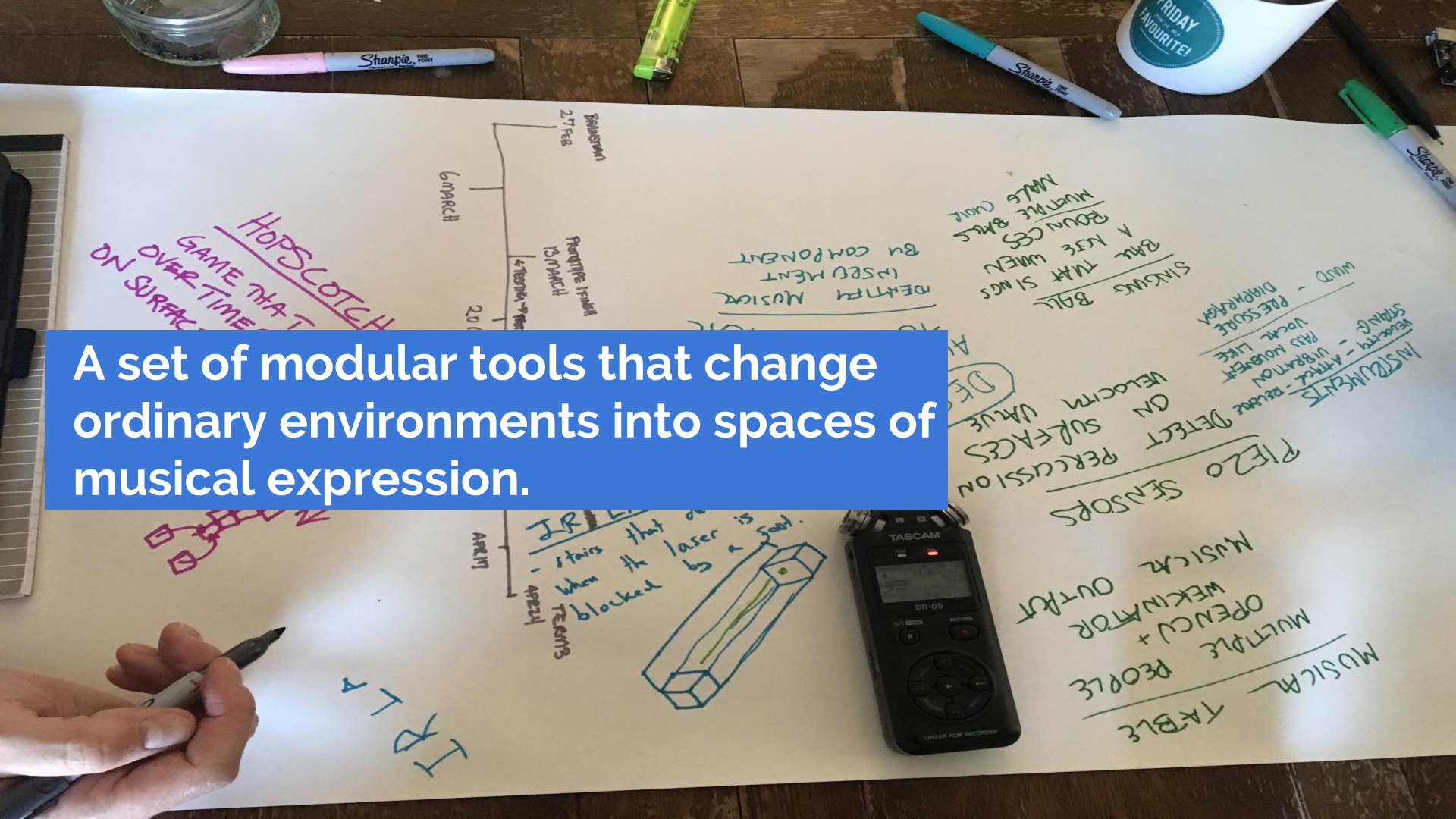
Musical Playground

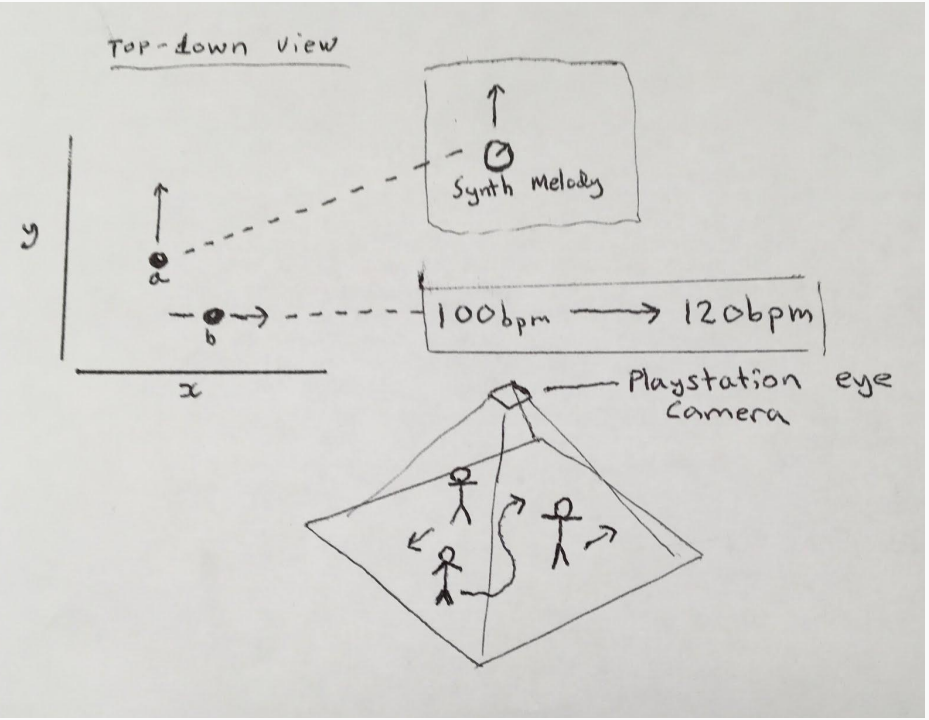
Alex Fletcher, Ben Tandy, Billy Karnchanapee, Matt Deline

