MobileHCI 2014
Interactive Tutorial

Mobile-based Tangible Interaction Design for Shared Displays

Ali Mazalek
mazalek@ryerson.ca

Ahmed S. Arif
asarif@ryerson.ca

Synaesthetic Media Lab
Ryerson University & Georgia Tech

synlab

http://synlab.gatech.edu

Schedule

9:00-9:10	Introductions
9:10-9:25	Tangible Tabletops and Mobile Devices
9:25-9:40	Interaction Scenario and Demonstration
9:40-10:30	Setup and First Exercise
10:30-10:50	Coffee Break in the hall
10:50-11:45	Design and Development Activity
11:45-12:00	Presentation and Discussion

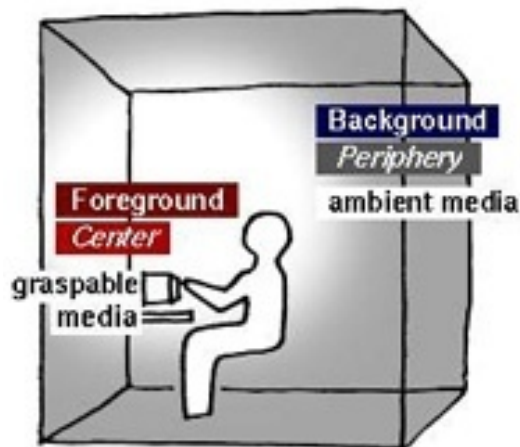# Tangible interaction

[Ishii & Ullmer 1997]

Gives physical form to digital information by seamlessly coupling the dual worlds of bits and atoms

Uses physical spaces, surfaces, and objects as both *controls* and *representations* of digital information

Two main categories:

Foreground interactions with graspable objects and augmented surfaces
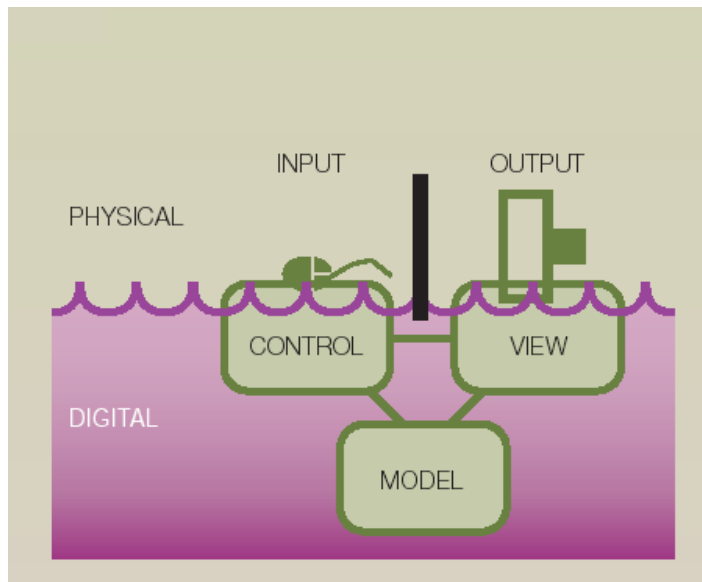
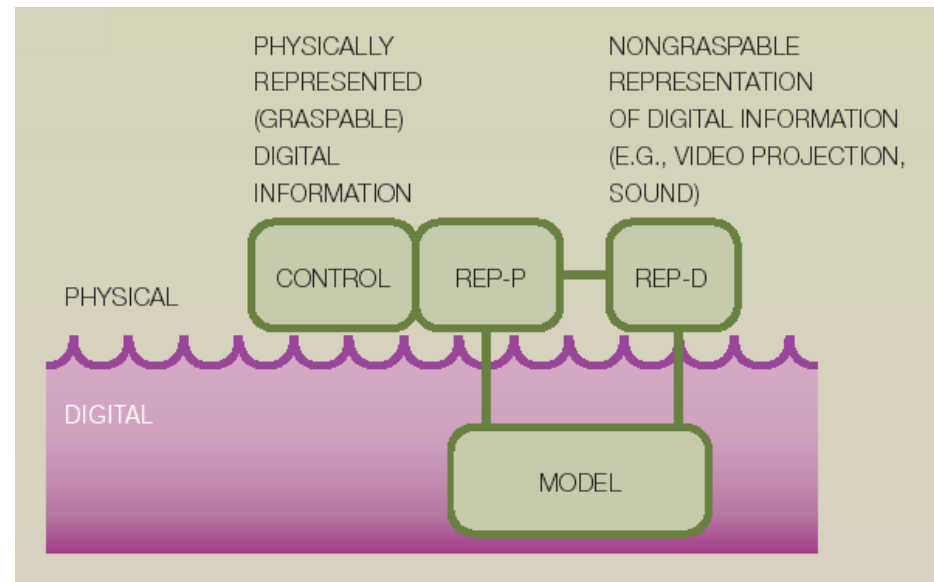Background information displays using ambient media

