PROJECT OVERVIEW

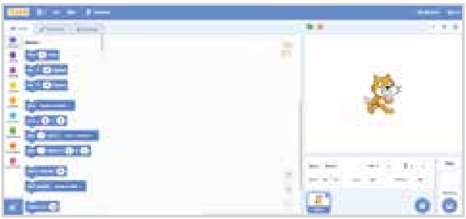
SCRATCH VIDEO SENSING

Video sensing is a new extension added to the Scratch 3.0 programming environment. It uses a video camera as a sensor to detect motion and color. When used with Scratch, it allows learners to control digital animations with physical interactions using moving objects or specific colors seen through a camera. Video sensing allows us to see virtual characters overlaid on the real world context, which encourages learners to create imaginative stories or interactive games. Programming interactive animations blurs the line between the physical and digital worlds; this proved a rich context for computational tinkering.

This document is a summary of the experiments and findings of our research and development of Scratch video sensing activities. We hope to inspire discussions and provide project ideas and possible directions for future development in this area.

Materials, tools, and hardware

Mac



Scratch 3.0
(online programming environment)

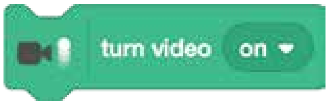
Webcam
(We used HD Pro Webcam C920)

the
tinkering
studio

exploratorium

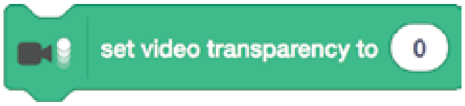
Video sensing blocks

These blocks allow learners to interact with their Scratch project using a webcam. You will find them in the **Video Sensing** category in Scratch.



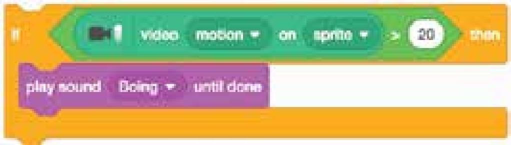
```
turn video on ▾
```

Turns on the webcam, which enables you to use video sensing blocks.



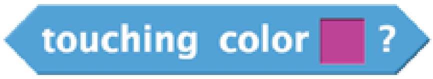
```
set video transparency to 0
```

Sets the transparency of the background on the scale from 0 to 100. 0 means full transparency, 100 means a complete white background.



```
if video motion ▾ on sprite ▾ > 20 then
    play sound Doing ▾ until done
```

A variable block. The video sensing can detect the amount of the motion on a specific sprite or the entire background.



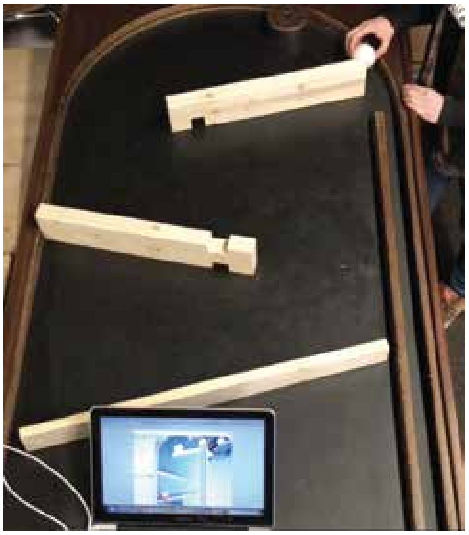
```
touching color ?
```

A color sensing block. The video sensing can detect a specific color that you chose in the color window.

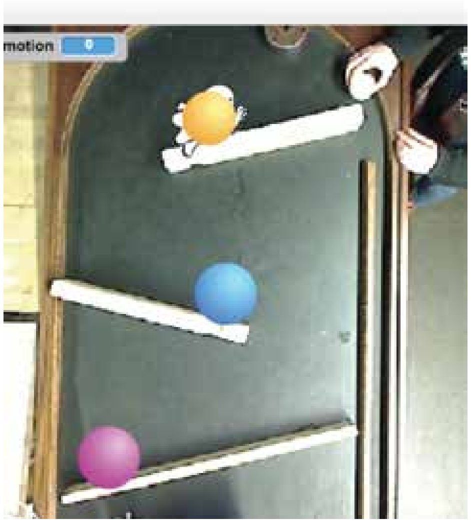
Early prototype

Our first experiment started with an idea that someone on the team asked "Could we create a Marble Machine and add a layer of augmented reality to play music, or trigger, virtual characters to pop-up as the ball rolls down the track?" Our team was curious to try this out with the Scratch video sensing.

We created a physical ball run and had a webcam capture the ball run stage, then we superimposed several virtual colored circles on the Scratch screen.

