Lookit coding manual

Table of Contents

Overview of a coding session: 1

Checking consent videos 2

Study-specific coding 5

- **Fussiness and distraction:** 7
- **TESTIMONY ("Learning from others"):** 8
- **NOVELVERBS ("Learning new verbs"):** 12
- **ONESHOT ("Intuitive probability"):** 13

Looking measures coding in VCode 14

- **Looking time coding:** 14
- **Preferential looking coding:** 15
- **Deleting marks and invisible children** 15
- **Saving your work** 16

Resolving coding conflicts: recoding 17

Exercise 19

Overview of a coding session:

A. Important: these steps must be done from the computer TMI (closest to the door).

At the start of the coding session, update video and spreadsheet data by double-clicking these two shell scripts under /Volumes/NovelToy2/CurrentProjects/Lookit/:

1. SyncLookit.sh (Uploads inclusion information from spreadsheets to Lookit server; adds new records from server to consent spreadsheets at the lab; downloads new videos and organizes them)
2. UpdateSpreadsheets.sh (updates coder spreadsheets based on consent spreadsheets, updates master spreadsheets from coders, and lets other coders know what you've finished)

B. Consent or study-specific coding

C. Run UpdateSpreadsheets again from TMI and fill in your Google Form to log the time.

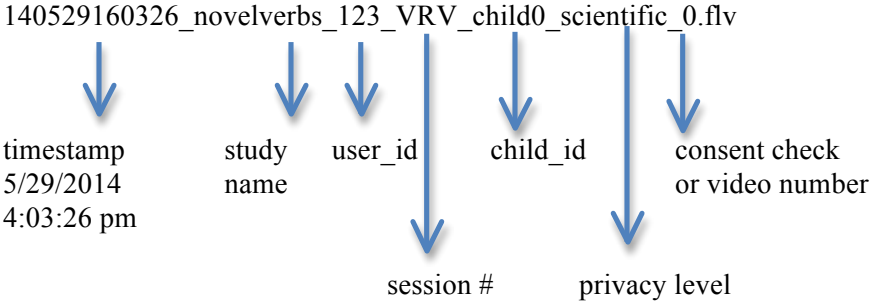
Checking consent videos

Coding consent is SEPARATE from study coding. If you're coding consent, you'll first determine whether you have valid consent videos and record that in the spreadsheets under Lookit/Consent/. Then you'll run UpdateSpreadsheets.sh again, and the videos with valid consent will be available on everyone's individual coding spreadsheets.

Open the consent spreadsheet for the study:
/Volumes/NovelToy2/CurrentProjects/Lookit/Consent/[studyname].xls

You will generally want to hide (never delete/rearrange!) some of the columns you don't need, to make it easier to see the relevant information.

Go to the first row where there's no decision in the 'consent' column and start there. For each row, find the associated consent video under /Volumes/NovelToy2/VideoClips/Lookit/[studyname]/consent/. You can find it using the account number (in the 'user_id' column) and the timestamp, which are both in the consent filename:



Our oldest video files may not include a session #.

If you can't find the consent recording here, check under
/Volumes/NovelToy2/CurrentProjects/Lookit/consent_backup/
If you find it there but not in VideoClips, make a note of that under 'consent notes' and move the file to the correct VideoClips consent directory.

Starting 4/15/15, we're getting all of the consent videos from the Wowza server, so we may have multiple consent videos from a participant (just means that they tried the recording multiple times). A record is created on the spreadsheet once the parent moves BEYOND consenting to the actual study.

You should always look for the video that actually matches the recordingSet on the spreadsheet first. (Sometimes there are multiple records on the spreadsheet for the same userid & childid; in this case you should look for and code both consent videos.)

- If that video is valid, you can move the "extra" consent videos (generally recorded right before that one) to the "consent_extra" directory. Parents may have encountered problems with recording (e.g., you might have a no-audio consent video, then one with audio, because they saw an error message—yay!), or they may have gotten confused, or they may have just wanted to make their video perfect. You don't need to watch and code all of these unless they're on the spreadsheet.

- If that video isn't valid, and you have other consent videos recorded just before it, watch those and see if any are valid. (This obviously only holds for technical or no-reading issues—maybe the parent tries "redoing" the video and then realizes she didn't need to, so she doesn't read the statement again. Or maybe you had audio on an earlier video but not this one). Mark each video that you watch. If one of them is a valid consent video, mark the record in the spreadsheet as 'yes' (consent) and make a note of this under 'consent notes.'

- If you are left with consent videos that don't correspond to a user & child who are in the spreadsheet as participating on that date, move them to consent_noparticipation after double-checking they're really not on the spreadsheet. These are generally people who tried making a consent video but then decided not to participate after all, OR who had technical problems and couldn't get past the consent screen.