drawing: Hiroshi Ishii

Tangible interaction model

From Ullmer & Ishii [2000] "Emerging Frameworks for Tangible Interaction"

GUI interaction model
Model-View-Controller

TUI interaction model
Model-Control-Representation (physical / digital)

Tangible systems taxonomy

[Ullmer & Ishii 2000] categorization of tangible systems:

Spatial systems

Spatial configuration of objects in a reference frame (e.g. tabletop) is computationally interpreted and augmented

Constructive systems

Modular electronically instrumented artifacts are used for constructing computationally interpreted physical structures

Relational systems

Logical relationships between tokens are mapped onto abstract computational interpretations

Associative systems

Associate individual physical artifacts with digital information

We focus on spatial systems: i.e. tabletops with tangible interaction…

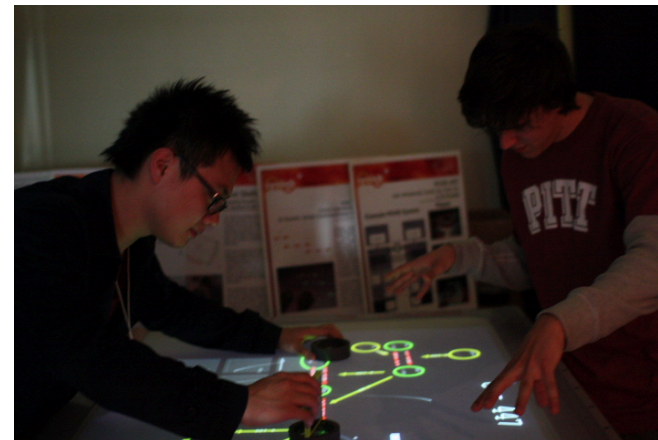# Tangible Tabletops – examples

Tangible Anchoring
Robinson et al. [2014]

Reactable
Jorda et al. [2007]

TViews Table Role Playing Game
Mazalek et al. [2008]

Pathways
Wu et al. [2011]

Tangible Tabletop Sensing

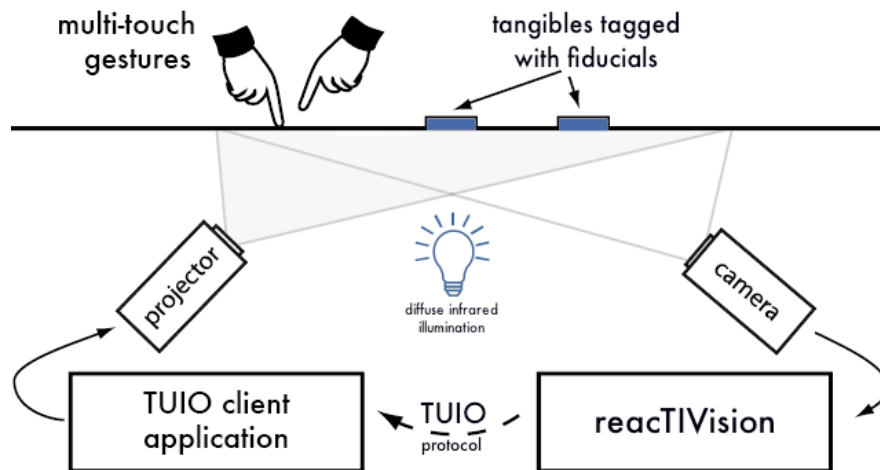Common sensing approaches include

Imaging (e.g. ReacTIVision, MS PixelSense)