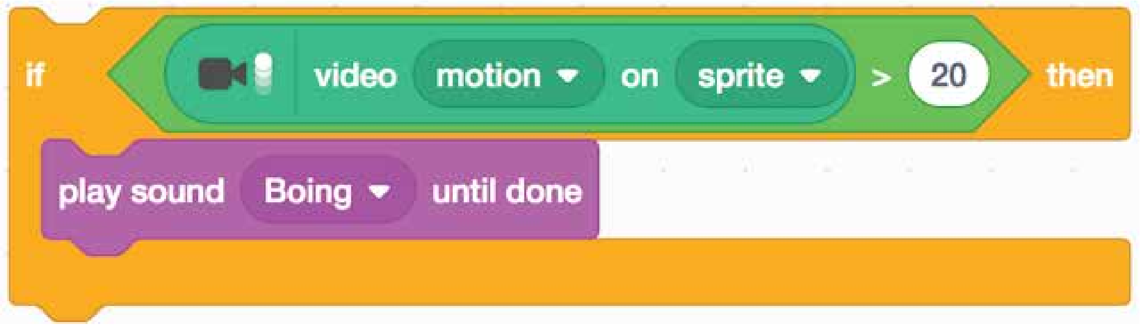


Physical Ball Run



The Ball Run as seen in the Scratch environment

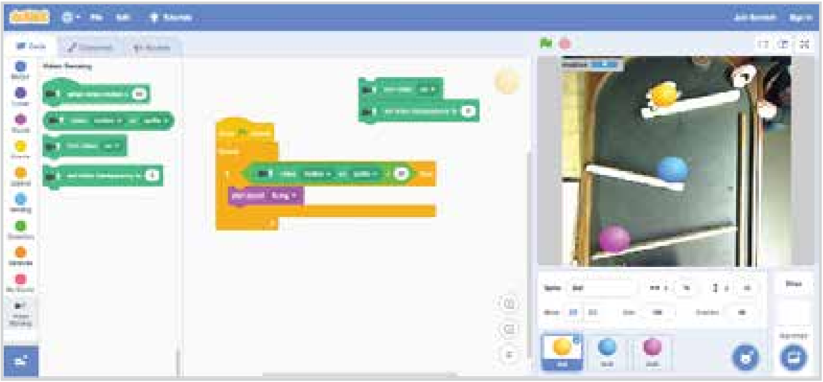
The code we used:



```
if video motion on sprite > 20 then
    play sound Boing until done
```

The code means
“If the video motion on this sprite is greater than 20, THEN play sound “boing””

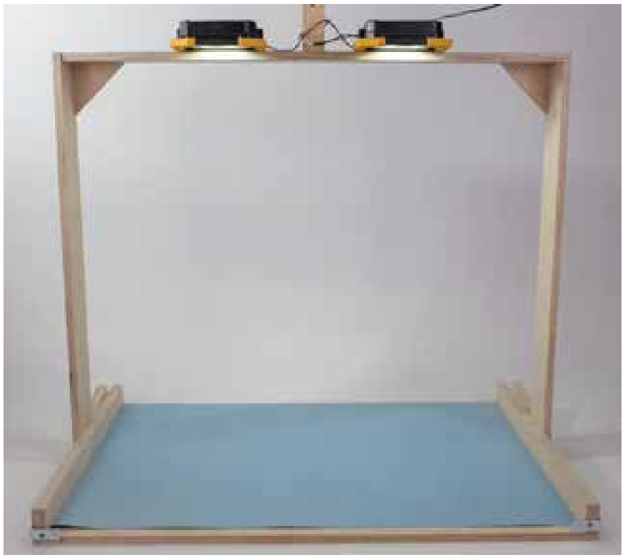
We assigned code to the virtual colored circles on the ball run stage. When the physical ball rolled over the circles, the video sensing detected the amount of motion. If the amount was greater than 20, it triggered a digital character (a ghost) to appear on the screen.



What we liked (What felt like tinkering to us)

- *The ability to have the virtual and physical object inhabit the same space felt creative, which encouraged our interactive visual story making. The possibility of what we could do with the video sensing technology seemed to be endless.*
- *There were two types of tinkering—one with physical and another with digital contents—and we saw those two were amazingly intertwined. Compared to other coding activities such as micro:bit or Digital Light Play, video sensing really merged the physical and digital environments into one space through a camera, and let us approach the two realms (physical and digital) more synthetically and harmoniously.*
- *We liked the endless feedback loop: moving physical objects, adjusting digital characters, and modifying code. The continuous process naturally invited collaboration and encouraged working with others.*
- *We enjoyed the performing aspect. Unlike other Scratch projects, a video sensing project requires you to perform each time you run the program. Depending on the performance (i.e. where the ball runs), your project might go in a different direction than you anticipated, but we actually enjoyed the improvisation and felt that process was tinkerable and generative.*