1. Once you've found the video, make sure that an adult makes the following statement: "I have read and understand the information above. I am this child's 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 the video quality is so good you're confident you can lipread)
- Video cuts off well before end of consent statement ("I have read and understand the information above. I am this child's legal guardian…") 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 'consent notes' 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.

2. 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) – Kim is user 342, Katy is 451
- ‘cutoff’ (looks/sounds good but stops before the end of the statement)

3. Marking the consent video itself:
   - If the video shows valid consent, change the ‘0’ at the end of the filename to ‘CONSENT’
   - If not, change the 0 to ‘NOCONSENT’ and remove the .mp4 or .flv extension.

   Examples:
   140529160326_novelverbs_123_VRV_child0_scientific_CONSENT.mp4
   140529160326_novelverbs_123_VRV_child0_scientific_NOCONSENT

   These will be automatically moved to the ‘consent_yes’ and ‘consent_no’ directories when updateSpreadsheets.sh runs.

4. Fill in the ‘coder name’ column with your name! Fill in ‘consent notes’ with either ‘child present’ or ‘no child present’ for every video, plus any other notes separated by commas (e.g. ‘child present, super cute—dog barks right at “we agree”). Don’t worry about the first column of the spreadsheet, ‘INCLUDE_IN_ANALYSIS’.

Special situations:
If you come across a consent video tagged as –WITHDRAWN or whose privacy level is INCOMPLETE, still watch it (by temporarily adding an .mp4/.flv extension) and add a consent code before the –WITHDRAWN tag. However, even if you confirm consent you may NOT open any of the videos if they also have this tag (either WITHDRAWN or INCOMPLETE). (An INCOMPLETE consent video with valid consent plus private/free/scientific test videos are completely okay, though.)

A note about privacy levels: By being on the Lookit IRB, you may view any videos submitted as long as the parent has consented and not withdrawn (no “—WITHDRAWN” tag in the filename). 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 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.

Study-specific coding

Each coder has a set of study spreadsheets under CurrentProjects/Lookit/Coding/YourName/ Only use these spreadsheets for coding.

The spreadsheets have a lot of information, most of which you'll rarely need. To make them easier to read, YOU CAN:

- Freeze the top row if it isn't already
- Hide columns you don't need (right click, hide)
- Change the font size
- Change any other formatting (shade in rows you need to code or that you don't need to code; you can even use conditional formatting to do this automatically, but for now it'll be erased each time the spreadsheets are updated)

BUT DON'T:

- Copy data manually
- Delete any rows or columns
- Rearrange columns, *other than the ones you enter data into*
- Rename any columns

If you come across issues that are not covered in these notes, put them in a separate column 'important' and discuss them at the next meeting with Kim. (Any time I say something like 'but make a note of it if that happens!' put it in the 'important' column.)

You can find the videos under VideoClips/Lookit; in the example above they'd be under VideoClips/Lookit/novelverbs/123child0/VRV/

For all studies:

- **Once you've coded a study, put an 'x' in the 'coded-YourName' column of your spreadsheet**. This indicates that we should expect to find a VCode file (if applicable) and the rest of the spreadsheet row filled out. Don't write anything but 'x' in this column—any other notes should go elsewhere. This column is used to inform other coders what's left to code, on a master spreadsheet that lets us know how we're doing overall, and in analysis code.

- **You can see who else has coded a file in the 'all-coders' column**. Prioritize records with fewer other coders, and skip any videos where 'usability_judgment' has a reason not to code it (e.g. 'Katy-novideo' means Katy has already checked this one and there's no video.)

- **If the privacy level is 'free', also note whether it's a useful example** under a column called 'example_notes'— e.g., "really focused kid, clear answers without prompting" or "great example of parent trying really hard to get the kid to say one thing and it not working" This is really helpful in preparing talks and training new coders.

- **If the video isn't usable, record why not in the 'video_usable' column**. This will be shared with other coders so other people don't have to figure this out themselves! In the video_usable column, write one or more of the following reasons, separated by commas. Valid reasons are shown in the table on the next page. Use the highest-up one possible (e.g. if there are only 3 videos and also there isn't audio, use incomplete_3.

