

Tasks:

1. **(5 points)** Add a few new include (in stdafx.h) and lib files to our project to enable overlays – see the OgreView.
2. **(10 points)** Create a new `ssuge::Exception` class (plus the macro function), as discussed in class.
3. **(15 points)** Create a new `ssuge::Singleton` templated class, as discussed in class.
4. Create a new `LogManager` class
 - a. **(8 points)** Inherits from `Singleton`. Don't forget the convenience macro and the singleton-initializer-specialization.
 - b. Include a log message which does two things:
 - i. **(8 points)** save the passed message (string) to a log file. I'd like you to use `ofstream` for this. Display the date and time in this format:


```
10/5/2017@14:37:05 This was my log message..
```
 - ii. **(16 points)** at the user's¹ discretion, display, using an Ogre overlay (with a fixed number of text "slots"). If we have filled the slots, bump the first message.
 - c. **(+8 points)** Make the text strings fade out (the user might have to tell us how long to keep them alive) and remove themselves when the time-to-display runs out.
5. **(5 points)** Create and destroy (in the destructor) the `LogManager` instance.
6. **(8 points)** When the user presses a key (in `Application::keyPressed`), create a log message (using the convenience macro).
7. **Side-note:** you may find it helpful to override the `frameStarted` method inherited by `Application` (from `ApplicationContext`). Make sure to call the base-class method and return false if it returns false (to indicate we want to quit).
8. **(+10 points)** When the user presses a key (F11, maybe), display some debug stats using an Ogre overlay that includes:
 - a. The number of triangles
 - b. The average fps
 - c. ...
 - d. (you can get these from the render window)
9. Here's a video of my completed solution: https://youtu.be/Z_N9RLn5d_g



¹ And by user, I mean the code calling it from the `Application` class. And no: this part is NOT optional.