Prototyping Video Sensing in the Tinkering Studio



Set-up:

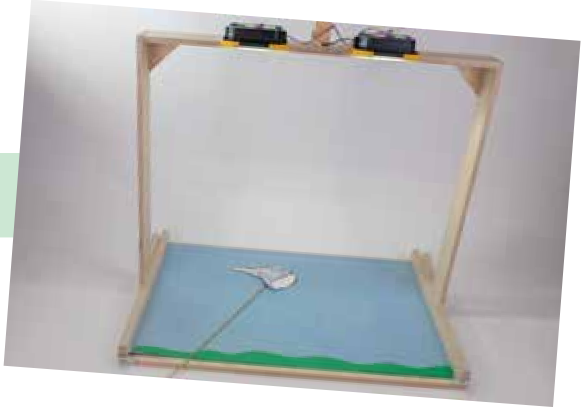
We created a couple of overhead camera platforms with lights and called them video sensing stations as a stage to capture learner's animation. This set-up helped to create an inviting atmosphere in the space. This also helped the tinkering space look more like a creative studio rather than a computer classroom.

Scaffolding an introduction to Scratch video Sensing:

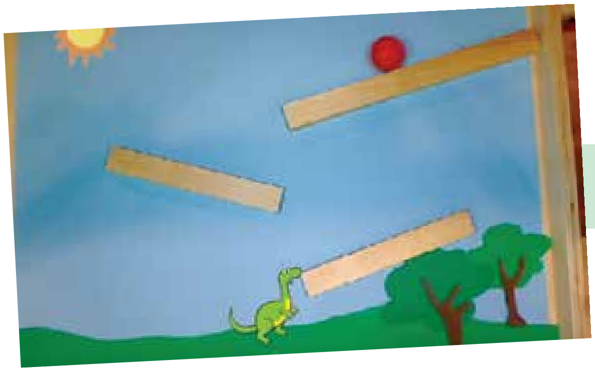
One of our big challenges was how to get learners engaged meaningfully within a short timeframe. Although Scratch is widely known, few people had experience using the video sensing capabilities of Scratch 3.0.

We started with a couple of simple program examples running and let learners play with the set-up:

Underwater scene with fish. The animation changed "When touching blue" and "When touching yellow"



Capturing virtual butterflies with a butterfly net "When touching net color"



Ball run animation was triggered "If the motion is greater than 20 then..." or "When touching ball color"

We first encouraged learners to play with the programs as they were set up, and while they played we pointed out the various parts, such as the webcam, the virtual sprites, and the code that was animating the characters. These simple starting points helped to spark learners' initial interest and often resulted in longer interactions.

The transfer of ownership

We found important shifts of ownership—from playing with the examples to forming their own ideas—could happen in a couple of ways:

1 Choosing your own sprite

We found learners' interest immediately heightened when we offered the collection of sprites and told them that they could switch sprites to their favorite one.

2 Making a small adjustment to the code

Sometimes making the smallest change in the code was enough to shift the ownership. It might be changing the number of steps or sizes of the sprites when responding to the video feed.

3 Using your own body as a physical input

Instead of the physical objects that we prepared, we often let learners use their own hands.

4 Creating your own character

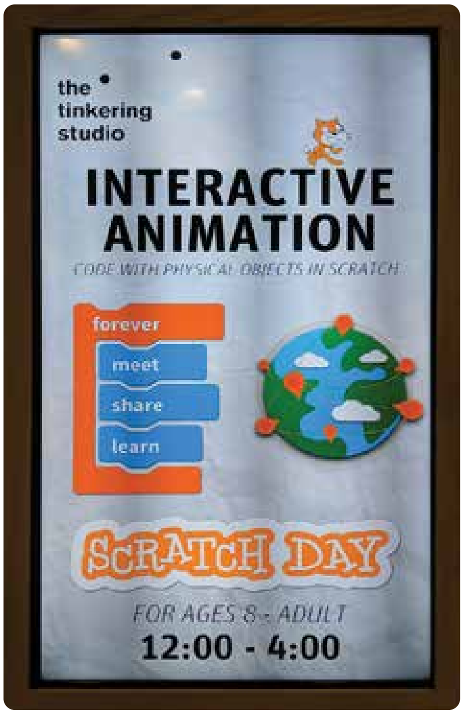
Learners were more engaged when they created characters by themselves We prepared a table with craft materials and tools (scissors and tape) to let learners create their own characters. Once they created a character, we took a photo of it to import as a digital sprite in Scratch. As soon as learners saw the character that they created in physical form get transported into the virtual world, they were immediately engaged.