27.03.2018

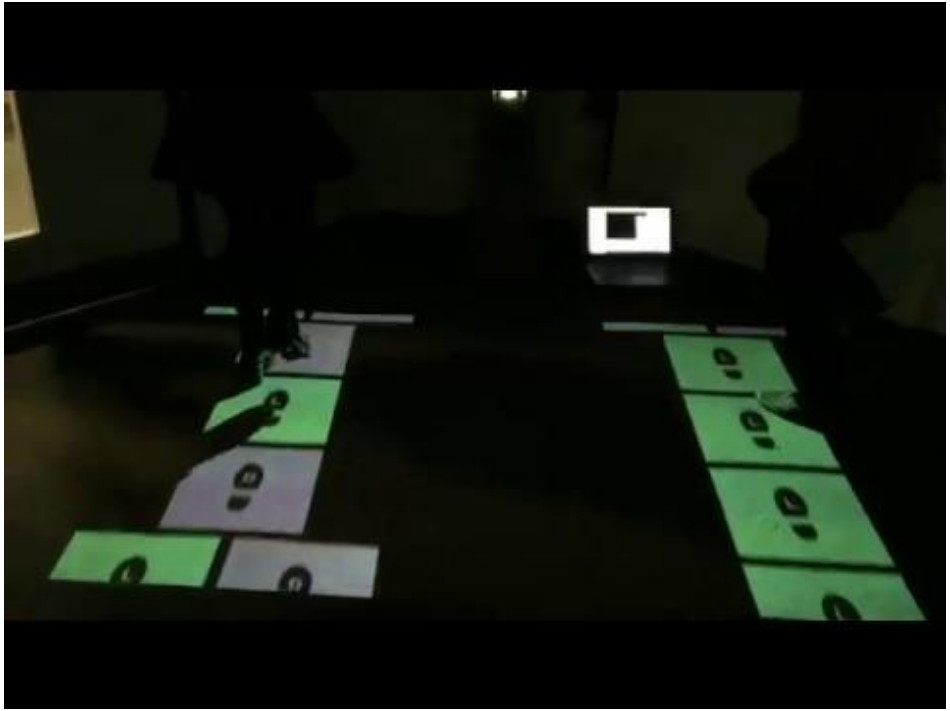
What is it?

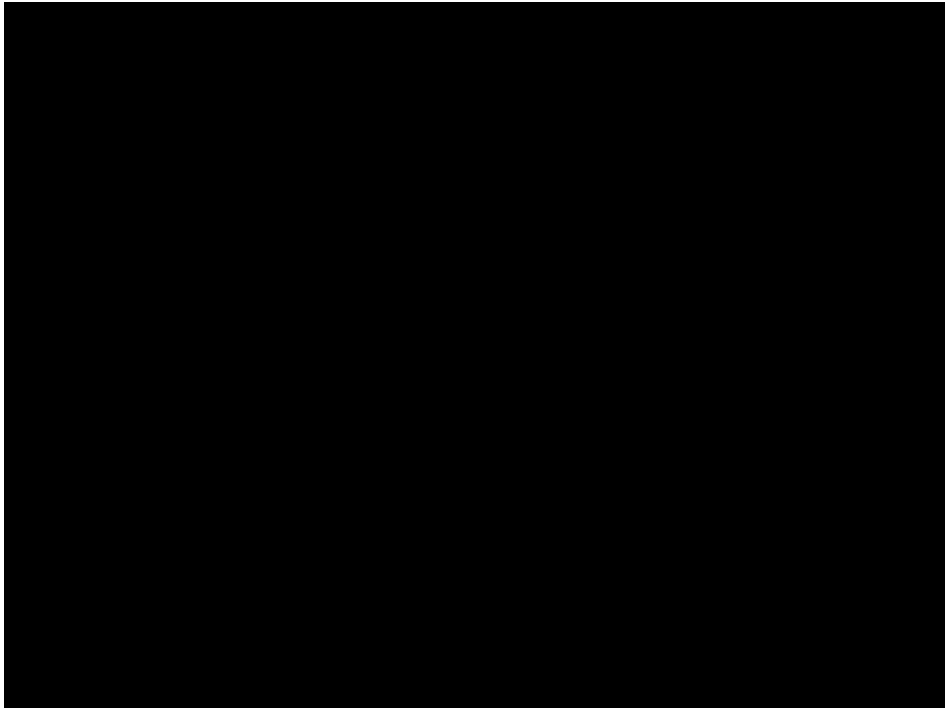

A set of modular tools that change ordinary environments into spaces of musical expression.

What does the experience look like?



