Experiment 3 Stimuli

Bethany Gardner

06/10/2022

Contents

Introductions to characters

```
d_char <- read.csv("exp3_stim_char.csv",
                   fileEncoding="UTF-8-BOM", stringsAsFactors=TRUE)
str(d_char)
```

```
## 'data.frame':    72 obs. of  15 variables:
##  $ group         : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ trial_id      : Factor w/ 72 levels "both_list1_char1",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ condition     : Factor w/ 4 levels "both","intro",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ character_set : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ target_pronoun: Factor w/ 3 levels "he/him","she/her",..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ target_id     : int  14 8 12 10 2 15 11 7 18 13 ...
##  $ target_image  : Factor w/ 6 levels "HT1","HT2","HT3",..: 2 6 1 1 4 2 1 6 3 2 ...
##  $ target_name   : Factor w/ 6 levels "alex","brian",..: 1 5 3 6 5 2 2 4 6 3 ...
##  $ target_file   : Factor w/ 36 levels "HT1_he_brian.png",..: 8 34 6 4 22 12 2 32 18 10 ...
##  $ brother       : Factor w/ 18 levels "HT1_brother_brian.png",..: 4 18 2 3 12 5 1 17 9 6 ...
##  $ sister        : Factor w/ 18 levels "HT1_sister_brian.png",..: 4 18 2 3 12 5 1 17 9 6 ...
##  $ text1_name    : Factor w/ 6 levels "This is Alex",..: 1 5 3 6 5 2 2 4 6 3 ...
##  $ text1_pronouns: Factor w/ 4 levels ", who uses he/him pronouns.",..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ text2_brother : Factor w/ 3 levels "He has a brother.",..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ text3_sister  : Factor w/ 3 levels "And he has a sister.",..: 1 2 3 1 2 3 1 2 3 1 ...
```

Stimuli file structure

- group = list assignment for PCIbex
- trial_id = unique code across all conditions/lists/experiment sections
- condition = between-subjects conditions (Nametag x Introduction)
- character_set = 6 sets of characters
- target_pronoun = character's pronouns (calling these variables "target" to match other stimuli lists)
- target_id = character code that can reconstruct set, image, name, pronouns
- target_image = character image (HT1, HT2, HT3, ST1, ST2, ST3)
- target_name = character name (Brian, Dan, Emily, Jess, Alex, Sam)
- target_file = file name for character image (image_pronoun_name.png or image_pronoun_name_nametag.png, according to condition)
- brother = file name for character's brother (character-image_brother_character-name.png)
- sister = file name for character's sister (character-image_sister_character-name.png)
- text1_name = text to display with character image ("This is [name]"

- text1_pronouns = text to display with character image (“.” or “, who uses pronouns.”, according to condition)
- text2_brother = text to display with brother image (“He has/she has/they have a brother.”
- text3_sister = text to display with sister image (“And he has/and she has/and they have a sister.”)

Conditions/lists

Split into 4 conditions:

1. Both = +Nametag, +Introduction
2. Intro = -Nametag, +Introduction
3. Nametag = +Nametag, -Introduction
4. Neither = -Nametag, -Introduction

```
summary(d_char$condition)
```

```
##    both   intro nametag neither
##      18      18      18      18
```

Counter-balanced with 6 lists of characters:

```
d_char$character_set %>% as.factor() %>% summary()
```

```
##  1  2  3  4  5  6
## 12 12 12 12 12 12
```

6 sets of characters -> 6 of each pronoun type, each of which appears 1x in each of the 4 conditions.

```
d_char %>% count(target_pronoun, target_id)
```

target_pronoun	target_id	n
he/him	10	4
he/him	11	4
he/him	13	4
he/him	14	4
he/him	16	4
he/him	17	4
she/her	1	4
she/her	2	4
she/her	4	4
she/her	5	4
she/her	7	4
she/her	8	4
they/them	3	4
they/them	6	4
they/them	9	4
they/them	12	4
they/them	15	4
they/them	18	4