video_usable column values:

novideo	There are actually no audio/video files associated with this session; either the parent consented and then left or there's a technical problem.
withdrawn	Video is present but has a WITHDRAWN tag at the end of the filenames
noprivacylevel_N	Same as incomplete_N, but if none of the video files have a privacy level in the filename. E.g. we have clips ending in _INCOMPLETE_1, _INCOMPLETE_2, _INCOMPLETE_3: noprivacylevel_3. We may be able to make an exception if the parent contacts us.
incomplete_N	We have all the videos through clip N, which is not the end of the study. E.g. we have clips 1, 2, and 3 out of 8: incomplete_3
missingvideo	We have some weird combination of videos, neither 1 thru N or all, due to a technical glitch. (E.g., we only have videos 11, 12, and 13.)
nochild	There isn't a child present—just an adult, or no one.
noaudio	There is no audio in the video files. This is only used in verbal-response studies (testimony, causaldomains, …).
onlyaudio	There is no video, but we do have audio. This is only used in looking studies (oneshot, novelverbs, …)
childoutofframe	The child is out of the frame for most/all of the study (but there is a child) – relevant for looking time and preferential looking studies only.
videoquality_REASON	The video quality is so poor that you can't code it. Use videoquality_framerate if the changerate column is less than 2. Other REASON values are screenmovement, blurry, dark, etc. – we'll sort these out as we continue coding.
fussy distracted	If you come across a kid you're pretty sure we can't include for one of these reasons, do just the F/D/P coding first, then check with Kim before using one of these labels.
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!)

Fussiness and distraction:

Fussy: The child cries at any point in the video, or squirms in an attempt to physically face away from the screen and/or get off the parent's lap (not just look away).

Distracted: (In the sense of 'Bob distracted Joe,' not 'Joe is a distracted guy'.) Any lookaway is clearly caused/motivated by an external event (e.g., the TV, a sibling walking by): i.e. if the event hadn't happened, the child probably wouldn't have looked away. Looking away to perform an action is a tricky edge case and should follow the principle that simple competition in the visual environment is not a distraction.

Distracted:

- child is fiddling with toy, but then drops it, now reaches for it
- child is snacking on chips from a bag and looks inside for one

Not distracted:

- Child reaches for and looks at keyboard
- Child looks at parent (unless parent just laughed/talked)
- TV in the background (unless just turned on/obviously changed)
- Parent talks but child doesn't look away

Err on the side of 'not distracted' – if it's unclear whether the lookaway was due to the external event, call it NOT distracted.

Additional edge cases can be added to the 'Coding notes' tab of the timesheet Google doc.

TESTIMONY ("Learning from others"):

1. Transcript

Transcribe the clip, writing everything the parent says or does within square brackets. Transcribe inaudible whispers, mumbling etc as *inaudible*. Transcribe replays and other pertinent actions where they occur as *replay*, *nod*, *point left*, etc
For instance (don't include the quotes):

"I don't know, I guess it's a modi [Are you sure?] *inaudible* yeah".

2. Interaction Coding

Enter a comma-separated list of any of the following that occur, in the order that they occurred:

replay = (parent plays the audio question again on the computer)

prompt = Any parent talk that isn't repeat_ or feedback_
e.g. "*go ahead*", "*what do you think*", "*who got it right*?", "*talk louder*"

repeat_question = Parent gives any acceptable repetition of the question
"*Who do you think was better?*" "*What do you think it's called?*"
repeat_options = Parent repeats the two options, either verbally or by pointing
"*The girl in the yellow shirt or the girl in the red shirt?*", "*Modi or danu?*"
* Code "*Who was better? Red or yellow?*" as repeat_question,repeat_options
* It is acceptable for parent to repeat the wrong options

feedback_pos = Parent gives positive feedback after child gives a response ("Good job.")
feedback_neg = Parent gives negative feedback or corrects an answer that the child gives
A duck. [Nuh uh…]", "Purple. [There was no purple.]", "You're silly. Are you serious??"
feedback_repeat = Parent repeats an answer that the child gives, to ask them for confirmation. Only code if child responds.
e.g. "*A spoon. [A spoon?]. Yeah*", "*Yellow*. [*You think the yellow shirt was better?*]"
* Do not code if parent is merely repeating to confirm the child's response for clarity,
e.g. "*Duck. [Okay a duck]*", "*Danu. [He said danu]*"

childpoint, parentpoint = Either the child or parent points at an informant
interpretpoint = After child points, parent confirms what he/she pointed at

finalanswer_[cleanedanswer] = Final response you used for the 'cleaned' column. Check next section "Cleaned answers" to figure out which child response to choose as final answer
childanswer_[cleanedanswer] = Any other response (including invalid ones)the child gives as answers to the question. Ignore responses the child gives to their parent.
* Make sure that the final answer is in the same sequential position as the response you chose.
e.g. "*Red.* ***Yellow***. *You choose! [Which one?] Red.*"
→ childanswer_red, **finalanswer_yellow**, prompt, childanswer_red

inaudible_parent / inaudible_child / inaudible_other = Any inaudible, fuzzy segments that you can't decipher.

dontknow = Child says he/she doesn't know

answer = Parent says the answer, e.g. "*uh uh… [danu]*"

3. Cleaned Answer

This is, roughly, "the answer" the child gave. It should correspond to what you coded as "finalanswer_[the answer]". Possible "cleaned answers" are in the table on the next page.
Definitions:

- An "valid" answer is a word that appeared in the test question. (red/yellow, spoon/duck/hat, danu/danu_var, etc).
- Invalid answer: responses coded as "other", "na", or "combination".
- Parent "interference" is answer, feedback_neg

Essentially, we divide the child's responses into chunks between parent speech/action. We choose the earliest block that we find a usable child response. Within that block, we use the child's last answer.