With these tools we can create different experiences





Development Team

Alex Fletcher

Audio Design, Synthesizer Programming in Helm, Interaction Design, Musical Playspace Prototype

Benjamin Tandy

Programming, Tool Development, Hardware Development, Hopscotch Game Prototype

Billy Karnchanapee

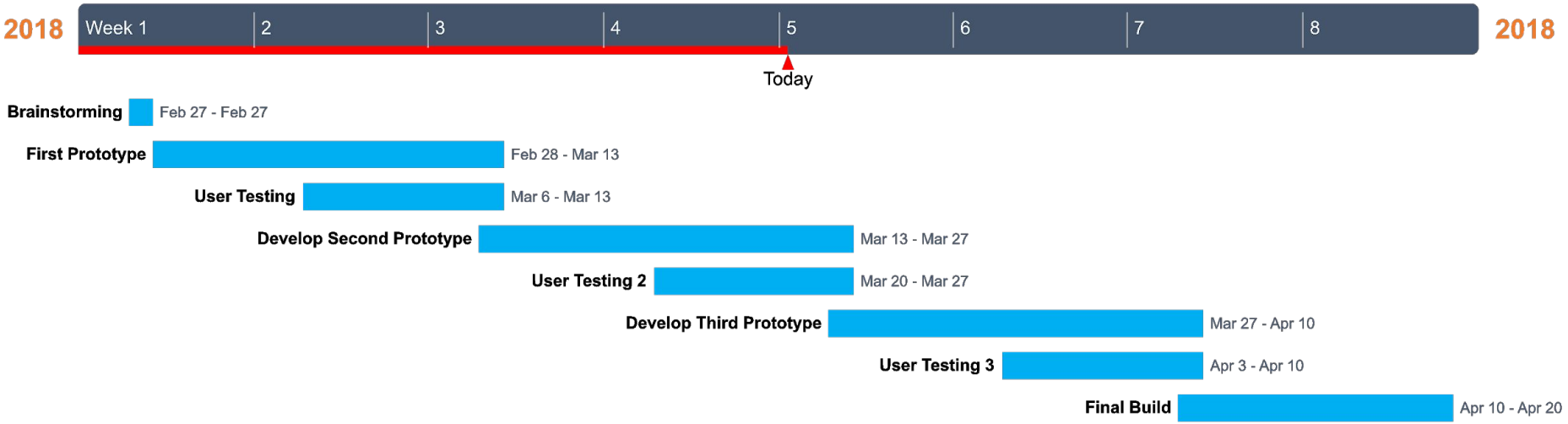
Programming, Audio Implementation, Prototype Development in Unity, Computer Vision Calibration

Matthew Deline

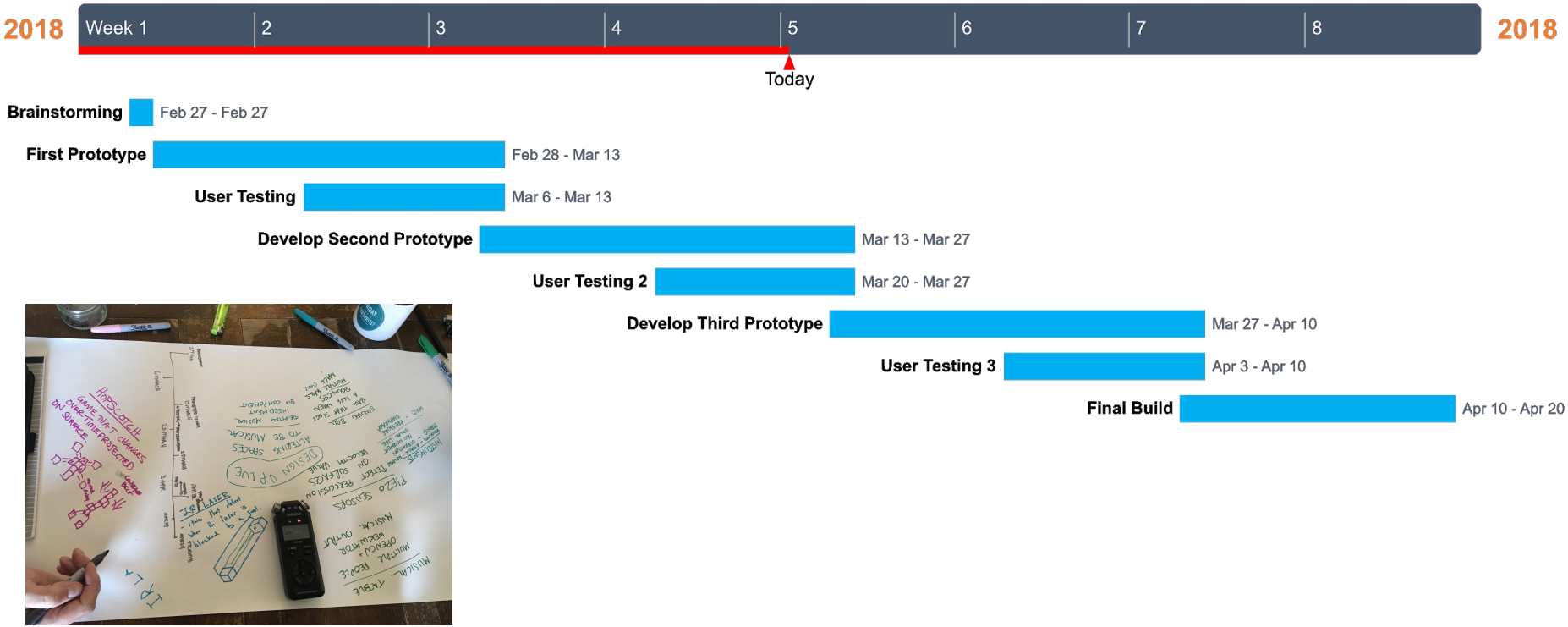
Project Management, Research, Shape Game Prototype, and Tool Development