4 conditions x 6 character lists = 24 groups, each of which has 3 trials (1 he/him character, 1 she/her character, 1 they/them character).

```
d_char %>% group_by(group) %>% summarise(n())
```

group	n()
1	3
2	3
3	3
4	3
5	3
6	3
7	3
8	3
9	3
10	3
11	3
12	3
13	3
14	3
15	3
16	3
17	3
18	3
19	3
20	3
21	3
22	3
23	3
24	3

Character image

Check to make the image displayed matches the condition/list variables:

```
d_char %<>%
  mutate(.after="target_file", target_path=str_c(
      target_image, "_",
      case_when(
          target_pronoun=="he/him"    ~ "he",
          target_pronoun=="she/her"   ~ "she",
          target_pronoun=="they/them" ~ "they"),
      "_", target_name,
      case_when(
          condition=="both"|condition=="nametag"  ~ "_nametag",
          condition=="intro"|condition=="neither" ~ ""),
      ".png")) %>%
   mutate(.after="target_path",
      target_file_check=(target_file==target_path))

summary(d_char$target_file_check)
```

```
##    Mode    TRUE
## logical      72
```

Character text

Check that text displayed matches the condition/list variables:

"This is [name]"

```
d_char %<>% mutate(.after=14, text1_name_check=
  str_detect(as.character(text1_name),
             str_to_sentence(target_name)))

summary(d_char$text1_name_check)
```

```
##    Mode    TRUE
## logical      72
```

+Introduction conditions: ", who uses pronouns."

-Introduction conditions: "."

```
d_char %<>% mutate(.after="text1_pronouns",
      text1_pronouns_check=case_when(
          (condition=="nametag" | condition=="neither") ~
              text1_pronouns==".",
          (condition=="both" | condition=="intro") ~
              str_detect(as.character(text1_pronouns),
               as.character(target_pronoun))))

summary(d_char$text1_pronouns_check)
```

```
##    Mode    TRUE
## logical      72
```

Sibling images

Check that sibling images displayed match the character variables (image, name):

```
d_char %<>% mutate(.after="brother",
      brother_path=str_c(
          target_image, "_brother_", target_name, ".png")) %>%
      mutate(.after="brother_path",
      brother_check=(brother==brother_path))

summary(d_char$brother_check)
```

```
##    Mode    TRUE
## logical      72
```

```
d_char %<>% mutate(.after="sister",
      sister_path=str_c(
          target_image, "_sister_", target_name, ".png")) %>%
      mutate(.after="sister_path",
      sister_check=(sister==sister_path))

summary(d_char$sister_check)
```

```
##    Mode    TRUE
## logical      72
```

Sibling text

Check that text displayed matches character variables (pronoun).

"[Pronoun] has a brother."

```
d_char$text2_brother %<>% as.character()

d_char %<>% mutate(.after="text2_brother", text2_brother_check=
    ((target_pronoun=="he/him" &&
        str_detect(text2_brother, "He has")) |
    (target_pronoun=="she/her" &&
        str_detect(text2_brother, "She has")) |
    (target_pronoun=="they/them" &&
        str_detect(text2_brother, "They have"))) &&
    str_detect(text2_brother, "a brother."))

summary(d_char$text2_brother_check)
```

```
##    Mode    TRUE
## logical      72
```

"And [pronoun] has a sister."

```
d_char$text3_sister %<>% as.character()

d_char %<>% mutate(.after="text3_sister", text3_sister_check=
    ((target_pronoun=="he/him" &&
        str_detect(text3_sister, "And he has")) |
    (target_pronoun=="she/her" &&
        str_detect(text3_sister, "And she has")) |
    (target_pronoun=="they/them" &&
        str_detect(text3_sister, "And they have"))) &&
  str_detect(text3_sister, "a sister."))

summary(d_char$text3_sister_check)
```

```
##    Mode    TRUE
## logical      72
```

