AR Design

Tue, August 11 (Week 8)
Design

As solving problems.

For software engineering: how to build an application that solves a problem.
For augmented reality: how to build an AR application that solves a problem.

Usually, having a framework for thinking helps a lot.
Model-view-controller

The dominant framework behind all applications built for users.

TODO app as an example:

Model: TODO tasks (data)

View: how to show the tasks

Controller: how to update the tasks

(Note: this is a very simplified version)
Model-view-controller

The advantage of MVC as a framework:

Allows splitting the software architecture into smaller pieces.

For example: if you separate the task from the button showing the task, you can write code for

1. storing the task in a database
2. updating the button based on the database

separately. Makes things easier.
Design Language

The set of terminologies you use to describe constructs.

Example: we used the word “button” in the previous slide. Surprisingly, we all now what it is.

There are names given to ideas and for 2D applications, there are implementations of them in code.
Example: Material Design

Google introduced Material Design in 2014.

While they build mobile and web applications, though the code behind them are written in different programming languages, they are based on the same design language.
AR vs. Others (PCs or Phones)

Model: almost the same

View: very different

Controller: very different (even PCs and Phones are different from each other)

View should be based on spatial components.

Controller is no longer based on keyboards, mouses, or touch screens.
Memory Lingering in Our Heads

People who work on AR applications have worked on 2D applications and cannot stop thinking in the 2D way.

Comparison:

For 2D applications, people built the design language from scratch.

For AR applications, people are building it based on the ones for 2D.
Many are coming from 2D applications.
WIMP

Windows, Icons, Menus, Pointer

Phones
To Inherit or Not to Inherit?

Smartphones inherited WIMP except for the mouse and pointer, by interpreting touches as mouse clicks.

Should we do the same thing for augmented reality? In that case we do:

Windows floating in air, icons for applications that pop up those windows, pointing using our hands or hand controllers.
How to Control AR Applications

One Example Question: Should we use eye gaze as a pointer?

Advantage: we can point at things very quickly.

Disadvantage: often we do not want to point at it but just look at it.
How to Control AR Applications

Another Example Question: Should we ask people to touch a button or point the button and make a gesture?

Advantage of touching: it is very simple to understand how to use the buttons.
Disadvantage: users have to move to the buttons to click them.
How to Control AR Applications

Holding arm in air is hard. Perhaps, it might be better to have floating screens, sometimes attached on walls. (There is a reason why mouses are so popular without floating in air.)
Design Guidelines

There are attempts to provide a design guideline.

Examples from Microsoft and Google:

Augmented Reality Design Guidelines

Introduction