Prototypes and Playtesting

Development Timeline



Development Timeline



Development Timeline

Brainstorming Feb 27 - Feb 27

First Prototype Feb 28 - Mar 13

User Testing Mar 6 - Mar 13

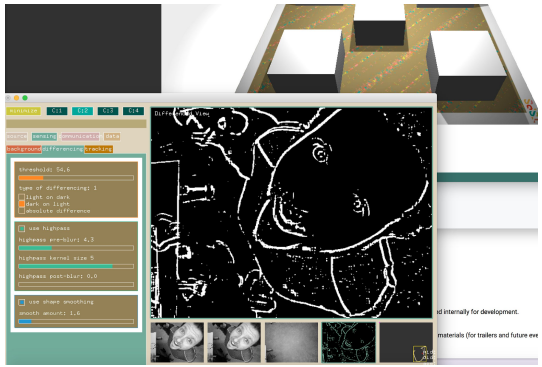
Develop Second Prototype Mar 13 - Mar 27

User Testing 2 Mar 20 - Mar 27

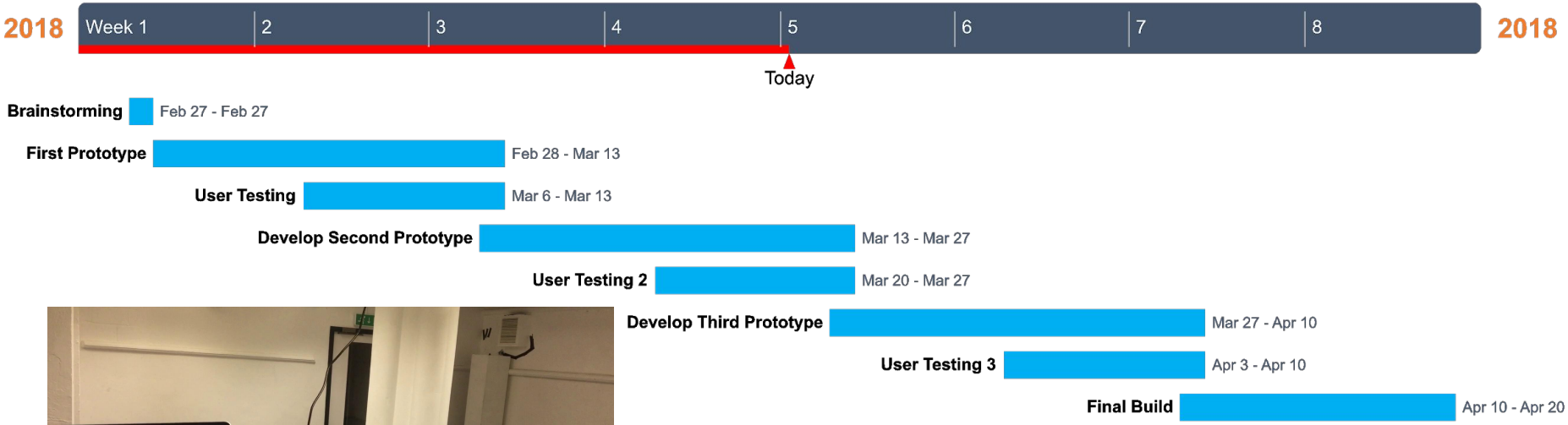
Develop Third Prototype Mar 27 - Apr 10

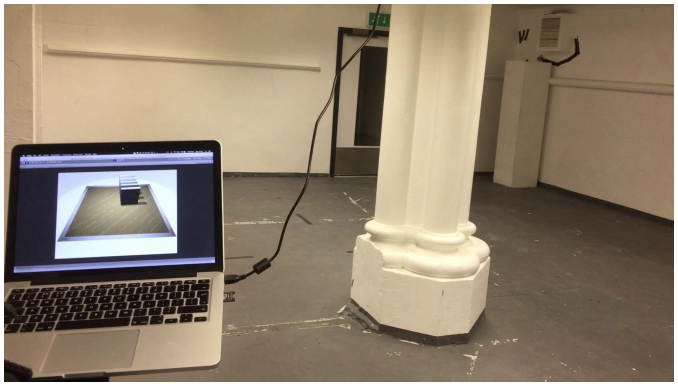
User Testing 3 Apr 3 - Apr 10

Final Build Apr 10 - Apr 20

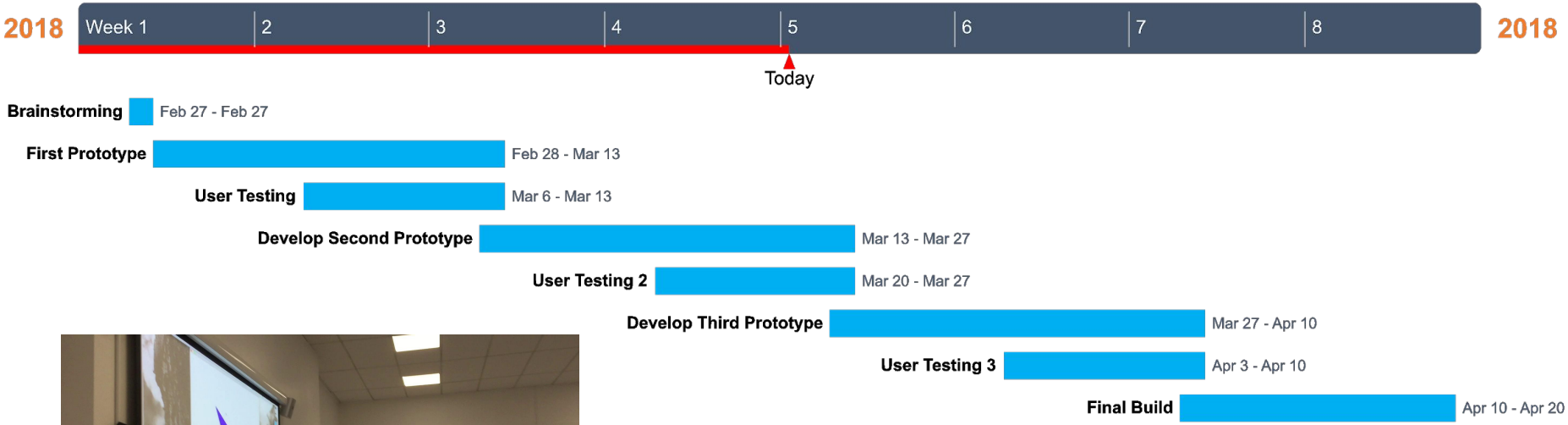


