CS4803/8803:
3D User Interfaces

Instructor: Blair MacIntyre
TSRB 231
blair@cc.gatech.edu

Today’s Agenda

- Introductions
  - Me
  - TAs
  - You
- Class Overview
  - Syllabus
  - Resources
  - Class Policies

Introductions

- Research: Augmented and Mixed Reality ("interactive 3D in the world"), Games, Graphics, HCI
- Professor, Coc/IC; Adjunt Professor, LMC
- Principal Research Scientist, Mozilla
- Design of MR/AR experiences (3D in the world)
- Background in math, CS, 3D graphics
- PhD research in distributed interactive 3D software design

Introductions

- TA

Now, It’s Your Turn

- Name
- Major, Year
- Interests
- Why 3D UI?

What is this class about?

- Organizing principles of 3D UI software
    - In particular, immersive 3D: AR, VR
- Practice in 3D UI implementation

- Three parts:
    - Software architectures
    - Fundamental Interaction Techniques
    - Advanced topics (animation, gestures, multimodal, etc.)

What this class is NOT about

- User-centered design
- Visual design
- Games
- Applications

Basic Course Info

- “Prerequisite”: 3451 / 6491
  - Remedial background texts: Any 3D graphics book
- Web materials
  - Up now:
    - https://3dui-class.github.io
  - General info (books/readings, exams, homework)
  - Syllabus
    - Will be updated throughout the semester
    - Will contain links to lecture slides

Resources

- Required (and free, links on syllabus web page):
  - *3D User Interfaces, Second Edition, by Joseph LaViola, Ernst Kruijff, Ryan McMahan, Doug Bowman, and Ivan Poupyrev. Available via O’Reilly Learning.*
  - *Understanding Virtual Reality: Interface, Application, and Design, 2nd Edition, William R. Sherman and Alan B. Craig, 2018. Available via Science Direct.*

Grading Criteria

- Grads and Undergrads similar, but different emphasis:
    - Two or three individual homework assignments
    - Research project -- two person teams
    - Writing, implementation, presentation
    - Midterm and final exam
    - In-class quizzes and activites
- See syllabus for grading breakdown

Other Policies

- **Homeworks** are in Typescript (required) and use web technology (Babylon.js)
  - Turn-in and late policy:
    - Due 11:59PM on the announced due date
    - Late turnins will be marked down 5% for each date they are late, maximum of four days
- **Project** work is more flexible
  - You can use other programming environments if web isn't adequate for your project
  - Multiple milestones involving written paper, implementation task, presentation and demo
- What you turn in must compile and run!
- Please pay attention to platform issues (Windows vs **Mac** vs Linux)

Important Note

- There will be minimal Typescript training in class
- If you are not comfortable with Typescript / Javascript / Web programming:
  1. Learn
  2. Drop course
- While examples and programming assignments use web technology, focus of the lectures is on broader 3D UI software concepts
  - You'll have to understand how these concepts are applied in our tools
  - I can help with a lot of this, but 3D toolkits are complex
  - Be prepared to do independent problem solving if necessary

GT Honor Code

- All homeworks are expected to be completed by individuals and not in collaboration with others. While you can ask others about Swing coding problems, no code sharing is allowed.
- Projects are done in teams of two; be clear about what code you wrote and what code you are reusing (if you're using external libraries, etc.)
- Obviously, all exams are to be completed by the student alone.

- Students are expected to follow the GT Honor Code. I am required to forward all suspected cases of academic misconduct to the Dean of Students, where they will be pursued to resolution.
- This is a very unpleasant experience for all involved, so please do not put us in this situation.

