Lookit coding manual

Coding to do this summer:

1. Training, using pilot data; review as needed (6+ records, partial videos)
2. Finish coding pilot data (10 records)
3. Coding study data (131 sessions so far) – start with participants who have completed study, to allow report generation (70 sessions, 13-15 each – 5 participants definitely completed). Expect to take ~1 week each, so we can probably get through at least the completed batches and send out feedback!

Other potential things to do:

* Field trip(s): distributing flyers/brochures – pediatrics offices, toy stores, clothing stores, libraries, MIT museum, grocery stores, mommy & me/support groups, ...
* Preparing & sending reports to participants

Table of Contents

Coding overview 2

Where do the files live? 3

Video privacy 4

Coding in VCode 5

Making a template file 5

Opening a video to code 6

Preferential looking coding 6

Behavioral coding 7

Deleting marks 7

Invisible children 7

Saving your work: .cod and .txt files 8

Training: 9

Checking consent and usability 10

# Coding overview

1. Open your coding spreadsheet (codesheet), a CSV file which will live at /Volumes/NovelToy2/CurrentProjects/PrefPhys/. This will show you all usable sessions.
2. Start with the first row where the “allCoders” column is an empty list [] and the usable column says something starting with ‘yes’ (e.g. ‘yes’ or ‘yes.blurry’). Code this session in VCode (see ‘Coding in Vcode’). When you finish coding a session, change the “no” in the “coded” column to a “yes.”
3. Also make a (short) note of anything unusual about the video overall in the ‘coderComments.YOURNAME’ field of the csv. Anything not captured by the VCode file belongs here, including:
   1. Any problems with the video file (very short video clips, missing audio) – missing audio and missing segments happen sometimes and it’s fine to still code, just make a note of it!
   2. Unusual difficulties with coding (dark or blurry, child is far away, webcam is in a weird spot, child seems to only look to one direction)
   3. Major distractions or unusual situations (e.g. two children participating at once, dog trying to eat webcam)
   4. If the privacy level is ‘scientific’ or ‘public’: unusually good-quality or cute video examples (e.g. child is briefly upside-down and we can still easily see her looking both directions, child is laughing throughout)  
        
      You do not need to make a note of things that would be apparent from the coding file, like that the child isn’t looking much after the first few videos or that the parent peeks a lot.
4. Continue coding sessions in order. If you only get partway through a session and have to leave, you can save the VCode file, leave the ‘no’ in the ‘coded’ column but make a note in the ‘coderComments.YOURNAME’ column so you remember next time.
5. At the end of the day, fill in your Google Form to log your time working in the lab.

# Where do the files live?

**Coding spreadsheets**: /Volumes/NovelToy2/CurrentProjects/PrefPhys/

This directory will have the csv files for the pilot and regular physics data, named 57bc591dc0d9d70055f775db\_pilot\_Alice.csv 583c892ec0d9d70082123d94\_physics\_Alice.csv  
 respectively.

**Video**: /Volumes/NovelToy2/VideoClips/PrefPhys/[study name – pilot or physics]/

This directory has a concatenated file studyID\_userID\_sessionID\_privacyLevel.mp4 for each usable session in the study. Coders may view any video in this directory; consent has already been confirmed and video was not withdrawn.

**Note about study IDs**: Each study has a unique ID that identifies it on Lookit. The study ID for the pilot study is 57bc591dc0d9d70055f775db and for the regular physics study is 583c892ec0d9d70082123d94. This is why you will see these strings over and over in filenames!

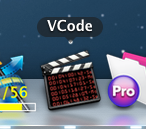
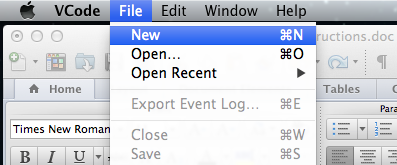
# Video privacy

When we collect video in the lab, we can be very sure the parent is okay with us videotaping: we ask, and we have parents fill out a consent form, and, well, they can see the camera! On Lookit, parents give informed consent via a videotaped statement before the study begins. This means that it’s technically possible for us to end up collecting video without having valid consent—for instance, if a parent doesn’t actually read the statement! *Unless we have a valid consent video, we can’t use the rest of the recordings from that child.* It is as if the parent did not agree to be videotaped in the lab—e.g. we handed them a form, but they didn’t look at it, and they couldn’t see the camera. Regular codesheets will only ever list sessions where consent has already been confirmed.