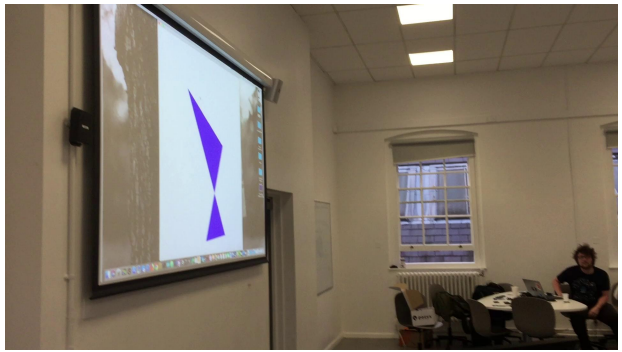
Development Timeline



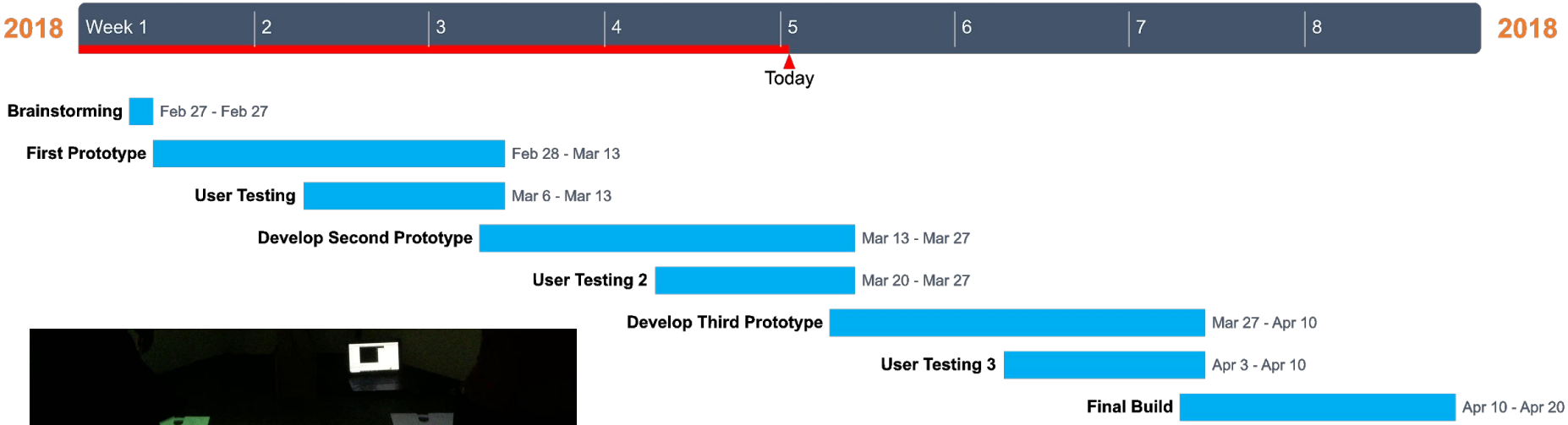


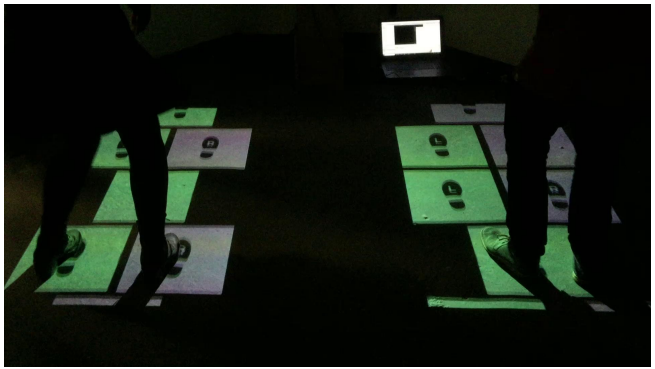
Development Timeline



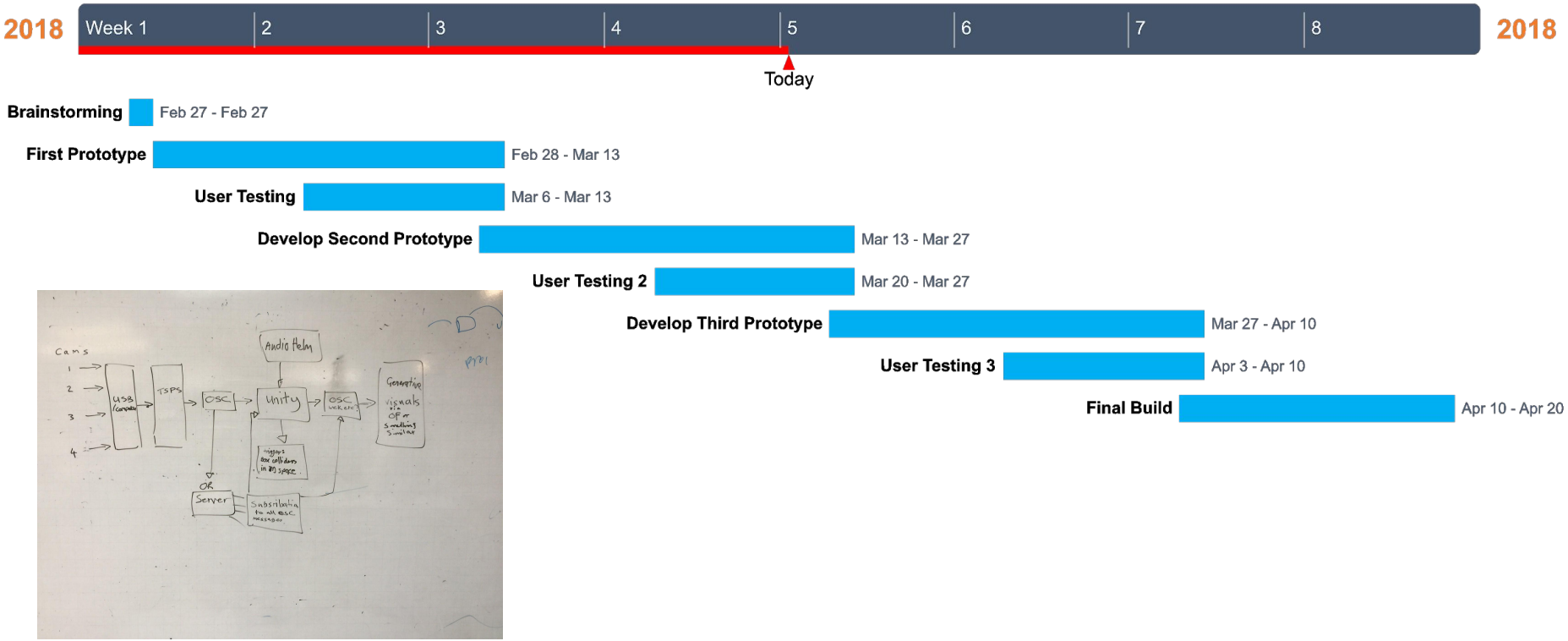


Development Timeline

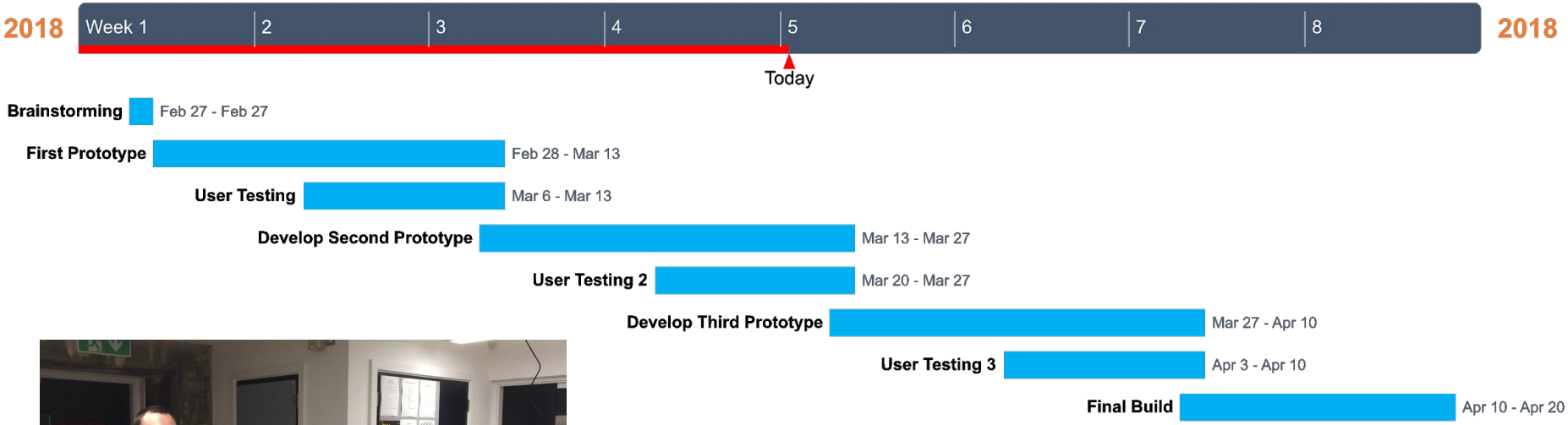




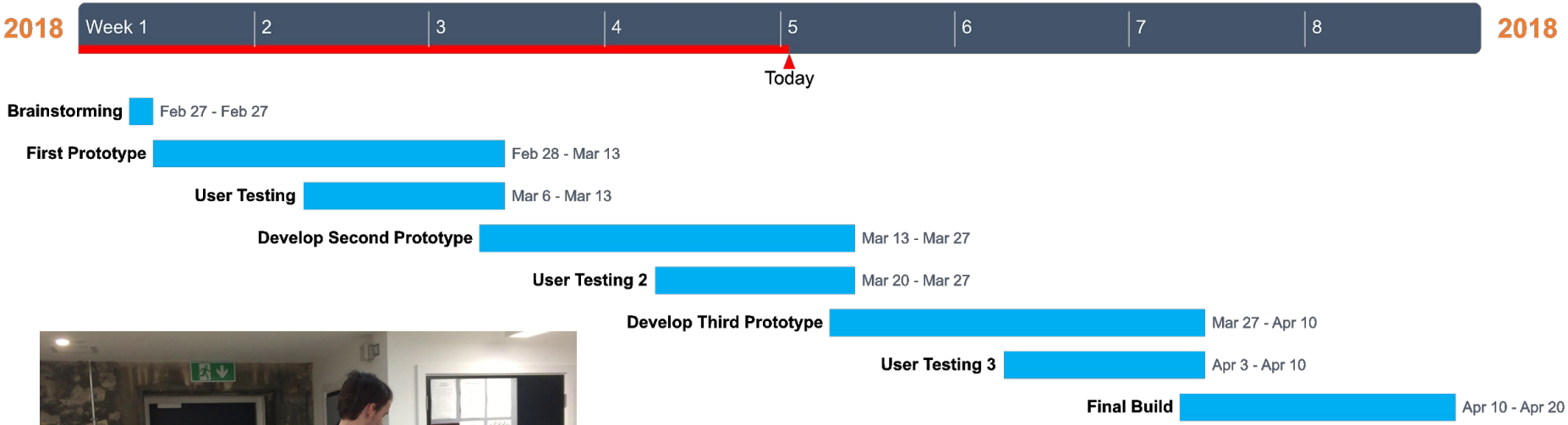
Development Timeline



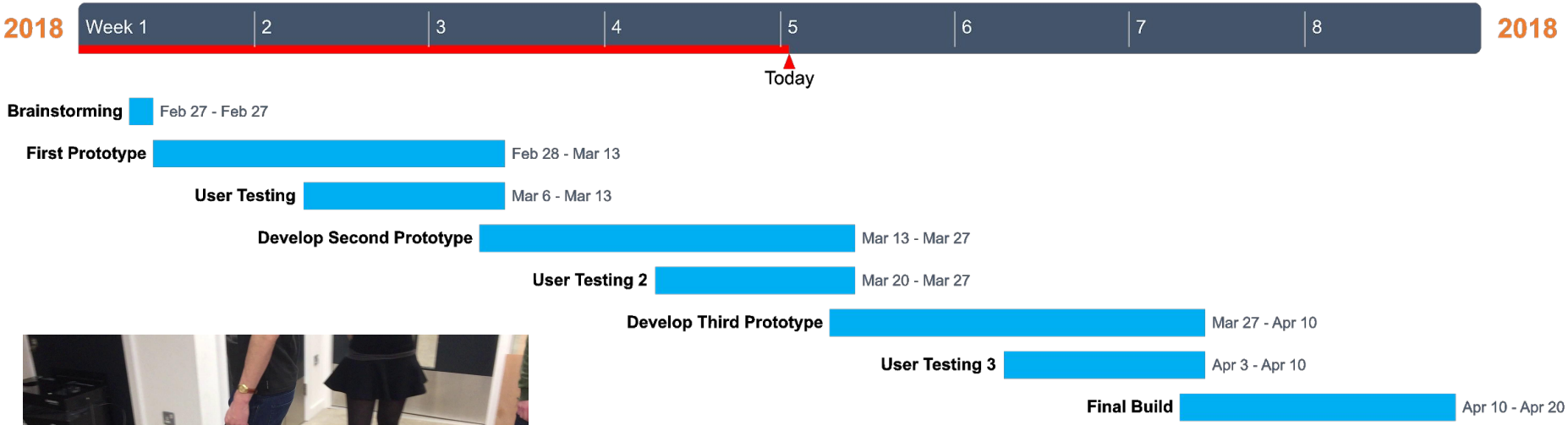
Development Timeline



Development Timeline



Development Timeline



Development Timeline

Brainstorming Feb 27 - Feb 27