Simple scenarios:

1. If there is no parent speech at all, take the child's **last valid response** (include invalid answers)
2. If there are no valid answers at all, take the child's **last response** (other or na) before any parent interference (feedback_neg or parent answer)

If there are valid answers mixed with parent speech,

3. Take the **last valid response** before the first parent speech.
4. Otherwise, if there are no valid answers in the current chunk of child speech, take the **last valid response** in the next chunk of child speech.
    a. If there was any parent interference before the chosen response, prefix it with "parent_" (e.g. "parent_correct", "parent_modi").
5. Repeat Step 4 until you find a valid answer. If there are no valid answers, you should be in scenario 2 (take last response before any parent speech).

Examples for Clip 2: "Transcript" → Interaction codes → Cleaned answer

"*A **wigabaloo***" → **finalanswer_other** → other

"*A wigabaloo… a duck, **spoon**!*" → childanswer_other,childanswer_incorrect,**finalanswer_correct** → correct

"*A wigabaloo…a spoon, a **duck**! [What do you think?] A spoon!*"
→ other,childanswer_correct,**finalanswer_incorrect**,prompt,childanswer_correct → incorrect

"*I don't know…[What do you think?] I don't know… a **wigabaloo***"
→ dontknow,prompt,dontknow,**finalanswer_other** → other

"*[What do you think?] I don't know…a silly, [Is it a spoon or a duck?] a **spoon**!*"
→ prompt,dontknow,childanswer_other,repeat_options,**finalanswer_correct** → correct

"A *horn! [No, actually it's a spoon or a duck] A duck? A **spoon**!*"
→childanswer_other,feedback_neg,repeat_options, childanswer_incorrect,**finalanswer_correct** → parent_correct

Cleaned answers are other, na, tech, or clip-specific options below.

If the child indicates an informant ("what the red one said" or pointing to a specific side), add on "_informant" e.g. gobi_informant, red_informant. The yellow shirt girl is always on the CHILD'S left.

Double-check with Kim if you are not sure about any response not included below, especially additional "combination" or "_var" responses.

For ALL clips

na	Child doesn't answer. Includes "don't know", silence, and inaudible
tech	No audio or cutoff.
other	Silly noises or words not from experiment (e.g. "horn")
both	Child clearly endorses both valid options, e.g. "red AND yellow!" or "it's a modi and a danu"

Clips 2/4/6/8

correct	Clip 2 – spoon; Clip 4 – bottle; Clip 6 – brush; Clip 8 – doll. Accept equivalent terms e.g. brush=comb, doll=dolly or baby
incorrect	Clip 2 – duck/hat; Clip 4 – fork/apple; Clip 6 – plate/key; Clip 8 – tree/cup. Accept equivalent terms e.g. ducky

Clip 11:

toma	Toma or 1 sound off (cannot overlap with 'gobi'): tuma, teema, doma, tona, dona, etc.
toma_var	Toba, Tomi, Tomia
gobi	Gobi or 1 sound off (that doesn't overlab with 'toma'): gubi, mobi, bobi, gopi, etc.
gobi_var	Gomi, Goba, Dobi, Gabi
combination	Tomagobi, Gobitoma, Goma, Tobi, etc
other	

Clip 13

modi	Modi or 1 sound off (cannot overlap with 'danu'): moda, mudi, dodi, etc
modi_var	Moni, Modu
danu	Danu or 1 sound off (cannot overlap with 'modi'): danuh, dinu, danya, danon, dayu, etc
danu_var	Dadu, Dani
combination	Modinu, Monu, Danumo, Dadi, etc.
other	Domi, Doni, Dobi, Domina, etc

Clip 15

dax dax_var	Dax or 1 sound off (that doesn't overlap with 'wug'): dox, bax, dast, etc. Dux, Dag
wug wug_var	Wug or 1 sound off (that doesn't overlab with 'dax'): wog, mug, one, etc. Wag, Wux
combination	Daxug, Dug, Wugax, Wax, Wex, etc.
other	

Clip 17

riff riff_var	Riff or 1 sound off (that doesn't overlap with 'fep'): raff, rick, Riss, Vrif, Biff, etc. Rip, Ref
fep fep_var	Fep or 1 sound off (that doesn't overlap with 'riff'): fop, fet, etc. Feff, Fip
combination	Riffep, Rep, Fepiff, Fiff, etc.
other	Vip, Bib, Wip, Sip, etc.