Production prompt displays

Stimuli file structure

```
d <- read.csv("exp3_stimuli_all.csv", stringsAsFactors=TRUE)

str(d)
```

```
## 'data.frame':    864 obs. of  28 variables:
##  $ group              : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ trial_id           : Factor w/ 864 levels "both_list1_example1",..: 1 2 3 37 38 39 73 74 75 109 .
##  $ condition          : Factor w/ 4 levels "both","intro",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ character_set      : int  1 1 1 2 2 2 3 3 3 4 ...
##  $ target_pronoun     : Factor w/ 3 levels "he/him","she/her",..: 1 2 3 1 2 3 1 2 3 1 ...
##  $ target_id          : int  14 8 12 10 2 15 11 7 18 13 ...
##  $ target_image       : Factor w/ 6 levels "HT1","HT2","HT3",..: 2 6 1 1 4 2 1 6 3 2 ...
##  $ target_name        : Factor w/ 6 levels "alex","brian",..: 1 5 3 6 5 2 2 4 6 3 ...
##  $ target_file        : Factor w/ 36 levels "HT1_he_brian.png",..: 8 34 6 4 22 12 2 32 18 10 ...
##  $ target_x           : int  525 275 525 525 275 525 525 275 525 525 ...
##  $ distractor_pronoun : Factor w/ 3 levels "he/him","she/her",..: 2 3 1 2 3 1 2 3 1 2 ...
##  $ distractor_id      : int  8 12 14 2 15 10 7 18 11 5 ...
##  $ distractor_image   : Factor w/ 6 levels "HT1","HT2","HT3",..: 6 1 2 4 2 1 6 3 1 5 ...
##  $ distractor_name    : Factor w/ 6 levels "alex","brian",..: 5 3 1 5 2 6 4 6 2 6 ...
##  $ distractor_file    : Factor w/ 36 levels "HT1_he_brian.png",..: 34 6 8 22 12 4 32 18 2 28 ...
##  $ distractor_x       : int  275 525 275 275 525 275 275 525 275 275 ...
##  $ object_image       : Factor w/ 36 levels "apple.png","avocado.png",..: 13 25 23 13 25 23 13 25 23
##  $ object_x           : int  725 175 725 725 175 725 725 175 725 725 ...
##  $ object_y2          : int  150 250 250 150 150 250 250 150 250 150 ...
##  $ object_y3          : int  100 300 300 100 100 300 300 100 300 100 ...
##  $ object_y4          : int  50 350 350 50 50 350 350 50 350 50 ...
##  $ sibling_image      : Factor w/ 36 levels "HT1_brother_brian.png",..: 10 36 5 3 24 8 1 35 15 9 ...
##  $ top_left           : Factor w/ 36 levels "HT1_brother_brian.png",..: 36 33 7 24 24 6 32 35 1 30 .
##  $ bottom_left        : Factor w/ 36 levels "HT1_brother_brian.png",..: 33 36 10 21 21 3 35 32 4 27
##  $ top_right          : Factor w/ 36 levels "HT1_brother_brian.png",..: 10 5 2 3 11 11 4 18 18 9 ...
##  $ bottom_right       : Factor w/ 36 levels "HT1_brother_brian.png",..: 7 2 5 6 8 8 1 15 15 12 ...
##  $ correct_description: Factor w/ 170 levels "Alex gave the apple to their sister.",..: 13 125 79 14
##  $ audio              : Factor w/ 31 levels "","example_alex-fork-his-sister.mp3",..: 2 13 9 14 13 7
```

Experiment variables:

- group = list assignment for PCIbex
- trial_id = unique code across all conditions/lists/experiment sections
- condition = between-subjects conditions (Nametag x Introduction)
- character_set = 6 sets of characters

Target character (the one described) variables:

- target_pronoun = character's pronouns
- target_id = character code that can reconstruct set, image, name, pronouns
- target_image = character image (HT1, HT2, HT3, ST1, ST2, ST3)
- target_name = character name (Brian, Dan, Emily, Jess, Alex, Sam)
- target_file = file name for character image (image_pronoun_name.png or image_pronoun_name_nametag.png, according to condition)
- target_x = location (275=left, 525=right)

Distractor character (pictured, but not described) variables:

- distractor_pronoun = character's pronouns
- distractor_id = character code that can reconstruct set, image, name, pronouns
- distractor_image = character image (HT1, HT2, HT3, ST1, ST2, ST3)
- distractor_name = character name (Brian, Dan, Emily, Jess, Alex, Sam)
- distractor_file = file name for character image (image_pronoun_name.png or image_pronoun_name_nametag.png, according to condition)
- distractor_x = location (275=left, 525=right)

Object variables:

- object_x: object horizontal location (175=left, 725=right)
- object_y2, object_y3, object_y4: object vertical path. (starts at center, 150 100 50 moves up, 250 300 350 moves down)

Sibling variables:

- sibling_image: sibling that receives object
- top_left, top_right, bottom_left, bottom_right: all 4 sibling images to display

Correct answer:

- correct_description: [name] gave the [object] to [pronoun] sibling
- audio: audio file for example and practice trials

Conditions/lists

Split into 4 conditions:

1. Both = +Nametag, +Introduction
2. Intro = -Nametag, +Introduction
3. Nametag = +Nametag, -Introduction
4. Neither = -Nametag, -Introduction

```
summary(d$condition)
```

```
##    both   intro nametag neither
##     216     216     216     216
```

Counter-balanced with 6 lists of characters:

```
d$character_set %>% as.factor() %>% summary()
```

```
##   1   2   3   4   5   6
## 144 144 144 144 144 144
```

6 sets of characters * 3 pronouns = 18 character variations, each with a unique image-pronoun-name combination.

```
d %>% select(target_id, target_pronoun,
             target_image, target_name) %>%
  unique() %>% arrange(target_id)
```

target_id	target_pronoun	target_image	target_name
1	she/her	ST1	alex
2	she/her	ST1	jess
3	they/them	ST1	emily
4	she/her	ST2	emily
5	she/her	ST2	sam
6	they/them	ST2	jess
7	she/her	ST3	emily
8	she/her	ST3	jess
9	they/them	ST3	alex
10	he/him	HT1	sam
11	he/him	HT1	brian
12	they/them	HT1	dan
13	he/him	HT2	dan
14	he/him	HT2	alex
15	they/them	HT2	brian
16	he/him	HT3	dan
17	he/him	HT3	brian
18	they/them	HT3	sam

4 conditions x 6 character lists = 24 groups, each of which has 3 trials (1 he/him character, 1 she/her character, 1 they/them character). The Nametag, but not the Introduction condition varies here, but need the items repeated to keep group assignment consistent in PCIbex.

36 = 3 example + 3 practice + 30 test trials per group

```
d %>% group_by(group) %>% summarise(n())
```

group	n()
1	36
2	36
3	36
4	36
5	36
6	36
7	36
8	36
9	36
10	36
11	36
12	36
13	36
14	36
15	36
16	36
17	36
18	36
19	36
20	36
21	36
22	36
23	36
24	36

```
d %<>% mutate(.after="group", trial_type=case_when(
  str_detect(trial_id, "example") ~ "example",
  str_detect(trial_id, "practice") ~ "practice",
  str_detect(trial_id, "test") ~ "test"))
summary(as.factor(d$trial_type))
```

```
##  example practice     test
##       72       72      720
```

Each trial has a unique ID, to double-check that no trial info is lost.

```
str(d$trial_id)
```

```
##  Factor w/ 864 levels "both_list1_example1",..: 1 2 3 37 38 39 73 74 75 109 ...
```

```
864/36
```

```
## [1] 24
```

Characters

Pronouns

Each list has 1 example, 1 practice, and 10 test trials for each pronoun type.

```
d %>% count(group, target_pronoun, trial_type)
```