Acoustic (e.g. TViews)

Electromagnetic (e.g. Sensetable, Tangible Viewpoints)

We use the reacTIVision computer vision based framework
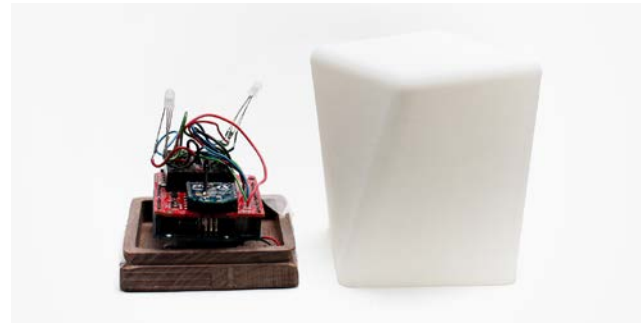
[Kaltenbrunner 2009]

Active Tangibles

Programmable physical objects with integrated display, sensing, or actuation technologies

Can be reconfigured over time, allowing users to modify their associations.
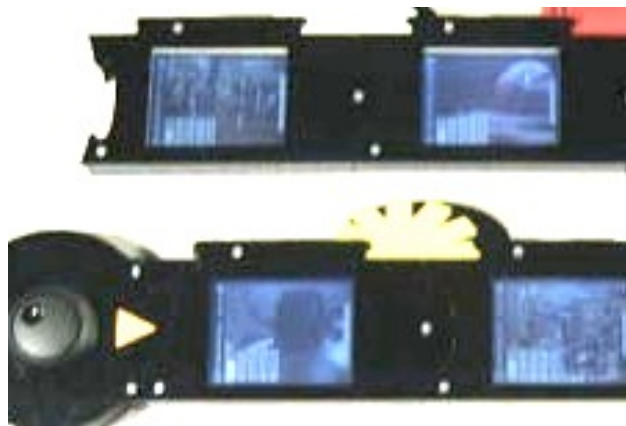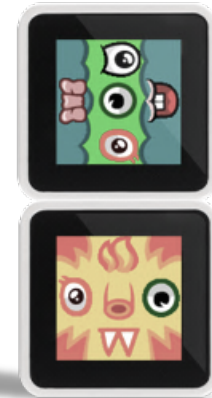
Manipulated in-air, on-surface, on-bezel

Active Tangibles – examples

Sifteos

Tangible Query Interfaces
Ullmer et al. [2003]

Tangible Video Editor
Zigelbaum et al. [2007]

SynFlo
Xu et al. [2013]

Mobile Devices as Active Tangibles?