First Prototype Feb 28 - Mar 13

User Testing Mar 6 - Mar 13

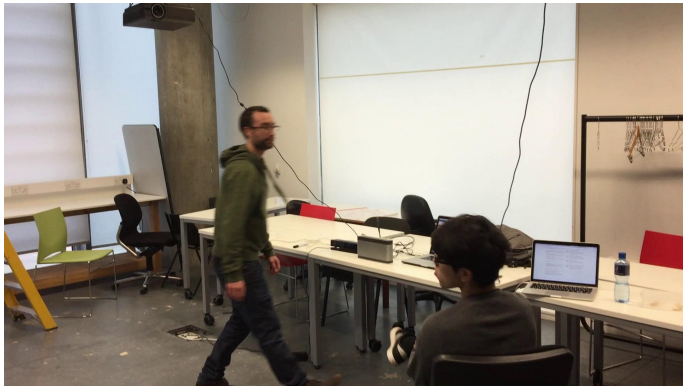
Develop Second Prototype Mar 13 - Mar 27

User Testing 2 Mar 20 - Mar 27

Develop Third Prototype Mar 27 - Apr 10

User Testing 3 Apr 3 - Apr 10

Final Build Apr 10 - Apr 20



Development Process

Initial Solutions

OpenTSPS Tracking with Camera

- Apple Mac running OS X
- Multiple PS3 Eye cams / Webcams
- Running an instance of openTSPS
- Passing the data into Unity via OSC

Initial Solutions

OpenTSPS Tracking with Camera

PROS

- Can run multiple cameras on one computer
- Good performance
- Easy to setup and integrate directly to Unity

CONS

- Requires a more powerful PC
- Makes placement harder as requires cameras with USB extenders
- Means the openTSPS software must be running to work

Initial Solutions for tracking

RPi OpenCV Blob Tracking

- Raspberry Pi 3+
- Picam or Webcam
- Running OpenFrameworks and OpenCV
- Using Python to run blob tracking and pass the data via Serial over network to a main PC unit

