The Journal of Open Source Software

Augmented Reality UNIX C++ Engine for Enhanced Visual Guidance in Woodworking

Andrea Settimi [1¶], **Hong-Bin Yang** [1], **Julien Gamerro** [2], **and Yves Weinand** [1]

1 IBOIS EPFL, Switzerland **2** Independent Researcher, Switzerland ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

Statement of need

Functionalities

Layer-stack flow

The layer stack is primarily responsible for managing the flow control of the AR engine. Designed as a modular system, each layer encapsulates the code for a specific domain of the AR application, such as camera processing, object tracking, UI, and rendering. The general order and expansion of these layers can be configured in the top-level main file `ACApp.cpp`.

Each layer in the stack inherits from a superclass interface defined in `Layer.h`, which includes event-like methods triggered at various points during frame processing (e.g., `OnFrameAwake()`, `OnFrameStart()`, etc). These methods are invoked by the main `Run()` function in the singleton application loop from `Application.h`. This design allows application tasks to be containerized and executed sequentially while facilitating data exchange between specific layers through the `AIAC_APP` macro, enabling the retrieval of any particular layer data. Exchange between layers can also take place in a more structured way with the integrated event system (`ApplicationEvent.h`), which is capable of queuing events from layers and trigger them in the next main loop.

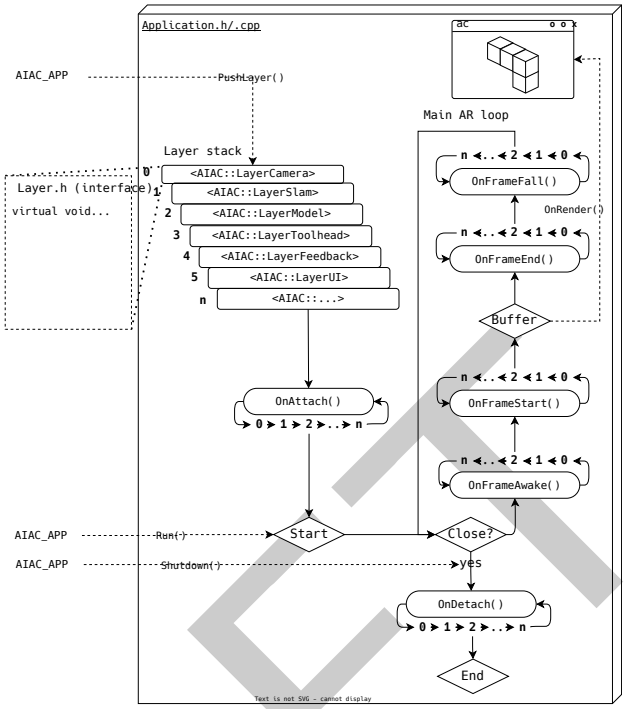

Figure 1: Illustration of the layer-stack design and the main loop for the AR engine.

Geometry framework

Computed Feedback System

AR rendering

Acknowledgements

References