



Project Description: SeeGOL

(Shoyler's Extremely Experimental Graphical Open Library)

Schuyler Martin <sam8050@rit.edu> <<http://shoyler.com>>
Computer Science, BS/MS
Rochester Institute of Technology
Computer Science MS Project, CSCI-788-02

The Premise

History

- IBM PCjr - 1984
 - IBM's first attempt at the home computing market
- 16-bit Intel 8088 processor
 - Modified 8086 with an 8-bit bus, 64kb-128kb RAM
 - 5¼ and 3½in floppy drive variants
- Major commercial flop
 - Could run stand alone programs and/or DOS



In an alternative universe...



The Future Sucks

- No modern C compiler can target the 8086
 - Some experimental compilers exist but are missing major features
- PCjr problems with floppies
 - Most models only have one 5¼in floppy drive
 - 3½in drives only read 768kb floppy format
 - Rare, 1.44mb are far more common



A monument to compromise



16-bit 8086



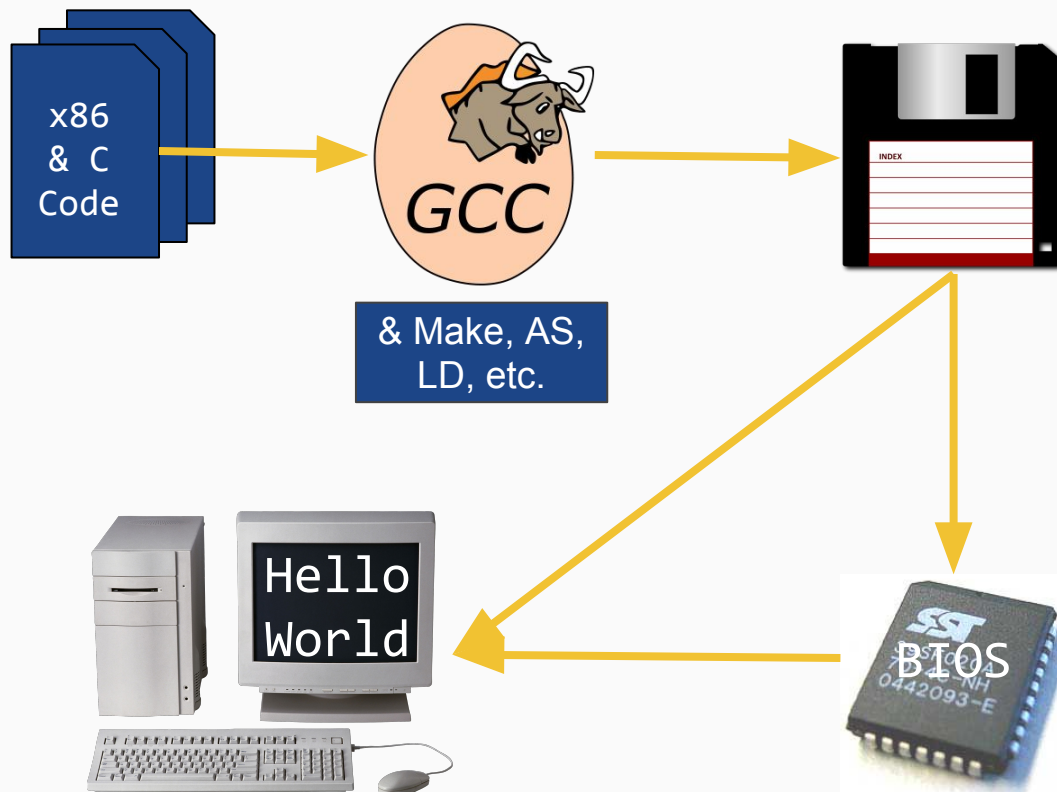
32-bit i386+
(with 16-bit Real Mode)

Project Plan

Stage 0: The Bootloader

- Boot the OS!
 - Off of a floppy
- Build the build system
- x86 assembly loads code onto the computer and calls code written in C

Development Toolchain and Booting



Linux System Start-up Logs

Stage 1: Debugging Tools

- Kernel-level input functions and print debugging
 - *KIO* library
- Has the ability to log text while in graphics mode

```
295
audit(1216470015.968:3): policy loaded auid=4294967295 ses=4294967295
INIT: version 2.86 booting
        Welcome to Red Hat Enterprise Linux Server
        Press 'I' to enter interactive startup.
Setting clock (utc): Sat Jul 19 05:20:22 MST 2008           [ OK ]
Starting udev:                                           [ OK ]
Loading default keymap (us):                             [ OK ]
Setting hostname rhce-prep.example.com:                 [ OK ]
No devices found
Setting up Logical Volume Management:
    No volume groups found                                [ OK ]

Checking filesystems
/: clean, 4871/263232 files, 72321/263056 blocks
/home: clean, 117/130560 files, 27384/522080 blocks
/var: clean, 1165/130560 files, 65117/522080 blocks
/dev/md0: clean, 12/883872 files, 45604/883456 blocks
/usr: clean, 81733/524288 files, 427747/524120 blocks
/boot: clean, 33/66264 files, 24068/265040 blocks

Remounting root filesystem in read-write mode:          [ OK ]
Mounting local filesystems:                             [ OK ]
Enabling local filesystem quotas:                       [ OK ]
```

Stage 2: Graphics Mode

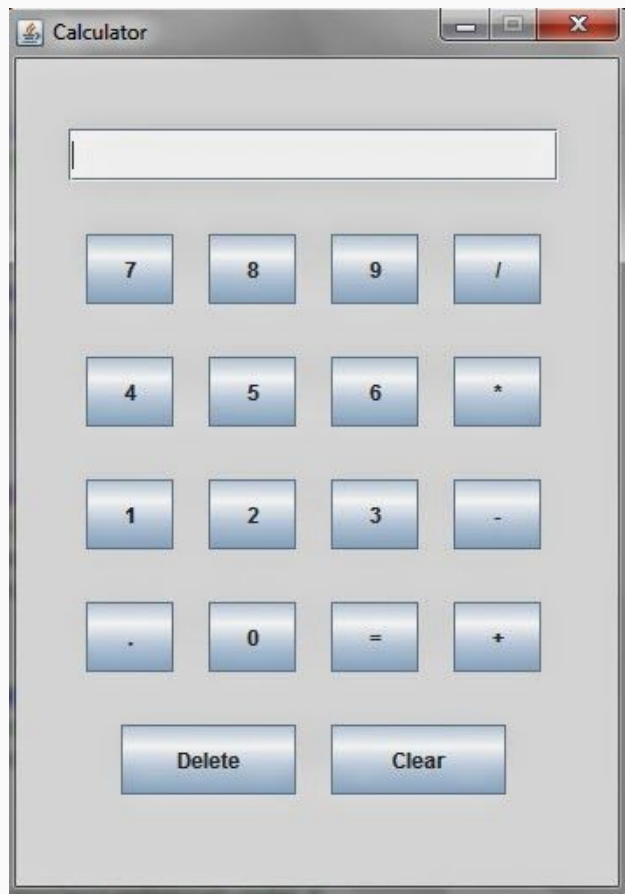
- Basic faculties to draw colored pixels to the screen
- In other words, “write the VGA driver”



Example of a Java Swing Program

Stage 3: Graphics Library

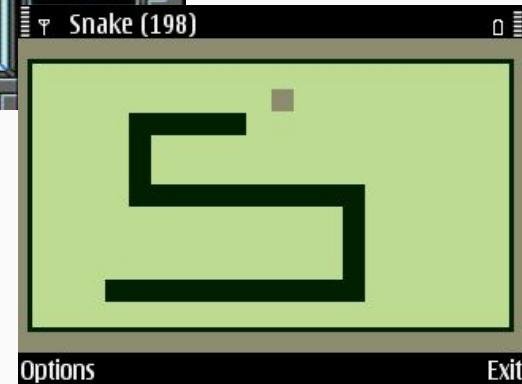
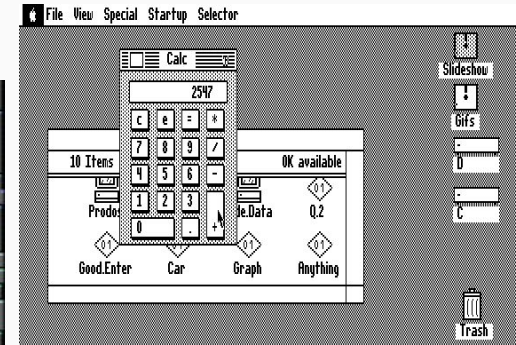
- Use the VGA driver to provide user-level GUI creation tools
- Analogous to *Swing* and *JavaFX* packages in Java



Stage 4+: Demo Programs & Advanced Features

- Proves the concept
- Software suite to prove that SeeGOL provides adequate features in the library

A Few Demo Program Ideas



Sources

- [1] Image content comes from freely available online resources
- [2] Diagrams and Code Snippets by Schuyler Martin
- [3] HSC Logo created by Kailey Martin
- [4] List of resources that were deemed helpful while making this project:
<https://github.com/schuylermartin45/seegol/blob/master/docs/links.txt>

Special Thanks

[1] Prof. Warren Carithers - Advisor

Warren, taught me almost everything I know about Systems Programming and Computer Graphics. Without him, none of this would be possible.

[2] Prof. Sean Strout - Mentor

Sean is a close friend of mine and initially sparked a lot of my interest in becoming a C wizard.

[3] Prof. Thomas Kinsman - Mentor

Thomas has taught me how to think creatively with visual problems

Questions?

Project available at <https://github.com/schuylermartin45/seegol>