Initial Solutions

RPi OpenCV Blob Tracking

PROS

- Cheap to duplicate units and cover larger areas
- Potential for wireless connectivity which would allow for easier placement
- Low power / Can run off battery potentially

CONS

- Potential issues with struggling performance
- Wireless connectivity could lead to latency issues
- Installation of OF and OpenCV is time consuming
- Would have to write blob tracking script
- Harder to debug

Software and Hardware Setup

Hardware

- Projectors
  - One pointed at the floor
  - One pointed at the far wall
- Cameras mounted above the space facing down
  - Up to 4 units to cover the space
  - Lenses adjusted to reduce FOV
- High End PC driving Audio and Visuals
- Apple Mac running multi-camera openTSPS instance

Software and Hardware Setup

Software

- openTSPS Instance handling tracking
- Audio and visuals driven in Unity
- Written in C#
- Uses shaders and procedural mesh generation
- Uses Helm as programmable dynamic synthesizer for Audio

Sound Design with Helm

Why Audio Helm?

“Make music in Unity with a full-featured native synthesizer, MIDI sequencer and sampler. You can make a musical game, create generative music and warp dynamic sound effects using **Audio Helm**.”

- Other digital audio workstations for Unity, like FMod and Wwise have limited functionality with their native synthesisers.
- They are more designed for sound design within a traditional game environment.
- As a musician Audio Helm is designed like any of the software instrument/analog synthesisers i am use to using.

Initial Design Approach

Main design value: Giving players agency to improvise music in space.

1st Iteration: Synth instrument in audio helm to trigger musical notes on collisions.

- Created 3 versions:
  - 5 note synth instrument
  - Drum sequencer
  - Melody that plays when you follow a path in the playspace.

Feedback:

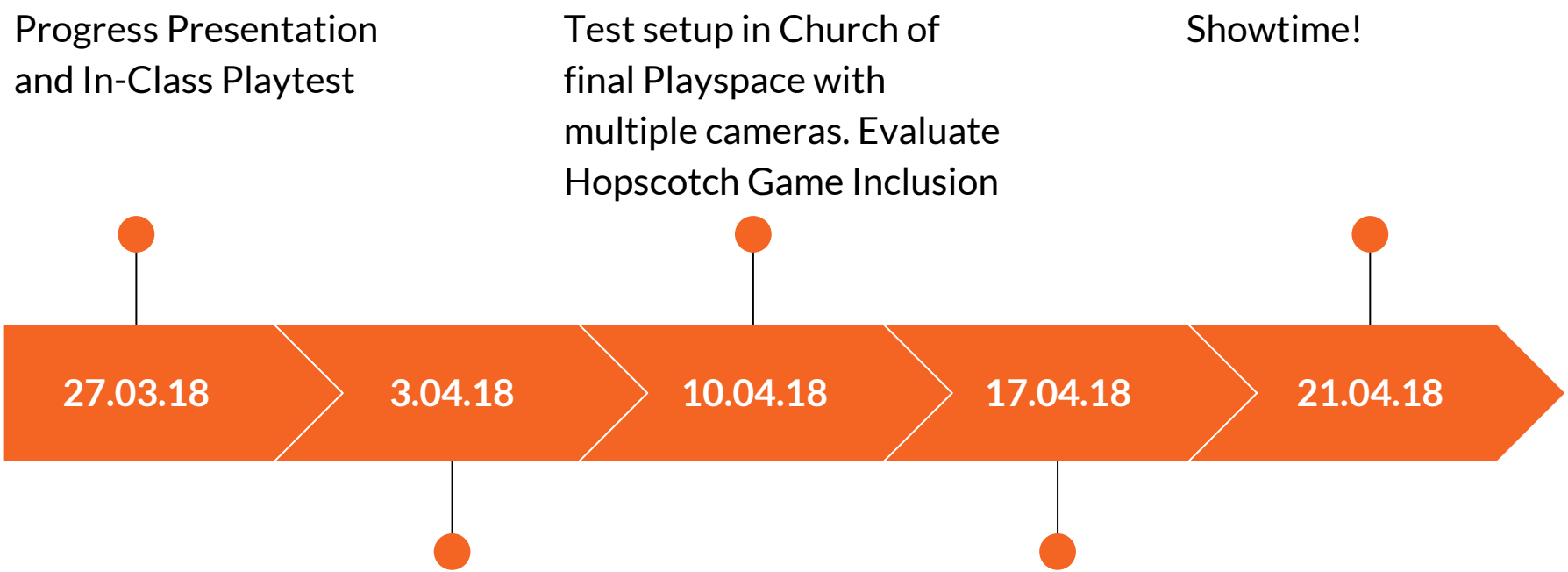
- Non-musicians were confused on what they were doing.
- Musicians understood what they were doing, but they wanted more freedom to explore other ways of manipulating the sound. For example changing the timbre of the sound as you move through the space.

Maturing The Design: 2nd Iteration

New design value: Give more freedom for musical expression for all

- Design a soundscape composition, that can be performed in many different ways.
- Assign an instrument or the bpm from the composition, to a player in the playspace.
- Affect player's instrument sound in interesting ways, mapped to their position and velocity in the playspace.
- Constrain the nature of synthesis manipulation so that there is no learning curve for non-musicians to participate.

Where do we go from here?


Progress Presentation and In-Class Playtest
Test setup in Church of final Playspace with multiple cameras. Evaluate Hopscotch Game Inclusion
Showtime!
27.03.18
3.04.18
10.04.18
17.04.18
21.04.18
Integrate Shape Game with Musical Playspace Prototype Using Shaders
Finalize Code and prepare for setup in Exhibition

Questions?