Motivation

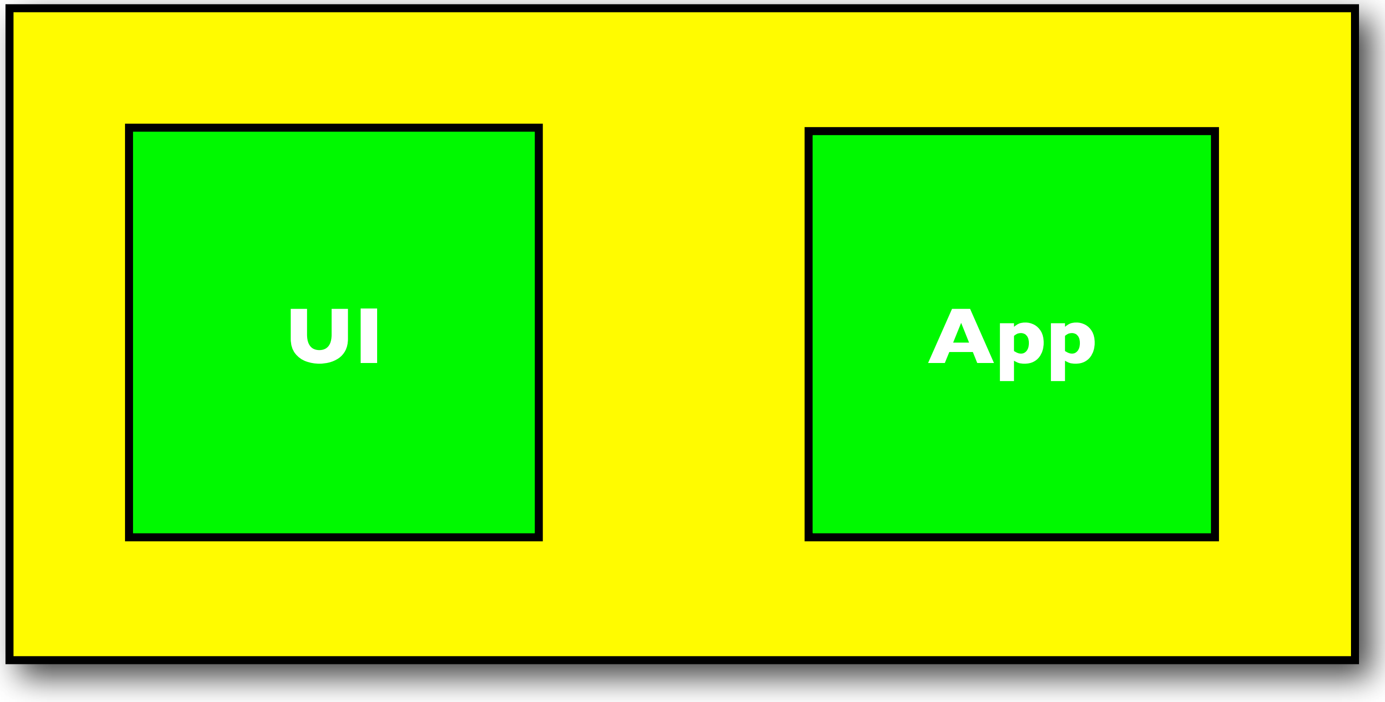
- AR and VR are “finally here”
    - Usable by more than just fans and tech researchers
    - Will soon be “just a feature”
- Tools are still relatively primitive compared to 2D UIs
- AR/VR/3D != Unity | Unreal

Why a class on 3D UIs instead of AR/VR/Games?

- Most systems are built for a user
  - Good user interfaces are critical
  - 3D UI’s are still the wild west compared to 2D
- Designing for users is *important* and *difficult*
  - Lots of code devoted to UI
    - Much more complex and hard in 3D compared to 2D
  - Hard to get right the first time (iteration necessary)

Programmer’s Perspective

- The “UI” is typically viewed as one component of the overall system
  - The part that “deals with the user”
  - Separate from the “functional core” (AKA the “app”)



Software Engineering and the UI

- Advantages of “separation of concerns”
  - Keep UI code separate from app code
  - Isolate changes
  - More modular implementation
  - Different expertise needed
  - Don’t want to iterate the whole thing

In practice, very hard to do...

- More and more interactive 3D programs are tightly coupled to the UI
    - Programs structured around UI concepts/flow
    - UI structure “sneaks into” application
- Side effect of the nature of 3D and the state of the tools
    - Tight coupling can offer better feedback/performance

Part I: Understanding 3D (and 2D) UI’s and Toolkits

- UI software architecture and organization (2D and 3D)
- Input and output
  - Devices, software abstractions
- Interaction techniques and how to implement them
- Toolkits and programming environments

Part II: Major 3D Interaction Types

- Selection and Manipulation
- Navigation and Locomotion
- System Control

Part III: Advanced Topics

- Two-handed input and interaction
- Animation
- Natural interaction types
  - Gesture, audio, video
- Advanced Sensing-based interfaces
  - Dealing with new sensors, context awareness
- Multi-modal interfaces

- Requests?