We also ask parents, at the end of the study, what uses of the video are okay with them. They may withdraw video entirely or select “private,” “scientific,” or “free” use levels. Again, any where video is withdrawn will not be listed on your codesheets. Because you are on the Lookit IRB, you may watch any videos where the parent has consented and not withdrawn video. You may show a video with ‘scientific’ or ‘free’ privacy level within the lab (e.g., to another UROP for advice about how to categorize it, or even just because it’s super-cute), but the ‘private’ ones can’t be shown *even to other lab members* unless they’re also on the Lookit protocol. To be on the safe side, check with Kim before sharing any video even within the lab for now.

For now, all coding must happen in the lab; video files cannot be copied to your personal computer for coding. Do not make copies of or email video or coding files.

# Coding in VCode

You’ll code each session video mp4 using VCode. Open VCode and go to File—> New. (VCode doesn’t open a window when you start.)  
 

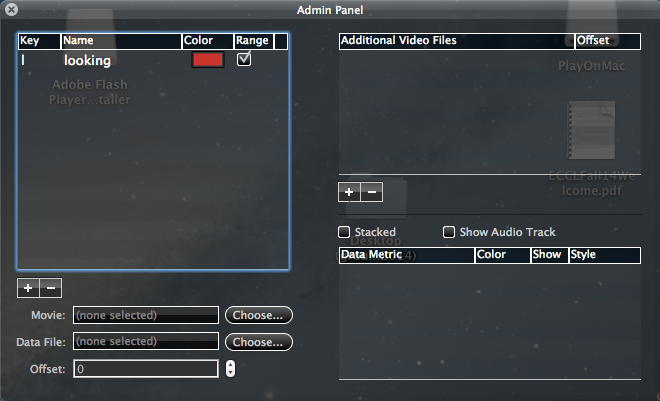
## Making a template file

You’ll have to close VCode and restart it between videos. To avoid entering event types every time, make a ‘template’ .cod file which just has all the types of events defined. Call it something like YOURNAME\_template.cod. For each video, you can then open the template, ‘Save As...’, and then proceed with coding for each video. Here are the event types to add:

|  |  |  |
| --- | --- | --- |
| **Event name** | **Type** | **Precision** |
| right | point | Nearest frame |
| left | point | Nearest frame |
| away | point | Nearest frame |
| end | point | Nearest frame |
| outofframe | range | Nearest frame |
| fuss | range | ~one second |
| peek | range | Nearest frame |
| point | range | Nearest frame |
| talk | range | ~one second |

## Opening a video to code

Open a video to code by clicking ‘Choose…’ next to ‘Movie’ (below, in the Admin panel). If the value in the ‘shortId’ column of your spreadsheet is SESSIONID, and the value in the ‘child.profileId’ column is USERNAME.PROFILEID, the concatenated session video you should code is:

/Volumes/VideoClips/PrefPhys/[study name – pilot or physics]/ studyID\_PROFILEID\_SESSIONID\_privacyLevel.mp4

You can generally find this more quickly by using PROFILEID first. This is a five-character code assigned to the child within a particular user account. Because the same child may do multiple sessions of the study, always double-check that you also have the correct SESSIONID (it’s fine to just look at the first/last three characters to verify).

Don’t check ‘interval playback mode’ (bottom right corner of the movie window), or you won’t be able to proceed frame-by-frame using the arrow keys.

## Preferential looking coding

Use three event types called left, right, and away. Don’t check the “range” box for any of them (this turns the events into “range events” which have a start and stop time, instead of point events which just occur at a particular time).



The names matter (call them ‘left’, ‘right’, and ‘away’), but the keys are up to you. Record the frames at which the child starts looking to YOUR left (by pressing ‘z’ in the example above), the frames at which she starts looking to your right (‘x’), and the frames at which she starts looking away from the screen (‘m’). After initially placing these marks, edit them to the nearest frame by going back and dragging the edges. Of course you’ll never be absolutely certain, but trust yourself here—humans are very good at figuring out direction of gaze!

**End of file**

Add a single non-range event called ‘end’ and mark the very end of the file. This is important to double-check VCode’s idea of exactly how long the file is with ffmpeg’s measurements of the duration of each component clip.