group	target_pronoun	trial_type	n
1	he/him	example	1
1	he/him	practice	1
1	he/him	test	10
1	she/her	example	1
1	she/her	practice	1
1	she/her	test	10
1	they/them	example	1
1	they/them	practice	1
1	they/them	test	10
2	he/him	example	1
2	he/him	practice	1
2	he/him	test	10
2	she/her	example	1
2	she/her	practice	1
2	she/her	test	10
2	they/them	example	1
2	they/them	practice	1
2	they/them	test	10
3	he/him	example	1
3	he/him	practice	1
3	he/him	test	10
3	she/her	example	1
3	she/her	practice	1
3	she/her	test	10
3	they/them	example	1
3	they/them	practice	1
3	they/them	test	10
4	he/him	example	1
4	he/him	practice	1
4	he/him	test	10
4	she/her	example	1
4	she/her	practice	1
4	she/her	test	10
4	they/them	example	1
4	they/them	practice	1
4	they/them	test	10
5	he/him	example	1
5	he/him	practice	1
5	he/him	test	10
5	she/her	example	1
5	she/her	practice	1
5	she/her	test	10
5	they/them	example	1
5	they/them	practice	1
5	they/them	test	10
6	he/him	example	1
6	he/him	practice	1
6	he/him	test	10
6	she/her	example	1
6	she/her	practice	1
6	she/her	test	10
6	they/them	example	1
6	they/them	practice	1
6	they/them	test	10
7	he/him	example	1
7	he/him	practice	1
7	he/him	test	10

The distractor character pronouns are split evenly between the other two pronoun types.

```
d %>% count(distractor_pronoun, target_pronoun)
```

distractor_pronoun	target_pronoun	n
he/him	she/her	144
he/him	they/them	144
she/her	he/him	144
she/her	they/them	144
they/them	he/him	144
they/them	she/her	144

Images

6 character images.

```
summary(d$target_image)
```

```
## HT1 HT2 HT3 ST1 ST2 ST3
## 144 144 144 144 144 144
```

```
summary(d$distractor_image)
```

```
## HT1 HT2 HT3 ST1 ST2 ST3
## 144 144 144 144 144 144
```

Each list contains 3 images.

```
d %>% count(character_set, target_image)
```

character_set	target_image	n
1	HT1	48
1	HT2	48
1	ST3	48
2	HT1	48
2	HT2	48
2	ST1	48
3	HT1	48
3	HT3	48
3	ST3	48
4	HT2	48
4	ST1	48
4	ST2	48
5	HT3	48
5	ST1	48
5	ST2	48
6	HT3	48
6	ST2	48
6	ST3	48

```
d %>% count(character_set, distractor_image)
```

character_set	distractor_image	n
1	HT1	48
1	HT2	48
1	ST3	48
2	HT1	48
2	HT2	48
2	ST1	48
3	HT1	48
3	HT3	48
3	ST3	48
4	HT2	48
4	ST1	48
4	ST2	48
5	HT3	48
5	ST1	48
5	ST2	48
6	HT3	48
6	ST2	48
6	ST3	48

Each image appears 2x with he/she and 1x with they.

```
d %>% count(target_image, target_pronoun)
```

target_image	target_pronoun	n
HT1	he/him	96
HT1	they/them	48
HT2	he/him	96
HT2	they/them	48
HT3	he/him	96
HT3	they/them	48
ST1	she/her	96
ST1	they/them	48
ST2	she/her	96
ST2	they/them	48
ST3	she/her	96
ST3	they/them	48

```
d %>% count(distractor_image, distractor_pronoun)
```

distractor_image	distractor_pronoun	n
HT1	he/him	96
HT1	they/them	48
HT2	he/him	96
HT2	they/them	48
HT3	he/him	96
HT3	they/them	48
ST1	she/her	96
ST1	they/them	48
ST2	she/her	96
ST2	they/them	48
ST3	she/her	96
ST3	they/them	48

Names

6 names (2 masc, 2 fem, 2 androgynous).

```
summary(d$target_name)
```

```
##  alex brian   dan emily  jess   sam
##   144   144   144   144   144   144
```

```
summary(d$distractor_name)
```

```
##  alex brian   dan emily  jess   sam
##   144   144   144   144   144   144
```

Each list has 3 names (1 masc, 1 fem, 1 androgynous).

```
d %>% count(character_set, target_name)
```

character_set	target_name	n
1	alex	48
1	dan	48
1	jess	48
2	brian	48
2	jess	48
2	sam	48
3	brian	48
3	emily	48
3	sam	48
4	dan	48
4	emily	48
4	sam	48
5	alex	48
5	dan	48
5	jess	48
6	alex	48
6	brian	48
6	emily	48

```
d %>% count(character_set, distractor_name)
```

character_set	distractor_name	n
1	alex	48
1	dan	48
1	jess	48
2	brian	48
2	jess	48
2	sam	48
3	brian	48
3	emily	48
3	sam	48
4	dan	48
4	emily	48
4	sam	48
5	alex	48
5	dan	48
5	jess	48
6	alex	48
6	brian	48
6	emily	48

