# SeeGOL: 16-bit Graphics-Capable Operating System

Schuyler Martin <sam8050@rit.edu> Advisor: Warren Carithers

## Introduction

- SeeGOL is a new operating system, written to target the 16-bit Intel x86 architecture using modern tools. SeeGOL was originally targeted for the IBM PCjr (1984). However there is no fully-featured modern compiler that targets the original Intel 8086 processor.
- Developing a 16-bit OS is both easier and harder than writing a 32 or 64-bit OS. All x86 computers believe they are 16-bit machines when they boot so the boot process is simple. This is called Real Mode.
- SeeGOL's first goal is to represent what I have learned in my BS/MS degree. I focused in systems programming for my undergrad and graphics for my graduate degree. Hence SeeGOL is an operating system (OS) with graphical capabilities.
- There are few good resources describing how to write an OS. The secondary goal of SeeGOL is to be an example OS that others can use for educational purposes.

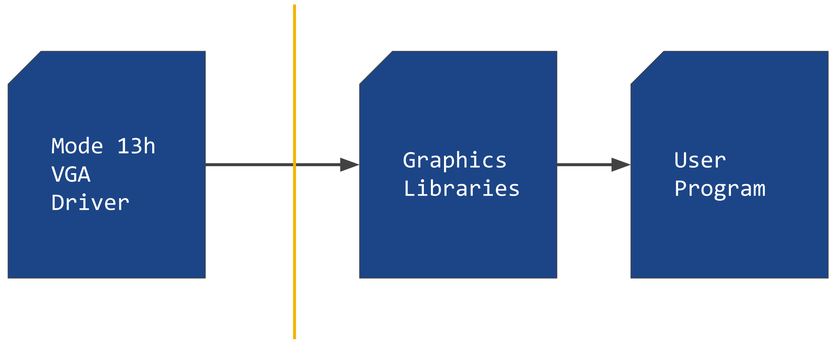
## Teaching Others to Build an OS

- SeeGOL was an open source project from day one, offering a unique learning opportunity for those interested in writing an operating system.
- Documentation is key. Each commit represents a revision made to SeeGOL. On top of that, I maintained a blog of the major issues encountered along the way. This can be found in the README file on GitHub.
- Writing an OS is no easy task. It takes dedication and time.

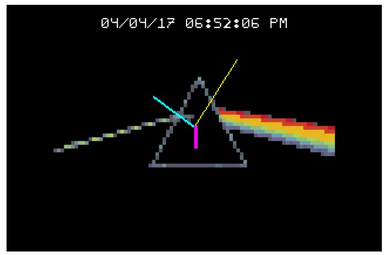

## How it Works - System Overview

- SeeGOL can be booted from an emulator, hard drive, or flash drive on any modern x86 machine (anything newer than and including the i386 processor).
- SeeGOL supports keyboard input, Intel's Real Time Clock (RTC), and IBM's Video Graphics Array (VGA). SeeGOL also includes a random number generator implemented as a software-defined Linear Feedback Shift Register (LFSR).
- Memory is not cheap in the 1980s. Real Mode can only address 1mb of RAM and offers no memory protection. Better yet, gcc limits the compilation size of SeeGOL to 28kb. To save space, all images in SeeGOL are stored using a custom run-length compression scheme I have invented, CXPM. CXPM is based off of the X Pixel Map (XPM) image format developed for the X Window System.
- There are no debugging tools that will work with SeeGOL. I had to develop my own I/O library and shell to manually test everything. I also had to invent my own font for the graphics mode.
- The graphics system is separated into two major components: VGA drivers and graphics libraries. A VGA driver must conform to a specification in SeeGOL. This forms a promise with the graphics libraries.
- A VGA driver provides hardware-optimized mechanisms for drawing pixels to the screen. The graphics libraries, on the other hand, are hardware agnostic. They provide user-level drawing functionality, such as drawing a UI pane or providing an implementation of Bresenham's line drawing algorithm.

## Rendering Pipeline

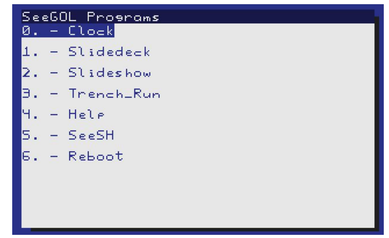


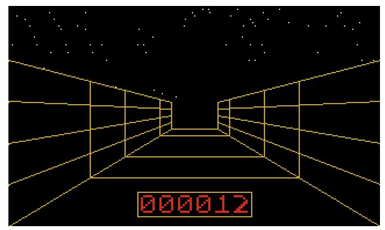

## Screenshots




User Clock Program

SeeGOL Program Menu




Trench Run Program

*Wish You Were Here* Render Test

Henrietta Semiconductor Corporation

Computer Science @ RIT
B. Thomas Golisano College of Computing & Information Sciences