## Behavioral coding

Also add **range** events called ‘peek’, ‘talk’, ‘point,’ and ‘fuss’.

|  |  |
| --- | --- |
| peek | Any periods when the parent’s eyes are open and facing the screen. Do mark even brief peeking. If you can see their eyes but they could only see the screen in their peripheral vision, that doesn’t count as peeking. To nearest frame. |
| talk | Any periods when the parent, or someone else nearby, talks to the child. Add a note to the event with what was said (you can use “\*inaudible\*” for inaudible speech or “\*noaudio\*” for speech where the video doesn’t have audio if you see the parent talking). Don’t worry about exact transcription if you can’t hear well. Don’t mark as talk if parent is speaking to someone else or there’s other background speech. The transcript is used to determine whether the parent could reasonably have influenced the child’s looking direction. To nearest second or so. |
| fuss | Any periods where the child is fussy (making fussy noises, squirming in an attempt to get off the parent’s lap, or crying.) Add a note with a “level” of fussiness: 1 = fussy noises, trying to get off lap but doesn’t seem too upset about it, 2 = unhappily trying to get off lap or actually crying. To nearest second or so. |
| point | Any periods where the parent points to the screen. If the parent is also peeking, add a note that says either ‘left’ (pointing to your left of screen), ‘right’, or ‘general’ (pointing at screen but not to particular event). To nearest frame. |

## Deleting marks

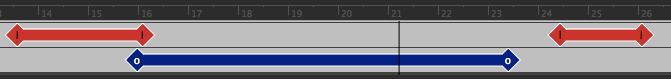
To delete an event, hold ‘Alt’ and click on it.

## Invisible children

If you can’t tell where the child is looking because his eyes are out of frame, use a range event called ‘outofframe’ and add an outofframe event for the duration of any periods where this applies. If you can see the side/back of the child’s head and he’s clearly not looking, or if he wandered into another room, etc. that’s NOT outofframe (it’s ‘away’). If you can see one of the child’s eyes and can pretty much tell where he’s looking, also NOT outofframe.

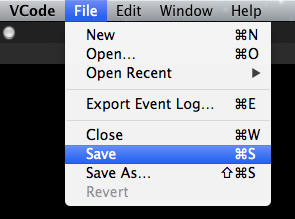
Outofframe will take precedence over any other coding during this period, but we’ll assume that the last mark before the outofframe period describes the time up until outofframe starts, and that the last mark before the outofframe period ends describes what’s going on when the eyes return. (Don’t try to line up e.g. a ‘left’ mark exactly with the end of the outofframe period to show that the child is looking to the left upon return – just put it a bit earlier.) Always code for looking time/preferential looking for the entire period not covered by ‘outofframe’.

Below is an example of outofframe coding (the lower blue bar) along with looking-time coding (the upper red bar). The child was looking at the screen when her eyes went out of view just before 16 seconds. When her eyes became visible again, she was not looking at the screen, but she looked again a bit past 24 seconds.

****

## Saving your work: .cod and .txt files

Once you’re satisfied with the coded video, go to File -> Save:

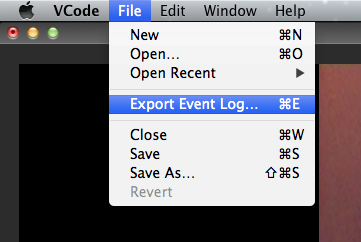


Save the file as /Volumes/NovelToy2/CurrentProjects/PrefPhys/[study name – pilot or physics]/YOURNAME/SESSIONID.cod. (For example,

…/PrefPhys/pilot/Alice/583c8cc0c0d9d70081123d9e.mp4.)

This format makes it easy to go back and check or edit coding later, since we can easily see it overlaid on the video.

ALSO export the events you listed by going to File -> Export Event Log.



Save the text file as /Volumes/NovelToy2/CurrentProjects/PrefPhys/[study name – pilot or physics]/YOURNAME/SESSIONID\_evts.txt, which will be the default name. This is what we actually use for analysis.

## Training:

To get started, we’ll have you code some videos from the pilot study that have already been coded, to make sure we’re on the same page and getting reasonable agreement. For these training videos, you only need to code the first three trials (e.g. 7-pref-phys-videos, 8-pref-phys-videos, 9-pref-phys-videos); however, DO include the ‘end’ mark at the very end of the file. The videos to code will show the coder ‘Training’ as having already coded these videos in the ‘allcoders’ column. Complete these sessions first, and let Kim know when you’re done with them. She’ll check your coding against previous work and go over differences with you, then either approve you to start “real” coding or ask you to do another small set.