Mobile-based Active Tangibles

Tangible Anchoring's "Changeable Tangible"
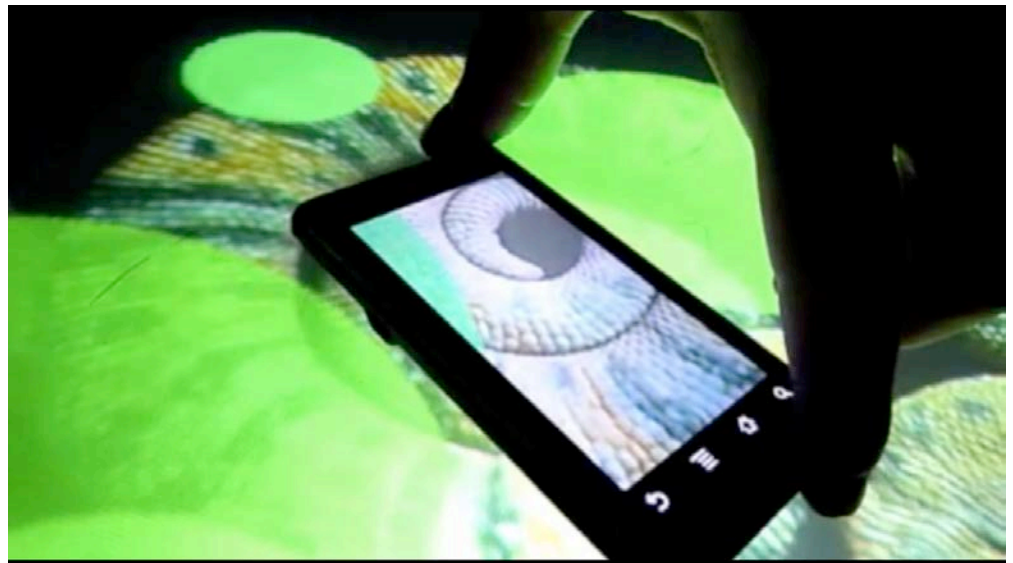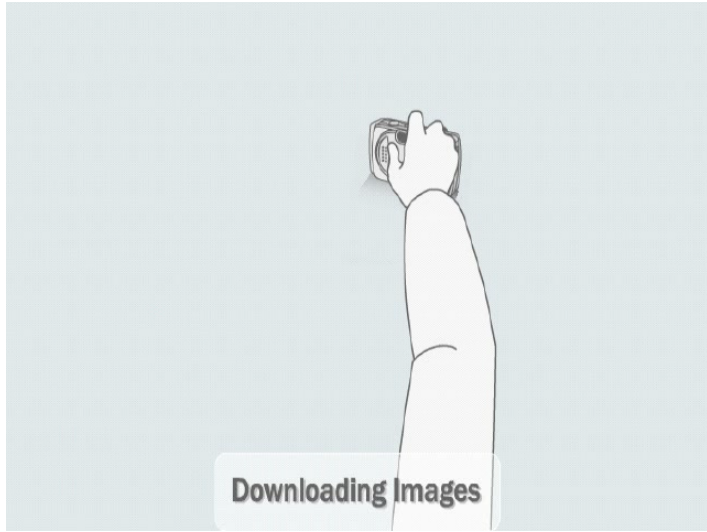
Robinson et al. [2010]

ROSS: Responsive Objects, Surfaces, and Spaces
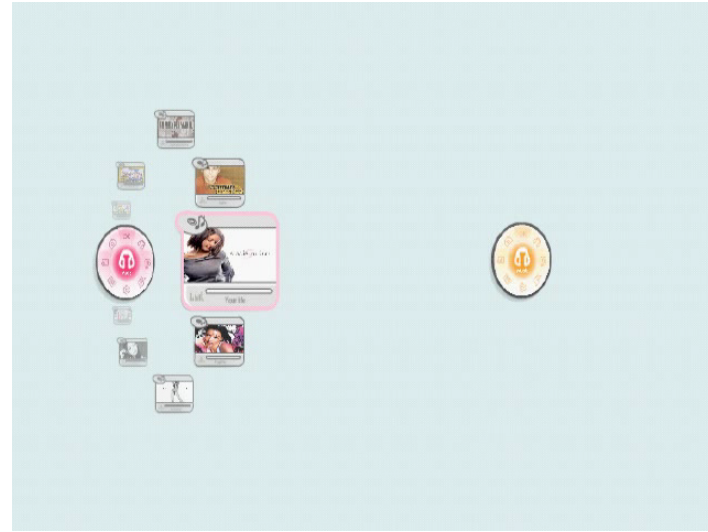
Toolkit for cross-device development
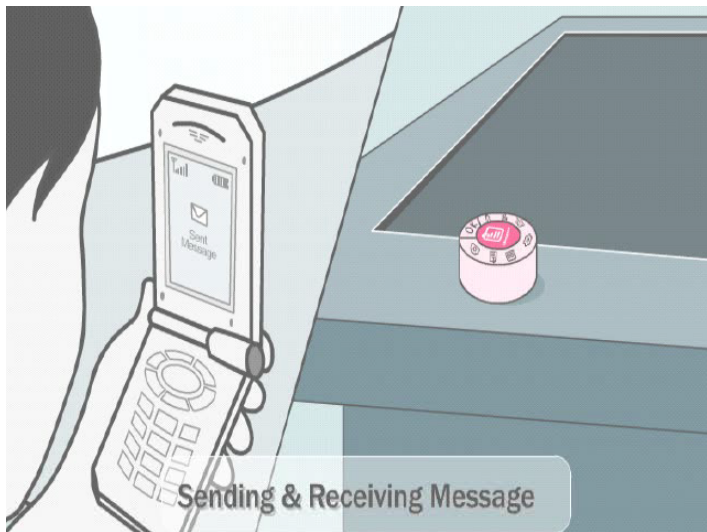
Wu et al. [2012]

