Intermediate Graphics & Animation Programming

GPR-300
Daniel S. Buckstein

Course Introduction & Graphics Programming Overview
Week 1

License

Introductions

- Course Instructor:

<u>Dan Buckstein</u>

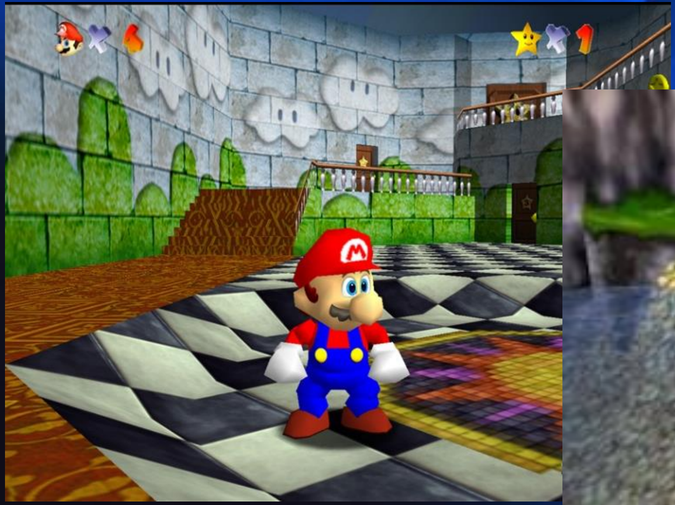
M.Sc, Computer Science, UOIT

B.IT, ***Game Development*** & Entrepreneurship

  - #uoitgamedev

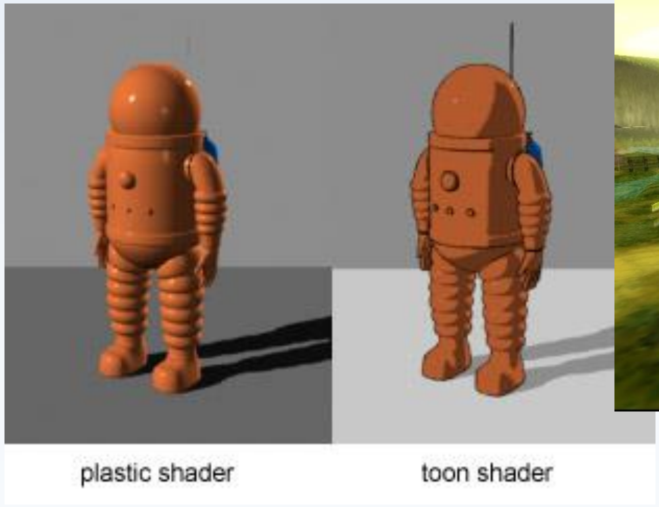
- Favourite games: Dragon Quest I-VI, Super Mario 64, Banjo-Kazooie

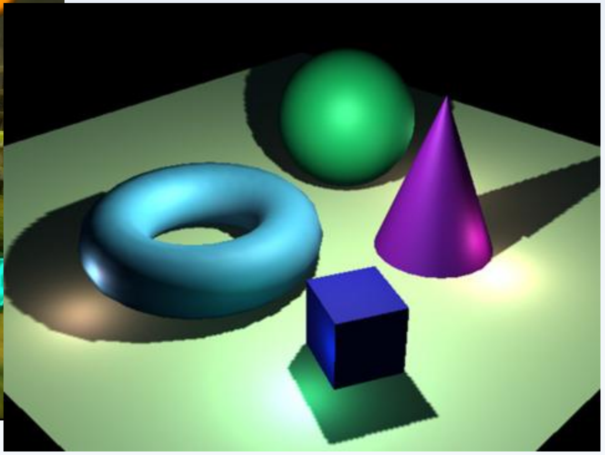
GRAPHICSSSSSS!!!!!!1!!!1!1one!!2



What is this course?

- Intermediate modern rendering with OpenGL
- Crash course in the fundamentals
- Industry-standard post-processing algorithms
- Intro to animation programming





What is this course?

- What is this course ***to you***?
- Fundamentally two things:

1. Portfolio building

Projects are creative in nature, and will show employers what you can do in this domain

2. Engineering

Low-level & tools programming that applies all you have learned thus far in your courses

Why does this course exist?

- Breadth of expertise
    - Introductory → Intermediate
- Learn to speak the other developer's language
- Industry need for **technical artists**

How to succeed in this course

- Practice programming often
- Do work often and on time
- Attend all lectures and tutorials
- Attend office hours to clarify issues
- ***<u>Do not procrastinate.</u>***

- This is your education… make the best of it!