# Checking consent and usability

Coding consent is separate from study coding. Currently this is being done by Kim only at ECCL, and separately by other researchers responsible for their own studies, so this is a reference . You will generally want to hide some of the columns you don’t need, to make it easier to see the relevant information.

You’ll need to mark consent for all videos that currently say “orig” in the “consent” column and where nVideosExpected >= 1.

The value in the ‘shortId’ column is the SESSIONID. To find the video for a given SESSIONID, look in sessions/[STUDYID]/consents/. The video filename will be in the form:

videoStream\_STUDYID\_1-video-consent\_SESSIONID\_TIMESTAMP\_RANDOM.flv

These are currently a bit of pain to find; we’ll figure something out soon.

1. Once you’ve found the video, make sure that an adult makes the following statement: “I have read and understand the consent document. I am this child’s parent or legal guardian, and we both agree to participate in this study.”

Things that are NOT OKAY:

* Adult does not say anything (e.g., parent appears to be reading silently) or speaks but does not read the consent statement (e.g., parent and child play with the webcam view of themselves they see during this segment)
* No audio (UNLESS parent is signing OR you confirm consent by email; run any of the latter by Kim)
* Video cuts off well before end of consent statement (“I have read and understand the information above…”) even though this is very sad.

Things that ARE okay:

* Choppy audio and/or video, or poor video quality (e.g. you can only see a blurry face)
* Parent’s face is not in the frame but an adult is clearly the one to make the verbal statement
* Child is not present during consent statement
* Minor variations on wording as long as meaning is the same (e.g. “I read and understood,” “I am this child’s father,” “and we agree to participate”)
* Audio only (but make a note of this under the ‘consentnotes’ column)

If you encounter any video you are not sure about, contact Kim or Laura. Just leave it (don’t mark it either consent or not) in the meantime. Never delete any data without checking with a supervisor.

1. Filling in the ‘consent’ column in the spreadsheet:

* If you determine that the parent consented, write ‘yes’ in the ‘consent’ column. Do not write anything but the exact string ‘yes’. (E.g., do not write ‘yes, but audio only.’)
* If you determine that the parent did not consent, record one of the following in the consent column:
  + ‘noreading’ (adult and child are there, but adult doesn’t read the statement)
  + ‘noaudio’
  + ‘nochild’ (adult in the frame who doesn’t read the statement plus no suggestion a child is just out of frame)
  + ‘testaccount’ (if it’s Kim or you or someone we know is working on Lookit)
  + ‘cutoff’ (looks/sounds good but stops before the end of the statement)

1. Fill in the “consentnotes” column with any other notes separated by commas.
2. **If consent is “yes”, also record whether the video is usable in the ‘usable’ column**. This will be used to filter sessions so coders only see ones they need to code! If it is usable, write “yes”. (For piloting, may also write ‘yes.[reasonitmightnotbe]’, e.g. ‘yes.blurry’ or ‘yes.attention’. If it is not usable, write the reason why not. Valid reasons are shown in the table on the next page. Use the highest-up one possible (e.g. if there is only a consent video and also there isn’t audio, use incomplete.)

|  |  |
| --- | --- |
| withdrawn | Video is present but the withdrawn column is TRUE. |
| incomplete | There are fewer than six trial videos. |
| nochild | There isn’t a child present, even in the later study videos—just an adult, or no one. |
| agerange | The child in the videos is clearly out of the age range for the study, regardless of what the registered birthdate says. (E.g., for a 4-12 month study a child is talking fluently and looks about 5.) |
| onlyaudio | There is no video, but we do have audio. |
| childoutofframe | The child is out of the frame for most/all of the study (but there is a child). |
| videoquality\_REASON | The video quality is so poor that you can’t code it. REASON values are framerate, cameraangle, screenmovement, blurry, dark, shortvideos, etc. – we’ll sort these out as we continue coding. |
| attention | <6 trials with child looking for at least a few seconds because child is fussy and/or distracted |
| postpone | There’s something weird about this video and Kim needs to check it or fix it before it’s coded. (E.g. it’s upside-down! You think that kid looks way too young! The individual clips are in the wrong order!) |

# Updating data

(Also only for reference – handled by Kim.) Videos and data on NovelToy2 are currently not a primary copy of the data. Updates will happen manually as needed: copy new concatenated videos to NovelToy2; fetch the codesheet and commit locally; get an updated codesheet and send it back to NovelToy2.