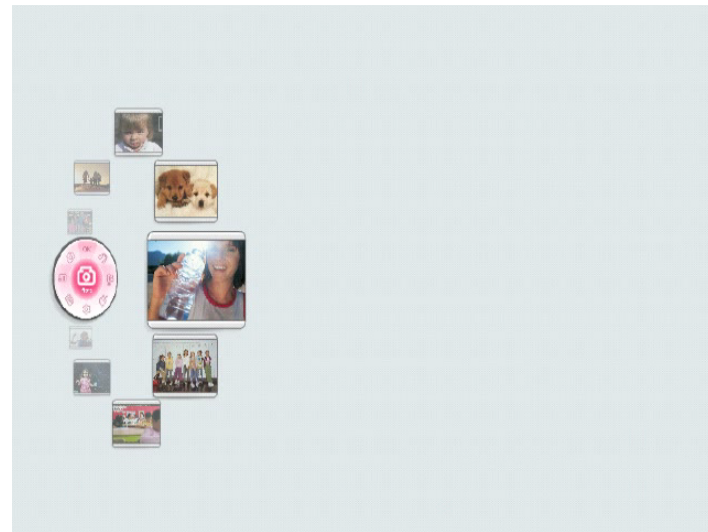# Interaction Scenarios

Downloading images from digital camera

Listening to common music playlists

Sending and receiving messages