Facilitation

The Scratch video sensing activity is still in the prototype stage and we are experimenting in many ways with how to engage learners in this activity. We discovered the following findings about facilitating this activity:

What we called it

We called the activity "Interactive Animation with Scratch", a simple way to help it sound more familiar to learners instead of calling it "Video sensing with Scratch." It was important to minimize the amount of new concepts and tools that learners had to know at the beginning, and keep it low threshold so that learners could recognize quickly what the activity was about.



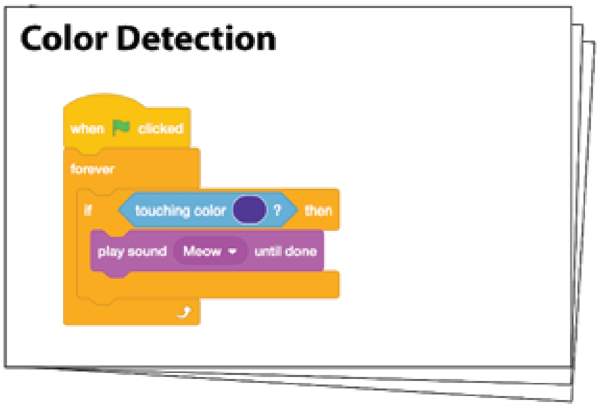
Tinkering Studio entry sign

Creating a microworld

Scratch is a versatile and open-ended platform, supporting learners to make anything they want. This was problematic because learners might easily deviate from their video sensing project at any time. While introducing (and throughout most of the entire activity) we tried to stay with the learner, guiding and curating the learner's experience focusing on video sensing. The important role of the facilitator here was to navigate their journey by identifying a core set of useful blocks for the video sensing project. This facilitating experience led us to create a video sensing "microworld" in collaboration with the Lifelong Kindergarten group at MIT Media lab. It is still a prototype, but we are hoping to identify a core set of frequently used coding blocks, sprites, and features in order to support learners exploring video sensing more independently.

Coding cards

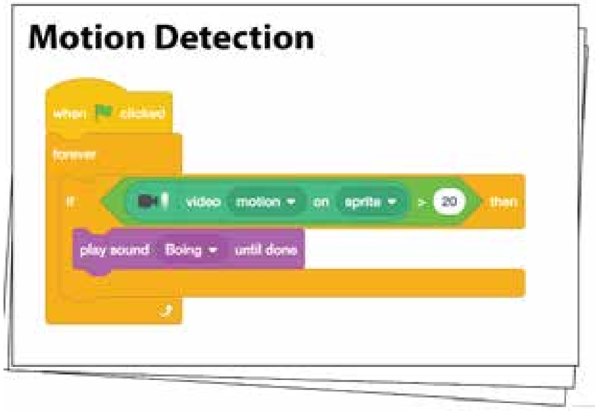
These simple coding cards also helped our facilitation. We made it clear that we were using a video feed—physical motion or specific color— as an "input", and the output was the Scratch animation. Calling out that the input could be either Color or Motion on the coding cards was helpful for learners to understand the cause and effect relationship in the project.



Color Detection

```
when 🏁 clicked
forever
  if touching color ● ? then
    play sound Meow ▾ until done
```

color detection coding card



Motion Detection

```
when 🏁 clicked
forever
  if video motion ▾ on sprite ▾ > 20 then
    play sound Boing ▾ until done
```

motion detection coding card

Color sensing turned out to be better for beginners than motion sensing, since it didn't get triggered accidentally by learner's hands or other objects.

Character creation prompts

After learners picked their own sprite or imported their own handmade sprite, we often asked simple questions such as "What would you like your character to do?" If it sounded too broad for them to answer, we asked more limited questions so that learners could quickly find the next steps.

For example:

"Where would you like your character to be?"
showing that the character can move with mouse

"How big would you like it to be?"
showing that it can change the size by the block "Change size to"

"How would you like it to move?"
showing some blocks in the motion section

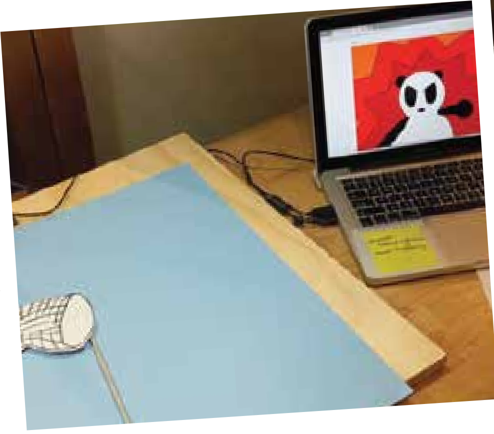
"Would you like to add some sound?'
showing the blocks in the sound section

