**Framework resume**

**Core Class**

- It does all the **general initializations**: SDL, window creation, OpenGL context etc.

- It creates a **Game Object** passing the window.

- Contains the **main event loop** of the project. So it **checks all inputs** (mouse, keydown, quit etc) and call the respective functions in Game.

- If the user doesn’t want to quit the game, then it **calculates the elapsed time** (elapsedSeconds) and then it does the following steps **every frame**:

1. Call the **update** function from the game with the elapsedSeconds (time in seconds!).
2. Call the **draw** function to draw in the back buffer
3. **Updates the screen**. It swaps the back and front buffer.

**Game Class**

- It manages (creates, updates, deletes, draws etc) all your game objects.

**Window Struct**

- Located in the Structs.h file.

- Contains **info** about the window, such as the **title** and the **dimensions**.

- The **Core class** needs it when it creates the SDL window

- The **Game class** needs it to know the dimensions of the window.

- It is passed to the Core object and this one will pass it to the Game object when it is created.

**Main function**

- It creates a Core object and call the run function.

- The run function is the main loop, and creates the Game object, handle all the events, call update and draw from the Game object.

**Precompiled header**

- Is a **header file** that includes all the files that hardly ever change and are frequently used in your project.

- This way you don’t have to include all the files that you need and you have them in one single file.

- This **reduces the build time** and you no longer have to explicitly include them in your code files.

- You only add a pair of special files (pch.h and pch.cpp) to your project and add these include directives in pch.h. Then you add the “#include pch.h” at the first line to each cpp file of your project.

**Detecting Heap memory problems**

- The function called **“StartHeapControl”** in the main function, detects **heap corruption and memory leaks**.

- This error is showed when we close the program. So it is useful to detect when we are not freeing the memory used by dynamic objects for example.