Building slideshows with images and music

Synlab's Interactive Table
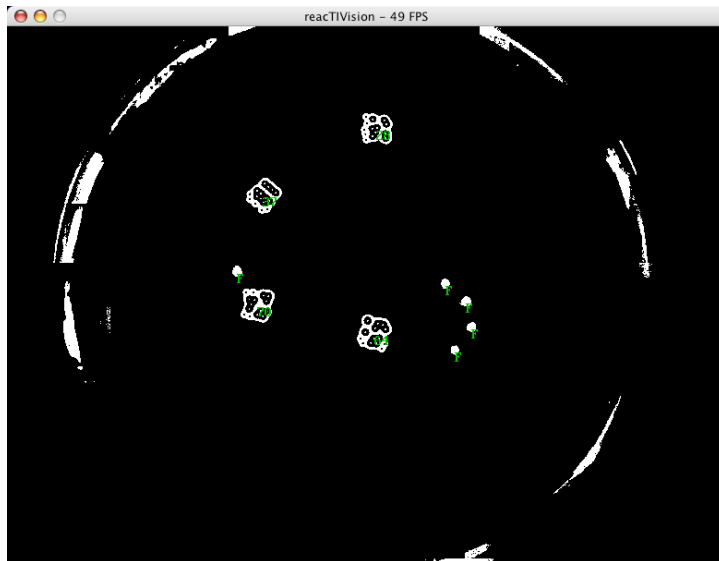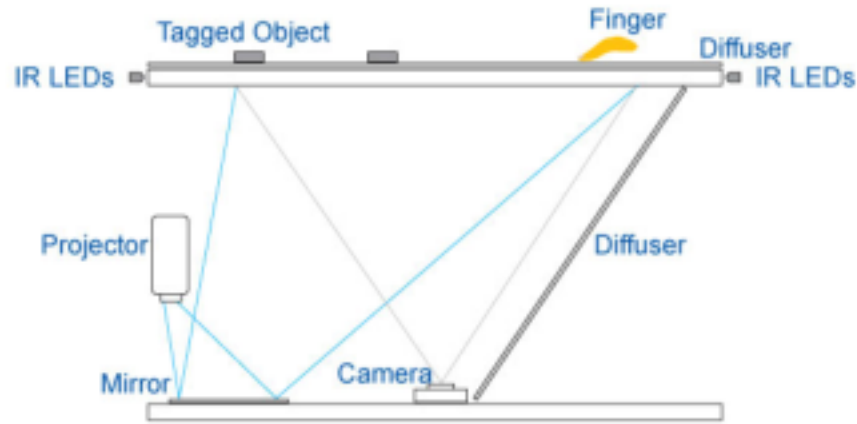