NOVELVERBS ("Learning new verbs"):

Clip 1: attentiongrabber
Clips 2-4: find clapping
Clip 5: attentiongrabber
Clips 6-8: find tickling
Clip 9: dialogues
Clip 10: attentiongrabber
Clips 11-13: find [verb]/what's happening

VCODE:

- Code the combined clip for preferential looking in VCode using three point events (not "range"): 'left', 'right', and 'away'. This can be done close to real-time (rather than adjusting frame-by-frame).
- Exception: Just code the dialogue clip (#9) for looking time (one range event – 'looking').
- Add a range event 'outofframe' as needed to indicate when it's not possible to tell where the child is looking because his/her eyes are not visible.
- Also make a point event for 'sound', which should correspond to the start of the first word spoken in our audio for clips 2-4, 6-8, and 11-13. Add a 'nosound' event at any point in a trial that does not have audible sound.
- Add a point event for 'trial', which should be at the first frame of each new clip.

SPREADSHEET:

- For each clip (1-13) mark whether the child was **f**ussing using the columns F1, F2, …, F8.

- For each clip (1-13) mark whether the child was **d**istracted using the columns D1, D2, …, D8. Enter either 0 (not distracted) or 1 (distracted).

- For only the "test" clips (6-8 and 11-13) mark whether the **p**arent interfered using the columns P6, P7, P8, P11, P12, P13 columns. Answer can be either 'no' or a comma-separated list of whichever of the following apply:
    - **talk** [parent talks *to child* in a way that potentially affects where the child is looking. Ex: 'Your bottle fell?' when the parent realizes the child's bottle has fallen and the child is already trying to find it is NOT 'talk.' But 'What do you see, what are they doing?' or 'WOW look at THAT!' or 'Keep watching, sweetie' is.]
    - **eyes** [parent's eyes are open for most of the trial]
    - **peek** [parent's eyes are briefly open]
    - **start** [parent's eyes are open for the first 1-2 seconds, to see that something actually happens on screen]
    - **noeyes** [parent's eyes are usually out of frame—can't tell if he/she is looking]
    - **point** [parent points at the screen]

  Examples of valid parent interference notes:
    - no
    - talk, peek

  Examples of **invalid** parent interference notes:
    - yes
- talk and peek

ONESHOT ("Intuitive probability"):

Clips 1-4: warmup trials, looking time to 50% probability lottery events
Clips 5-8: test trials, looking time to 75% or 25% probability lottery events

VCODE:

- Code the combined clip for looking time in VCode using one range event 'looking'. Adjust the starts/stops of looks frame-by-frame.
- Add a range event 'outofframe' as needed to indicate when it's not possible to tell where the child is looking because his/her eyes are not visible.
- Also make a point event for 'trial', which should be at the first frame of each new clip (when the red number in the corner changes).

SPREADSHEET:

- For each clip (1-8) mark whether the child was **f**ussing using the columns F1, …, F8. 0 = not fussy, 1 = fussy.

- For each clip (1-8) mark whether the child was **d**istracted using the columns D1, …, D8. Enter either 0 (not distracted) or 1 (distracted)

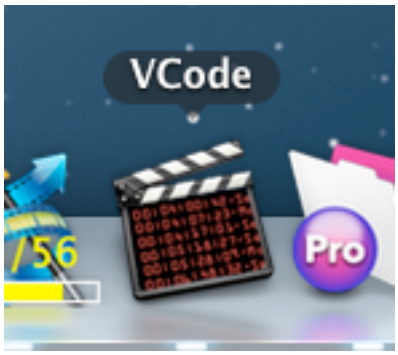
- For only the last four clips (5-8) mark whether the **p**arent interfered using the columns P5, …, P8. Answer can be either 'no' or a comma-separated list of whichever of the following apply:
    - **talk** [parent talks *to child* in a way that potentially affects where the child is looking. Ex: 'Your bottle fell?' when the parent realizes the child's bottle has fallen and the child is already trying to find it is NOT 'talk.' But 'What do you see, what are they doing?' or 'WOW look at THAT!' or 'Keep watching, sweetie' is.]
    - **eyes** [parent's eyes are open for most of the trial]
    - **peek** [parent's eyes are briefly open]
    - **start** [parent's eyes are open for the first 1-2 seconds, to see that something actually happens on screen]
    - **noeyes** [parent's eyes are usually out of frame—can't tell if he/she is looking]
    - **point** [parent points at the screen]