How to succeed in this course

- Additional readings will be provided

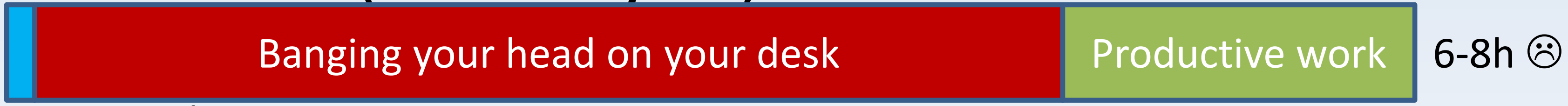
- Do your own research to excel with the course content

- When in doubt…

    - …or just come find me

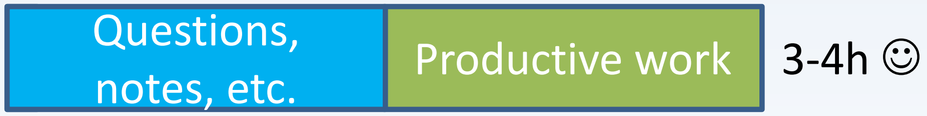
How to succeed in this course

- Time spent completing assignments:
- 8h/wk spent on course in total
    - 3h in-class, 1-2h reading & studying, 3-4h doing work
- Traditional (not okay ☹):



^prep to do assignment

- The way you want to do it:

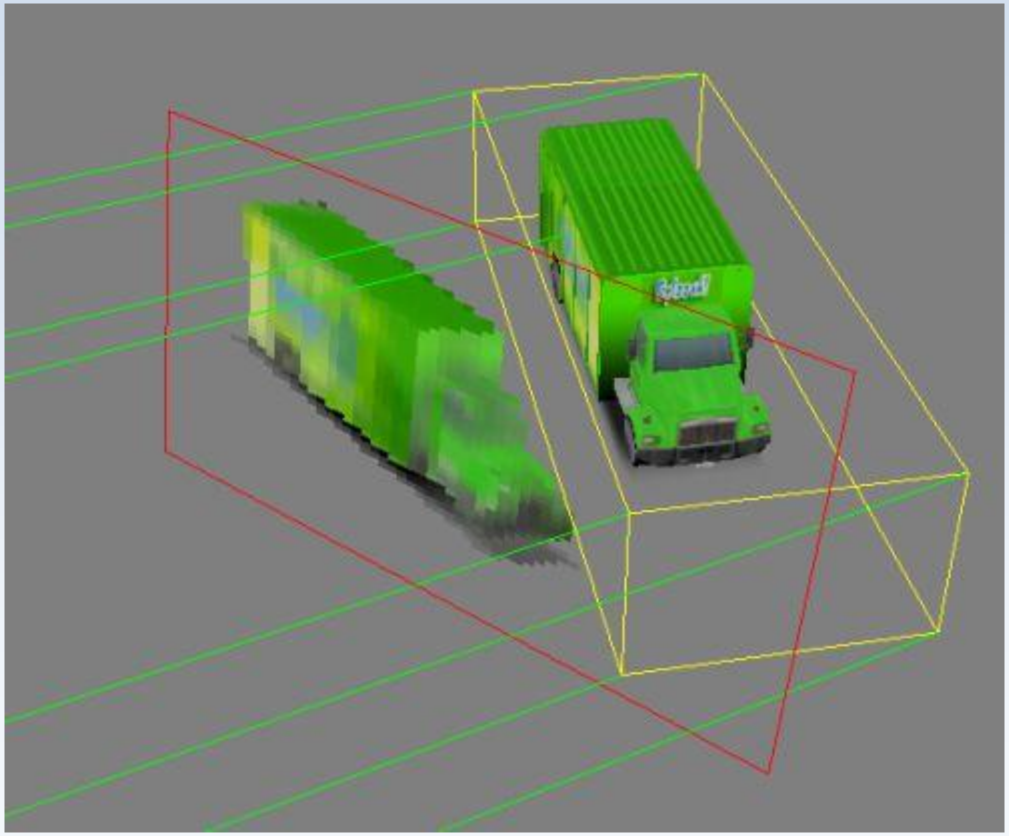


^prep to do assignment

SYLLABUS REVIEW

- Course syllabus is posted on Canvas
- Find course link for ***GPR-300: Inter. Graphics & Anim. Prog.***
- Syllabus is posted under the ‘Syllabus’ tab
- Other stuff posted under ‘Modules’