Simulated Tabletop Setup
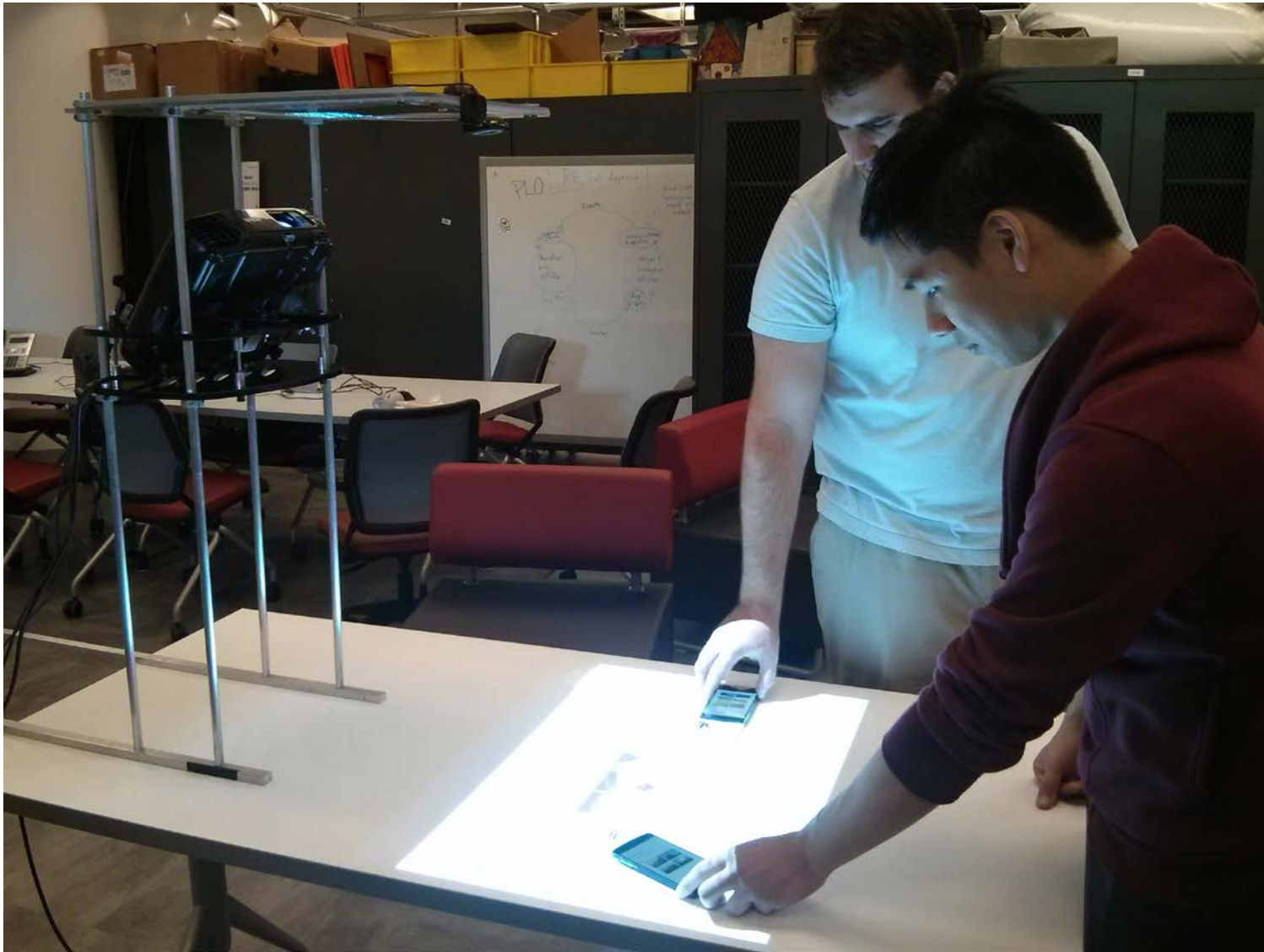
Live Demo – photo sharing scenario
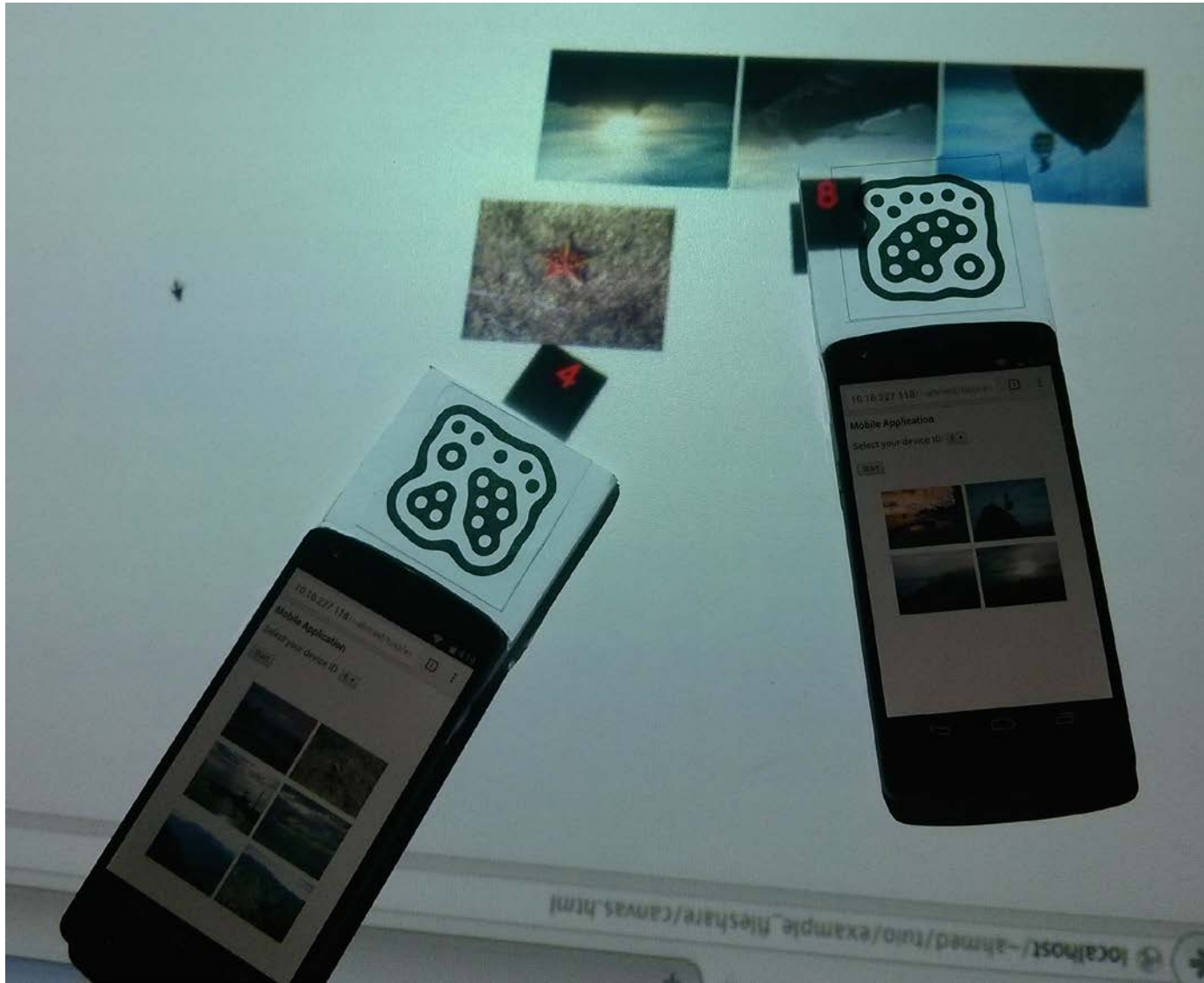
Prepare Your Laptop