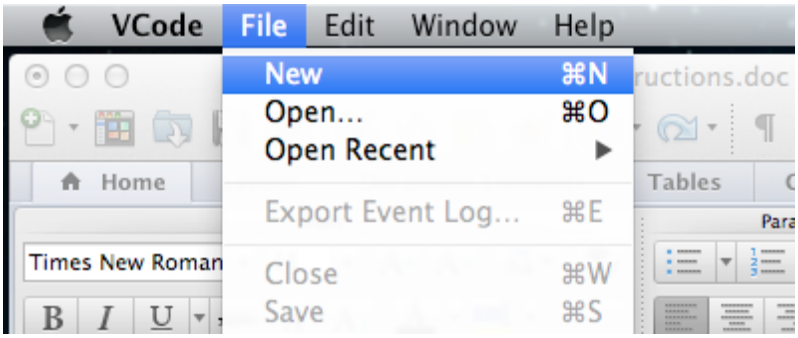
  Examples of valid parent interference notes:
    - no
    - talk, peek

  Examples of invalid parent interference notes:
    - yes
    - N
    - Talk and peek

Looking measures coding in VCode

Open VCode and go to File—> New. (VCode doesn't open a window when you start.)

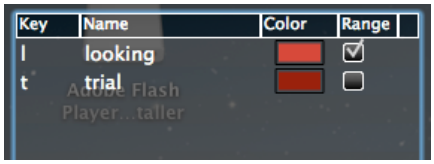




Open a video to code by clicking 'Choose…' next to 'Movie' (below, in the Admin panel):

Looking time coding:

Create two types of events, 'trial' and 'looking.' Make 'looking' but not 'trial' a 'range' event, meaning it will be defined as from time A to time B rather than just occurring at a single timepoint. Their names are important (should always be named 'trial' and 'looking'). The keys you use as shortcuts are up to you—I've set trial to 't' and looking to 'l' here. (Don't use uppercase letters though! Then you'll actually have to use shift or capslock to enter events.)



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.

Press 't' (or whatever key you've chosen) to record a 'trial' event at the first frame of each new clip (when the red number in the corner changes). Press 'l' to start and stop 'looking' events, which denote the periods when the child is looking at the screen. **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 a bit here—humans are very good at figuring out direction of gaze.

Preferential looking coding:

As with looking time coding, you'll need a trial event, but instead of a range event for looking you'll use three point events: left, right, and away:

The names matter (call them 'left', 'right', and 'away'), but the keys are up to you. In addition to coding the start of each trial as for looking time, 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').

Don't check 'interval playback mode' (bottom right corner of the movie window).

You may be asked to do this coding frame-by-frame (using interval playback) or real-time (to get through the data faster but less precisely).

Deleting marks and invisible children

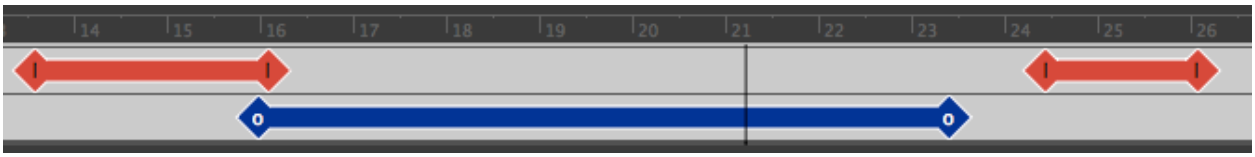
To delete an event, hold 'Alt' and click on it.

If you're re-coding, you may come across events 'deleted' before we figure out how to delete them properly. Range events with the end before the start are treated as deleted, as are point events with the note 'x' (double-clicking on an event with a black border lets you see and edit the note). You can leave these alone or delete them with alt-click.

If you can't tell where the child is looking because his eyes are out of frame, use the range event called 'outofframe' and code for this wherever you're coding for preferential looking or looking time. 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. If you can see one of the child's eyes and can pretty much tell where/whether he's looking, also NOT outofframe.

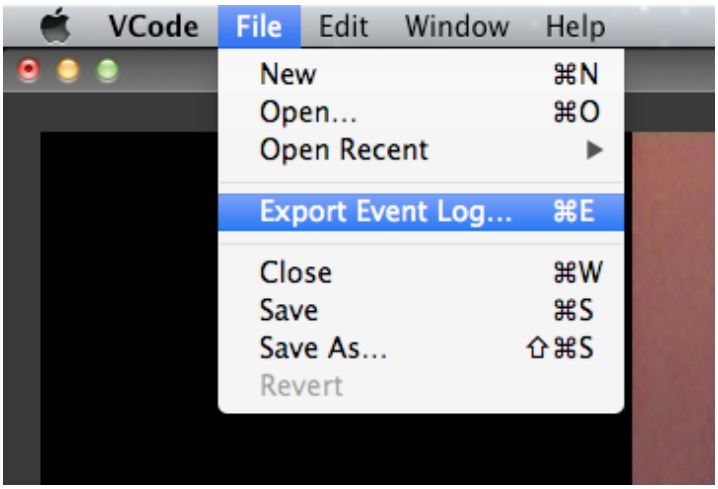
Outofframe takes precedence over any other coding during this period. Always code for looking time/preferential looking for the entire period not covered by 'outofframe' – otherwise, we may end up thinking (for instance) that the child looked away just before going out of frame. It's fine for looking and outofframe coding to overlap (if a frame if contained both in outofframe and in looking, we'll know the child is out of frame and give that priority).