Masculine names appear 2x with he/him and 1x with they/them. Feminine names appear 2x with she/her and 1x with they/them. Androgynous names appear 1x with he/him, 1x with she/her, and 1x with they/them. [1x = (1 practice + 1 example + 10 test) x 4 conditions]

```
d %>% count(target_name, target_pronoun)
```

target_name	target_pronoun	n
alex	he/him	48
alex	she/her	48
alex	they/them	48
brian	he/him	96
brian	they/them	48
dan	he/him	96
dan	they/them	48
emily	she/her	96
emily	they/them	48
jess	she/her	96
jess	they/them	48
sam	he/him	48
sam	she/her	48
sam	they/them	48

```
d %>% count(distractor_name, distractor_pronoun)
```

distractor_name	distractor_pronoun	n
alex	he/him	48
alex	she/her	48
alex	they/them	48
brian	he/him	96
brian	they/them	48
dan	he/him	96
dan	they/them	48
emily	she/her	96
emily	they/them	48
jess	she/her	96
jess	they/them	48
sam	he/him	48
sam	she/her	48
sam	they/them	48

Siblings

Image for sibling that receives object matches character.

```
d %>% mutate(sibling_group=str_sub(sibling_image, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	72
HT1	HT1_sister_	72
HT2	HT2_brother	72
HT2	HT2_sister_	72
HT3	HT3_brother	72
HT3	HT3_sister_	72
ST1	ST1_brother	72
ST1	ST1_sister_	72
ST2	ST2_brother	72
ST2	ST2_sister_	72
ST3	ST3_brother	72
ST3	ST3_sister_	72

Example & practice trials

Odd numbers mean that example+practice trials aren't perfectly balanced, so less confusing to check test trials separately.

Sibling that receives object is distributed evenly.

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  mutate(sibling_group=str_sub(sibling_image, 0, 11)) %>%
  group_by(sibling_group) %>%
  summarise(n=n())
```

sibling_group	n
HT1_brother	12
HT1_sister_	12
HT2_brother	12
HT2_sister_	12
HT3_brother	12
HT3_sister_	12
ST1_brother	12
ST1_sister_	12
ST2_brother	12
ST2_sister_	12
ST3_brother	12
ST3_sister_	12

Target characters that appear on the left (1/3 of example, 2/3 of practice) have matching sibling images.

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(target_x==275) %>%
  mutate(sibling_group=str_sub(top_left, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	4
HT1	HT1_sister_	8
HT2	HT2_brother	8
HT2	HT2_sister_	4
HT3	HT3_brother	4
HT3	HT3_sister_	8
ST1	ST1_brother	4
ST1	ST1_sister_	8
ST2	ST2_brother	8
ST2	ST2_sister_	4
ST3	ST3_brother	4
ST3	ST3_sister_	8

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(target_x==275) %>%
  mutate(sibling_group=str_sub(bottom_left, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	8
HT1	HT1_sister_	4
HT2	HT2_brother	4
HT2	HT2_sister_	8
HT3	HT3_brother	8
HT3	HT3_sister_	4
ST1	ST1_brother	8
ST1	ST1_sister_	4
ST2	ST2_brother	4
ST2	ST2_sister_	8
ST3	ST3_brother	8
ST3	ST3_sister_	4

Target characters that appear on the right (2/3 example, 1/3 of practice trials) have matching sibling images.