See: http://www.asarif.com/tutorials/mobiletangibles/

Have you all done this?

Now let's form teams…

Each team needs:

Laptop with necessary libraries and host/client installed

Example code

http://www.asarif.com/tutorials/mobiletangibles/tuio.zip

2 x Google Nexus5 phones with tags

Prototyping materials (construction paper, markers, etc.)

Getting Started

See: tuio/example_fileshare/readme.txt

1. Find out your localhost ip using the following command:

    ```
    $ ifconfig |grep inet
    ```

2. Update the ip inside the info.xml's <localhost> tag

3. Start the reacTIVision.app

4. Navigate to the example folder

    ```
    $ cd /tuio/example_fileshare
    ```

5. Start the mobile server using the following command:

    ```
    $ node mobileServer.js
    ```

6. Start the tuio.js server using the following command:

    ```
    $ node src/server.js
    ```

canvas.html

JavaScripts

jquery-1.7.2.js

socket.io.js

lodash.js

Tuio.min.js

TuioCanvas.js

} Libraries

requestanimationframe.js

loadXML.js

CSSs

example.css

mobileStyle.css

} Style sheets

canvas.html

Loads Tuio canvas

Function to allow drag and drop behavior

Function to load images from the mobile app

Connect to the socket.io server

Listen to messages (file URIs) and load the images on to the canvas

Function to override default drop behavior

Simulates copy-paste

TuioCanvas.js

Loads a Tuio canvas

Connects to the Tuio host server

Identifies a tangible object based on markers

  Draws an object on the canvas when connected

  Dynamically changes object position, orientation based on the markers

  Handles the image files loaded by the mobile.html

mobile.html

JavaScripts

mobileScript.js

mobileServer.js

loadXML.js

CSSs

mobileStyle.css

mobile.html

Programmatically changes localhost address using info.xml

Loads images from a device folder (a folder in the network in this case) based on user selection

mobileScript.js

Sends messages (image URLs) to all connected hosts

First Exercise

In the musical playlist scenario above, when two devices are moved close to each other on the table, the intersection of their playlists is shown on the table's surface.

Your Task

Implement similar functionality for the photo scenario, so that if two phones are moved close together, only the photos they share in common are automatically shown on the table canvas.

You can work in the /tuio/example_commonpictures/ directory. To get you started, the directory contains a copy of the code from /tuio/example_fileshare/.

Design and Development Activity

We've given several scenarios of how mobile devices and interactive tabletops can be used together.

Your Task:

Come up with a new scenario of your own choice. Sketch out your scenario on paper. Then you can either:

1) Create a low-fidelity mockup of your scenario using the available materials (construction paper, post-its, scissors, glue, markers).

OR

2) Do a first (quick and dirty) implementation of your scenario using the previous source code example as a starting point.

Presentation & Discussion