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

Once you're satisfied with the coded video, go to File ->Export event log:

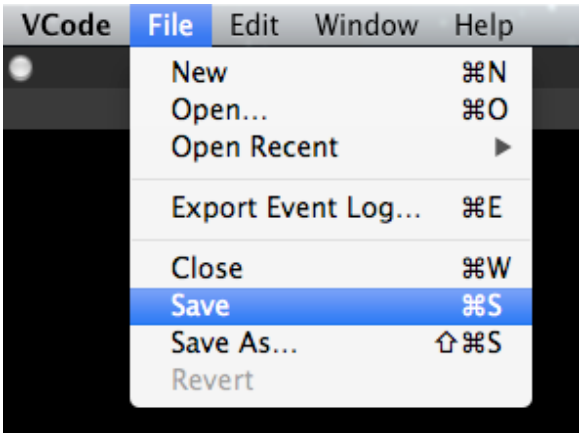


Save the text file under
CurrentProjects/Lookit/VCode/[study]/[yourname]/[studyname]_[subjectnumber][childID].txt

For example,
CurrentProjects/Lookit/VCode/novelverbs/Kim/novelverbs_763child0.txt

If there is more than one usable video for a particular child, add the session code to the end of BOTH fields, like this:
CurrentProjects/Lookit/VCode/novelverbs/Kim/novelverbs_763child0_Y7bh3z.txt
CurrentProjects/Lookit/VCode/novelverbs/Kim/novelverbs_763child0_Mn5tho.txt

ALSO save the file in .cod format by using File -> Save. Use the same filename with a .cod extension. This allows us to easily go back and see your coding overlaid on the video, compare it to someone else's coding, and make edits if needed without redoing the whole thing.



Now comes the frustrating thing about VCode: you'll have to close the program and restart it between videos. To deal with this problem, make a 'template' .cod file which just has all the types of events defined. Call it something like Myname_template.cod. For each video, open the template, 'Save As...', and then proceed with coding for each video.

Resolving coding conflicts: recoding

When two blind coders disagree about looking time or preferential looking results, you may be asked to recode certain trials. Notes will appear in your 'coded-Name' column when recoding is required. To resolve these conflicts, look through this column for anything other than an 'x' or empty cell:

There are two types of notes you may see:

- If you previously coded a file and there is a disagreement with another coder, your 'x' will just turn into a comment like "check trials 3 and 7."

- If you did not previously code a file, you may be asked to "adopt" another coder's work, then check certain trials. (The other code no longer works in the lab and/or no longer works on this study.)

  To do this, find the associated .cod and .txt files from the other coder's VCode directory (CurrentProjects/Lookit/VCode/studyName/otherCoder/) and move them to your own directory. (Do not leave a copy in the original coder's directory!)

To resolve the conflict, open the corresponding .cod file (now in your directory) in VCode.

> Note: VCode may say it can't find the movie; this is just because coding was done on a different computer and nothing is actually lost. Go to Window -> Admin window, and click 'Choose…' next to Movie to re-select the combined clip.

In VCode, double-check your work for the trials you were asked to check. This includes:

- Check that the trial markers before/after these trials are correctly placed and 'sound' is correctly placed for Novelverbs

- Check that outofframe coding is used if appropriate and that the starts and ends of these periods are correct, and that there is accurate looking-time or preferential-looking coding for all regions outside the outofframe periods

- Check ‘looking’ and ‘left’/’right’/’away’ timing and revise to the nearest frame (even if coding was not originally done frame-by-frame); make sure any brief looks/lookaways are marked

You can move existing marks around, add new ones, and delete existing marks, however works best for you to revise the coding.

You may find you agree with yourself almost entirely, and that’s okay. This may be a case where you were right and the other coder missed something, or where there’s a genuine disagreement about whether a child is looking, or where you really only disagree by a few frames but that makes a large difference in looking time because of a threshold.

When you’re satisfied with the trials that needed recoding, re-save the .cod file and re-export the .txt file. You do not need to save your original files. Change the mark in your ‘coded-Name’ column to an ‘x’.

Exercise

1. What should you do after you finish coding for the day?

2. What account number is associated with this video?
   150326120214_testimony_452_YhgUob_child0_scientific_12.mp4

3. [Consent coding] You watch a consent video where an adult reads "I have read and understand the information above. I am this child's legal guardian" and then the video cuts out, but audio continues and he finishes the statement. There is no child present. Is this a valid consent video?

