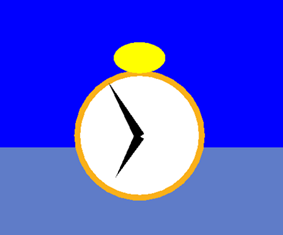
For my coding approach, I focused on coding one shape at a time, then adding additional features. This allowed making the project manageable, and taught me an enormous amount about OpenGL each week, from rendering meshes, to changing the .glsl shader files when needed for my 2-texture surfaces, to lighting. I heavily relied on the learning modules and StackOverflow for my code because this was a complex project, and they made learning the library significantly faster.

There were numerous custom functions that made the project easier and could be reusable in future projects. The most important one is probably ViewManager::ProcessKeyboardEvents(), which can be copied into any project using this framework to enable easy camera navigation. There are numerous other functions that will be helpful in changing the camera, such as Scroll\_Callback and Mouse\_Position\_Callback, but this implements the most important functionality. Also, I used custom functions to create shapes, from early basics like SetTransformations(), moving on to more complex ones like SetShaderColor() and void SetShaderTexture(), and on to SetShaderMaterial() and DefineObjectMaterials(), which I can reuse to set up custom textures in future projects.

However, the one I am most proud of is SceneManager::DrawClock, which I used to take my basic clock complex object, and use a series of transformations to create it at an arbitrary position. Building on this, and with help from StackOverflow and ultimately from Grok 4 Expert, I was able to use this function to place a clock object at any position, with any rotation, and with scales for each axis, ultimately recreating the feel of the painting.

Still, there is more code that could have been encapsulated in a function. In particular, the creation of the “floor” and “background wall” objects could have been done in a function that would have enabled an arbitrary starting position, and passing the textures in as a parameter for each one. Also, it would have been nice to create a function that automatically reads from a folder and creates a list of textures from the images in it, rather than manually typing in pictures to create textures from like I did.

Nonetheless, I am amazed at how much I learned in this course, going from this:



To this:



And ending up with this:

