그래픽스 언어 대모험

키네티쨩

발표 이유

- 갓과목 그래픽스를 수강하세요
- 님들은 저처럼 삽질하면 안됩니다
- 걍 얌전히 C++을 써보세요
- C++을 안쓸거면 잘 들으시면 됩니다.

해야되는거

- OpenGL
- GLUT 라이브러리 - 티폿을 그려야하므로

하라고 한거

- C++ / OpenGL
- 유니티 오브젝트 하이어라키
- 물체 움직이고 드래그하기
- 점 잇기
- 투명 박스 만들기
- 레이 트레이싱 (광자를 추적하여 개 사실적인 이미지 뽑기)

고려는 해본거

- Python
- ~~Java~~
- Ruby(WebGL)
- C#
- Clojure
- OCaml
- Racket
- Swift
- Objective-C
- Scala
- Rust
- Nim
- Common Lisp
- F#
- Julia

진짜로 코딩해본거

- Python
- C#
- Swift
- Rust
- Nim
- Common Lisp

이딴거로 한 이유

- 조교를 괴롭히고 싶어서

첫번째 - C#

- OpenGL을 쓰려면 OpenTK라는 라이브러리가 필요
- C#으로 3D 뭔가 만들거면 걍 DX를 씁시다

문제점

- 언어는 내가 본 언어중에 최고의 완성도를 지님
- GLUT 지원이 끊겨서 쓰기 힘들다
- 티폿을 못그림 ㅜㅜ
- GLFW 라이브러리는 맥 Xamarin으로 쓰기 힘듬

다시 첫번째 - Python

- 개빨리 만들 수 있음
- 하루만에 만들었다
- 웬만하면 편하게 Python 쓰세요
- 게임 만들고 싶어서 들은거면 걍 C++을 씁시다

두번째 - Common Lisp

- 조교를 괴롭히기 위한 최상의 방법이 무엇일까 고민
- 괄호와 괄호와 괄호

장점 / 단점

- 일반적으론 Emacs를 쓰지만 Sublime Text에 paredit을 깔면 쓸만한 에디터가 된다
- 함수형이다 - 디버깅이 편함
- 사실은 절차형으로 대부분 코딩한다
- Common Lisp가 Lisp계의 C++이라 쓸데없는 함수와 기능이 많음
- cl-opengl 라이브러리가 개발이 덜 되서 GL의 모든 함수를 지원 안함
- documentation이 전혀 안되어있음

세번째 - Common Lisp / Rust

- 김젼의 꼬임에 넘어가서 Rust를 사용
  - 다만 해당 시기엔 Rust가 1.0도 안나오고 GL 라이브러리도 한창 개발중이라 맥에서 왠지 빌드가 잘 안됨
  - 그래서 그냥 GL은 Lisp로 그리고 Rust로는 점만 잇는걸로

Rust - 장점 / 단점

- 메모리 적으로 완벽한 프로그램을 만들 수 있음
- 컴파일러에게 코드를 던져주는 노예가 되어야 한다
- 우리가 원하는 것은 2주안에 빨리 보이는 것만 그럴듯하게 대충 만드는것
- 결론적으로 과제로 쓰는건 비추

네번째 - Common Lisp

- 했다

5번째 - Swift

- OpenGL을 안쓴다
- 좋은 언어임 (옵씨보다 일백배)
- 코딩이 재밌어짐
- 조교가 맥이 없다
- 못씀 ㅜㅜ

5번째 - Rust

- 쓰려고 객체 디자인을 하다보니 문제가 발생
- 객체 상속이 불가능
  - 뭐 안쓰고도 할 수 있는데 이거 없이 쓰는걸 생각하기 귀찮았음
- 시간도 없었음 (약 5일)

5번째 - Nim

- Ray Tracer는 속도가 개 느리기 때문에 Python 쓰기가 불가능
- Python이랑 비슷한데 컴파일이 되서 속도가 빠른걸 찾음

장점

- Python-like 문법 + Static Type System
- 괜찮은 속도 (Rust랑 비슷하거나 약간 느림)
- C / javascript / objective-c와 같이 쓸 수 있음 (import/export 지원)
- 웬만한 C 라이브러리를 쉽게 가져다가 쓸 수 있다
- 잡다한 편리한 기능이 있음 (type inference / automatic boundary check / fast GC (non-tracing))

단점

- 개발자가 원칙이 없음 (자기가 재밌을 만한건 다 집어넣은 느낌)
- 대소문자 구분이 맨 앞글자만 / camelCase, snake_case를 같은 것으로 판단
- light.f(x) = f(light, x)
- proc f(light: type_name, x: type_name): ret_type_name:
- method f(light: type_name, x: type_name): ret_type_name:

그래도 쓸만한 이유

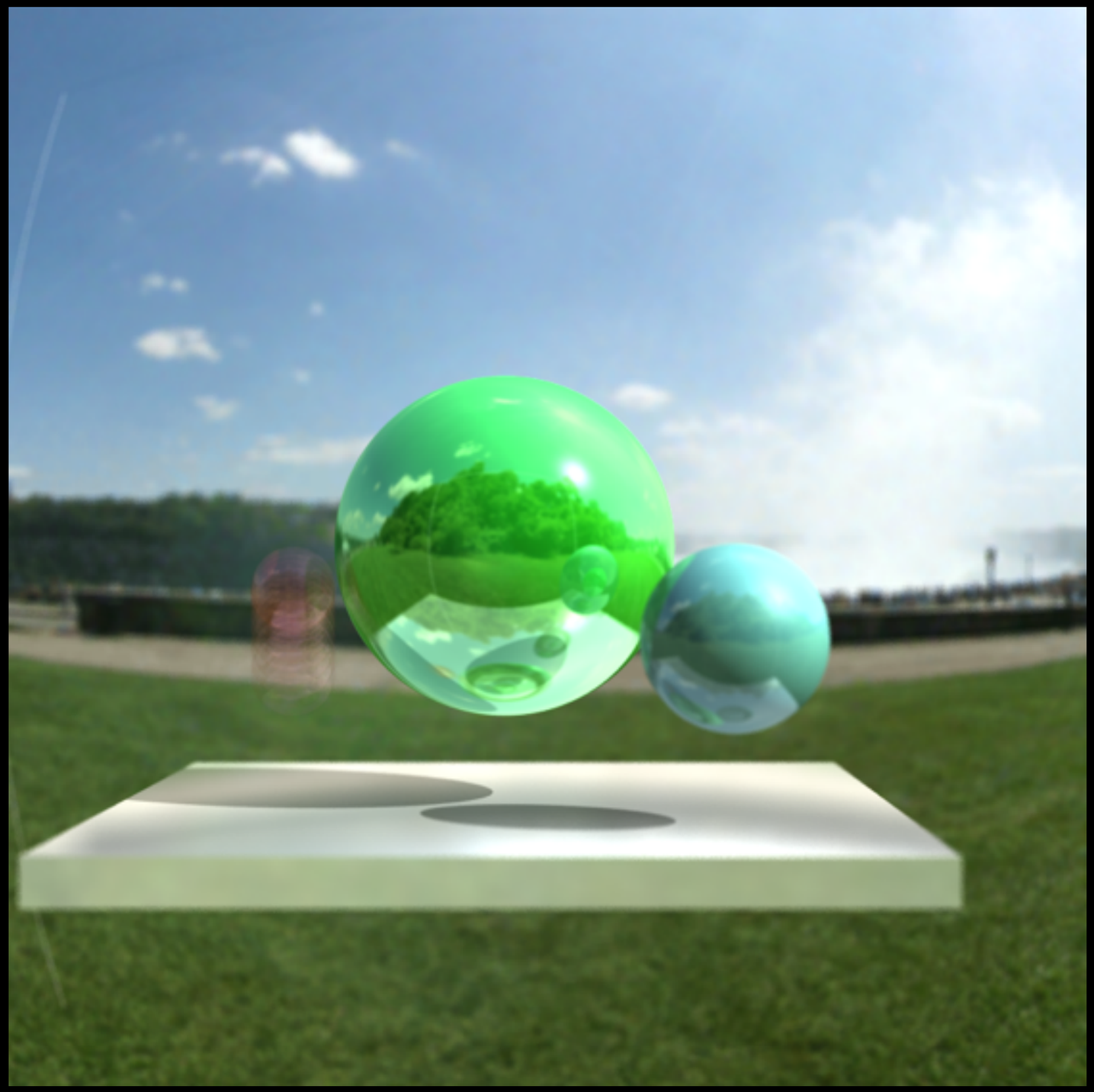
- 유지보수를 안할거면 빠르게 작성 가능
- python이랑 비슷해서 python을 알면 속도를 보장하면서도 쉽게 코딩 가능
- 과제에만 쓸거라면 상당히 쓸만하다

5번 과제를 하기 위해

- 기말이라 온갖 기말 프로젝트와 시험이 산재
- 시간이 없었다
- 하라고 한거 – phong illumination, 구 그리기, 폴리곤 그리기, recursive reflection/refraction, texture mapping
- 추가구현 점수 5점씩
- 복잡한 구현보다는 추가구현 점수를 잔뜩 노리자

결과

- 좋은 점수를 받음



결론

- 내가 생각하기에 충분히 조교를 괴롭힌것 같지 않다
- 그냥 C++을 씁시다
- C++을 모르면 Python을 씁시다
- 김젼을 사랑한다면 Rust를 씁시다
- 갓과목 그래픽스를 들읍시다