4. [Consent coding] You watch a consent video where a mother reads the statement, but in the middle the baby on her lap starts screaming. The parent yells "Shut up, you stupid thing, we need to finish this goddamn task and get paid, are you going to ruin everything?" She finishes the consent statement. Is this a valid consent video? What should you do?

5. [Consent coding] Which of these can go in the 'consent' column?
   'yes' 'yes, noaudio' 'noreading' 'novideo' 'testingaccount'

6. You are giggling at an adorable video of a baby who just can't stop smiling at the camera. Your friend, who is also UROPing in ECCL, walks by. What determines whether you can show her the video?

7. You start coding a video but quickly realize it is a video of an adult holding up a picture of her child on a phone. It is a static image (the child is not on Skype or similar). What should you do?

8. Here is part of a coding spreadsheet. Cross out the rows you shouldn't code at all. Put a dot next to ones that get higher priority.

user_id	endedEarly	child_id	recordingSet	coded-Rianna	all-coders	notes	video_usable	usability_judgments
592	"false"	"child0"	"x1KQN3"		['Harrison']			['Harrison-missingvideo']
1559	"false"	"child0"	"kNLylh"		['Scout']			['Scout-childoutofframe']
1561	"false"	"child0"	"P6hVYP"		['Scout']			[]
1310		"child1"	"KNp76d"		['Harrison']			['Harrison-novideo']
1572	"false"	"child0"	"NMXT8V"		[]			[]
1576	"false"	"child0"	"nWw4qg"		['Scout']			[]
539	"false"	"child2"	"LLLzBq"		['Harrison']			[]
1598	"false"	"child0"	"DqXyHb"		['Scout']			[]
1606		"child0"	"brh82M"		[]			[]
1770	"false"	"child0"	"WrxvR4"	x	['Rianna']			[]
1771	"false"	"child0"	"7GHLR2"		[]			[]
1774	"false"	"child0"	"VYZHWH"	x	['Rianna']			[]
1779	"false"	"child0"	"2k8Mhw"	x	['Rianna']			[]
1789		"child0"	"nZKNBN"		['Rianna']		novideo	['Rianna-novideo']
1789		"child0"	"1N6bPl"		['Rianna']		novideo	['Rianna-novideo']
1789	"false"	"child0"	"mP0g85"	x	['Rianna']			[]
1790	"false"	"child0"	"z1YwpD"	x	['Rianna']			[]
1787	"false"	"child0"	"r3BXQt"		[]			[]
1792	"false"	"child0"	"6WxbBt"		[]			[]
1791	"false"	"child0"	"gQ1KBM"		[]			[]
1795	"true"	"child0"	"BJMZV0"		[]			[]
1796	"true"	"child0"	"kxpP46"		[]			[]

9. What should you write in the video_usable column if you find that the videos for a looking-time study are just audio files (no video)?

10. [Testimony] The child is asked whether something is a brush or a key. She answers 'key… no, silly, brush!' Her mom says 'So what is it really called?' and she repeats 'A brush… do you think it's a key?' What do you write in each column associated with this clip?

11. [Novelverbs, Oneshot] List all the event types you'll need to include in a VCode template file for this study.

12. [Novelverbs] A child cries briefly during clip 3 but briefly recovers. His mom talks to him, soothing him. During trial 4 he doesn't look at the screen at all but seems content. You can't see what he's looking at. During trial 6 his mother closes her eyes. He tries to get off his mother's lap and she stands him up instead, opening her eyes briefly as she does so. He looks at the screen, but then his older brother walks by and blows a raspberry and the baby looks at the brother. During trial 7 the mother's eyes are open as she talks to the older brother, but the baby has stopped attending to their conversation and is watching the screen. Fill in the fuss/distract/parent coding for these clips.

F3	D3	F4	D4	F5	D5	F6	D6	P6	F7	D7	P7

13. [Oneshot] A child is squirming to get out of her father's lap at the start of trial 3, but then calms down. Her dad opens his eyes briefly as he talks to her. During clip 4 the TV is turned on behind them and the child looks over at it. During clip 5 the father gets a phone call and talks briefly to someone. The child looks at the phone when it rings but then looks back at the screen. During clip 6 the child makes some fussy sounds. Fill in the fuss/distract/parent coding for these clips.

AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR
F3	D3	F4	D4	F5	D5	P5	F6	D6	P6

14. [Novelverbs, oneshot] You are coding the video associated with this row. What will your two VCode files be named?

user_id	endedEarly	child_id	recordingSet
1523	"true"	"child0"	"4H6yJl"

15. [Novelverbs, oneshot] How can you delete a 'looking' event in VCode?

16. [Novelverbs, oneshot] What should you do if a child leans forward during part of a trial so that only her forehead is visible? What if this is true throughout most of the entire video?