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(target_x==525) %>%
  mutate(sibling_group=str_sub(top_right, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	8
HT1	HT1_sister_	4
HT2	HT2_brother	4
HT2	HT2_sister_	8
HT3	HT3_brother	4
HT3	HT3_sister_	8
ST1	ST1_brother	4
ST1	ST1_sister_	8
ST2	ST2_brother	4
ST2	ST2_sister_	8
ST3	ST3_brother	4
ST3	ST3_sister_	8

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(target_x==525) %>%
  mutate(sibling_group=str_sub(bottom_right, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	4
HT1	HT1_sister_	8
HT2	HT2_brother	8
HT2	HT2_sister_	4
HT3	HT3_brother	8
HT3	HT3_sister_	4
ST1	ST1_brother	8
ST1	ST1_sister_	4
ST2	ST2_brother	8
ST2	ST2_sister_	4
ST3	ST3_brother	8
ST3	ST3_sister_	4

Distractor characters that appear on the left (2/3 example, 1/3 of practice trials) have matching sibling images.

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(distractor_x==275) %>%
  mutate(sibling_group=str_sub(top_left, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	8
HT1	HT1_sister_	8
HT2	HT2_brother	8
HT2	HT2_sister_	8
HT3	HT3_brother	8
HT3	HT3_sister_	8
ST1	ST1_brother	4
ST1	ST1_sister_	4
ST2	ST2_brother	4
ST2	ST2_sister_	4
ST3	ST3_brother	4
ST3	ST3_sister_	4

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(distractor_x==275) %>%
  mutate(sibling_group=str_sub(bottom_left, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	8
HT1	HT1_sister_	8
HT2	HT2_brother	8
HT2	HT2_sister_	8
HT3	HT3_brother	8
HT3	HT3_sister_	8
ST1	ST1_brother	4
ST1	ST1_sister_	4
ST2	ST2_brother	4
ST2	ST2_sister_	4
ST3	ST3_brother	4
ST3	ST3_sister_	4

Distractor characters that appear on the right (1/3 of example, 2/3 of practice trials) have matching sibling images.

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(distractor_x==525) %>%
  mutate(sibling_group=str_sub(top_right, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	4
HT1	HT1_sister_	4
HT2	HT2_brother	4
HT2	HT2_sister_	4
HT3	HT3_brother	4
HT3	HT3_sister_	4
ST1	ST1_brother	8
ST1	ST1_sister_	8
ST2	ST2_brother	8
ST2	ST2_sister_	8
ST3	ST3_brother	8
ST3	ST3_sister_	8

```
d %>% filter(trial_type=="example"|trial_type=="practice") %>%
  filter(distractor_x==525) %>%
  mutate(sibling_group=str_sub(bottom_right, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	4
HT1	HT1_sister_	4
HT2	HT2_brother	4
HT2	HT2_sister_	4
HT3	HT3_brother	4
HT3	HT3_sister_	4
ST1	ST1_brother	8
ST1	ST1_sister_	8
ST2	ST2_brother	8
ST2	ST2_sister_	8
ST3	ST3_brother	8
ST3	ST3_sister_	8

Test trials

Sibling that receives object is distributed evenly.

```
d %>% filter(trial_type=="test") %>%
  group_by(sibling_image) %>%
  summarise(n=n())
```

sibling_image	n
HT1_brother_brian.png	20
HT1_brother_dan.png	20
HT1_brother_sam.png	20
HT1_sister_brian.png	20
HT1_sister_dan.png	20
HT1_sister_sam.png	20
HT2_brother_alex.png	20
HT2_brother_brian.png	20
HT2_brother_dan.png	20
HT2_sister_alex.png	20
HT2_sister_brian.png	20
HT2_sister_dan.png	20
HT3_brother_brian.png	20
HT3_brother_dan.png	20
HT3_brother_sam.png	20
HT3_sister_brian.png	20
HT3_sister_dan.png	20
HT3_sister_sam.png	20
ST1_brother_alex.png	20
ST1_brother_emily.png	20
ST1_brother_jess.png	20
ST1_sister_alex.png	20
ST1_sister_emily.png	20
ST1_sister_jess.png	20
ST2_brother_emily.png	20
ST2_brother_jess.png	20
ST2_brother_sam.png	20
ST2_sister_emily.png	20
ST2_sister_jess.png	20
ST2_sister_sam.png	20
ST3_brother_alex.png	20
ST3_brother_emily.png	20
ST3_brother_jess.png	20
ST3_sister_alex.png	20
ST3_sister_emily.png	20
ST3_sister_jess.png	20

Target characters that appear on the left have matching sibling images, split as evenly as possible between brother and sister.