Impostor Syndrome



http://www.gamasutra.com/features/20060105/Figure5.jpg

Accessibility

- Again, feel free to approach me to discuss
- Works both ways…

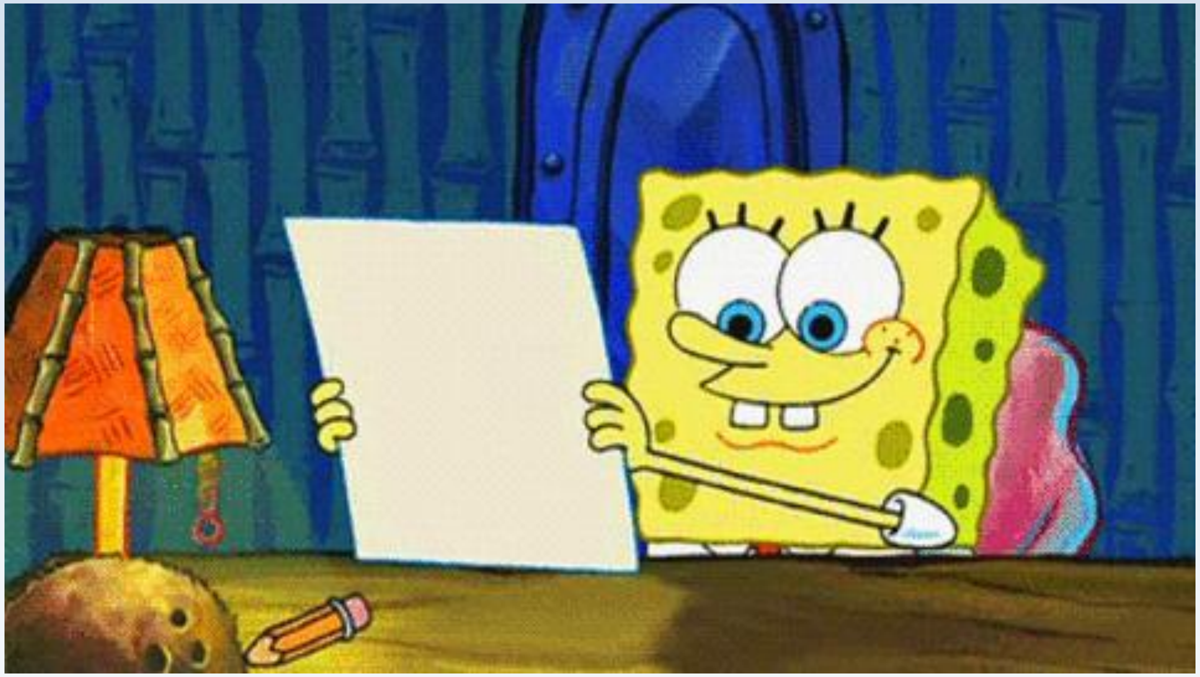
- Please sit closer to the front
- Please speak up
- Please do not mumble

Zero Tolerance for Plagiarism

Do not plagiarize.

Questions???

- Questions so far???



“Tools vs. Applications”

- Metaphor: what is this?

“Tools vs. Applications”

- Your survival in this course (and the rest of the program) relies partially on your ability to distinguish between ***tools*** and ***applications***
- ***TOOLS***: mathematical formulas, algorithms, theories, concepts, definitions…
- ***APPLICATIONS***: **use in your games!**

“Tools vs. Applications”

- Example: LERP

TOOL: The algorithm implemented in C/C++

```
vec3 lerp(vec3 v0, vec3 v1, float t)
{...}
```

APPLICATION: Move a character from A to B

```
myPos = lerp(posA, posB, posT);
```

APPLICATION: Colour blending

```
darkCyan = lerp(blue, green, 0.5);
```

“Tools vs. Applications”

- Math and programming go hand-in-hand!!!
- TOOLS vs. APPLICATIONS
- Algorithms are just mathematical formulas!!!
  - Tools
- Implementation of an algorithm is in code
  - Applications

“It’s Just Data”

- Course motto: ***“It’s just data.”***
- Remember this always!
- Algorithms can be used in many ways!
- Moral of the story: we are using algorithms to process ***data***
- Different purposes call for different applications of the same tools!!!

“It’s Just Data”

- Variables are just numbers
- Algorithms are just functions that take in and spit out variables

variable→ algorithm→ variable→ algorithm→…

```
float, int, vec2, vec3, mat4, frame,
keyframe, sequence, skeleton...
```

At the end of the day, it’s just stuff we process!

Frameworks!!!

- This year you should focus on building a solid ***framework***: a collection of tools (algorithms)
- Why bother?
- Do you want to implement your shader code every time you want to use it?
- Wouldn’t you rather call a function or instantiate a class for any case or problem?
- TL;DR: simplify your life ☺

Frameworks!!!

- Introducing ***animal3D***: the minimal 3D animation framework
- Graphics
- Windowing
- Input… and more!

Use version control

- Recommended SCM & GUIs:
  - **Git**, SmartGit
  - **Mercurial (Hg)**, Tortoise Hg

- Course materials delivered using **Git**

Highly-Recommended Software

- DIY:

• Visual Studio	→ programming IDE
• Tortoise Hg (and plugin)	→ source control
• p4merge	→ visual diff tool
• Rapid Environment Editor	→ env. var. editor
• FMOD Sound System	→ sound library & API
• Everything Search	→ super fast file search
• 7zip	→ compression
• cmake	→ cross-platform config tool
• TeXstudio & MiKTeX	→ for fancy PDFs

- Dan’s starter package:

• animal3D	→ graphics framework
• Developer SDKs	→ fun prerequisites
• RakNet	→ networking library

The end.

- Questions? Comments? Concerns?