"What would you like it to do when it touches the ball, the net, etc..?"
showing the character can react to a physical motion or specific color

These questions helped learners navigate through the project. They may not have clear intentions at the beginning, but taking a moment to talk about their immediate next step could eventually guide them to shape the project over time. As a facilitator, it was important to know a few simple Scratch blocks to get started and to make suggestions for blocks based on the learner's ideas. Also, facilitators could be more like interpreters, guiding them by showing how learner's ideas could be expressed in the form of Scratch blocks.

Examples of visitor's projects

A visitor's hand trying to catch a butterfly



Don't punch the panda game

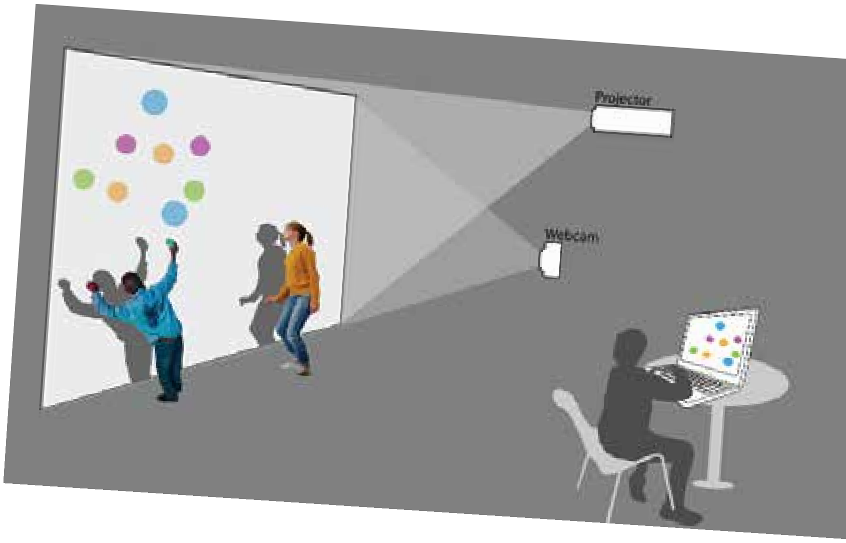
A virtual character climbing down the ladder

Possible Next Steps

Full body, large scale video sensing:
It’s always fun to play with scaling things up! We haven’t tried video sensing in a large scale environment, but we are curious to know how learner’s experience will change when we scale things up from a table top station to an immersive large environment.

Sharing with large community:
Scratch has a large user community but sharing video sensing projects has not been so easy because physical components rely on a live video feed. We would like to explore other ways of sharing video sensing projects that can support both physical and digital worlds, as well as the coding information.

Software development—Enabling “following” the physical motion:
The most common request from learners during the R&D process was how to make a sprite “follow” a physical object. For example, learners often wanted to move their hand in front of the camera and have a sprite follow wherever the hand was moved. Currently, it is possible to write a program to keep a sprite away from the hand, but doing the opposite is not so easy.



Software development—Video sensing Microworld:
We would like to revise the prototype version of the video sensing microworld that the our collaborators at the Media Lab created. The microworld might include a set of prompts, blocks, materials, and environments that work well together to support rich explorations within the area of video sensing. Our goal is to introduce a microworld that learners can start with and can be expanded as needed.

Stand alone video sensing station:
Once we developed a microworld, it might also lead us to the development of a stand- alone video sensing station. We see Scratch video sensing as a good candidate to become an independent exhibit station with a stage, webcam, computer and a good set of props and materials.

Final Thoughts

Throughout the R&D process with visitors in the Tinkering Studio we felt that video sensing definitely provided a rich environment for computational tinkering. When visitors saw a video sensing project for the first time it was met with an excitement and surprise. Visitors were intrigued to program their own animations and test them by interacting with the program via webcam.

Video sensing opened up a new canvas for us, where physical and digital content could reside in the same space. Learners improvised physical components on the canvas while designing digital animation on a virtual layer. Projects that spanned the physical and digital realms really came to life across the layers.

We will continue to explore the many aspects of how video sensing can be tinkered with, and keep reflecting on what seems to be a rich experience for learners when tinkering in the digital and physical worlds.

ACKNOWLEDGEMENTS

SCIENCE SANDBOX Simons Foundation This work was supported by a grant from Science Sandbox, an initiative of the Simons Foundation