```
d %>% filter(trial_type=="test") %>%
  filter(target_x==275) %>%
  mutate(sibling_group=str_sub(top_left, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	28
HT1	HT1_sister_	32
HT2	HT2_brother	32
HT2	HT2_sister_	28
HT3	HT3_brother	28
HT3	HT3_sister_	32
ST1	ST1_brother	28
ST1	ST1_sister_	32
ST2	ST2_brother	32
ST2	ST2_sister_	28
ST3	ST3_brother	32
ST3	ST3_sister_	28

```
d %>% filter(trial_type=="test") %>%
  filter(target_x==275) %>%
  mutate(sibling_group=str_sub(bottom_left, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	32
HT1	HT1_sister_	28
HT2	HT2_brother	28
HT2	HT2_sister_	32
HT3	HT3_brother	32
HT3	HT3_sister_	28
ST1	ST1_brother	32
ST1	ST1_sister_	28
ST2	ST2_brother	28
ST2	ST2_sister_	32
ST3	ST3_brother	28
ST3	ST3_sister_	32

Target characters that appear on the right have matching sibling images, split as evenly as possible between brother and sister.

```
d %>% filter(trial_type=="test") %>%
  filter(target_x==525) %>%
  mutate(sibling_group=str_sub(top_right, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	32
HT1	HT1_sister_	28
HT2	HT2_brother	32
HT2	HT2_sister_	28
HT3	HT3_brother	32
HT3	HT3_sister_	28
ST1	ST1_brother	28
ST1	ST1_sister_	32
ST2	ST2_brother	28
ST2	ST2_sister_	32
ST3	ST3_brother	32
ST3	ST3_sister_	28

```
d %>% filter(trial_type=="test") %>%
  filter(target_x==525) %>%
  mutate(sibling_group=str_sub(bottom_right, 0, 11)) %>%
  count(target_image, sibling_group)
```

target_image	sibling_group	n
HT1	HT1_brother	28
HT1	HT1_sister_	32
HT2	HT2_brother	28
HT2	HT2_sister_	32
HT3	HT3_brother	28
HT3	HT3_sister_	32
ST1	ST1_brother	32
ST1	ST1_sister_	28
ST2	ST2_brother	32
ST2	ST2_sister_	28
ST3	ST3_brother	28
ST3	ST3_sister_	32

Trials where distractor character is on left have matching sibling images.

```
d %>% filter(trial_type=="test") %>%
  filter(distractor_x==275) %>%
  mutate(sibling_group=str_sub(top_left, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	36
HT1	HT1_sister_	28
HT2	HT2_brother	36
HT2	HT2_sister_	28
HT3	HT3_brother	36
HT3	HT3_sister_	28
ST1	ST1_brother	28
ST1	ST1_sister_	28
ST2	ST2_brother	28
ST2	ST2_sister_	28
ST3	ST3_brother	28
ST3	ST3_sister_	28

```
d %>% filter(trial_type=="test") %>%
  filter(distractor_x==275) %>%
  mutate(sibling_group=str_sub(bottom_left, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	28
HT1	HT1_sister_	36
HT2	HT2_brother	28
HT2	HT2_sister_	36
HT3	HT3_brother	28
HT3	HT3_sister_	36
ST1	ST1_brother	28
ST1	ST1_sister_	28
ST2	ST2_brother	28
ST2	ST2_sister_	28
ST3	ST3_brother	28
ST3	ST3_sister_	28

Trials where distractor character is on right have matching sibling images.

```
d %>% filter(trial_type=="test") %>%
  filter(distractor_x==525) %>%
  mutate(sibling_group=str_sub(top_right, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	32
HT1	HT1_sister_	24
HT2	HT2_brother	32
HT2	HT2_sister_	24
HT3	HT3_brother	32
HT3	HT3_sister_	24
ST1	ST1_brother	32
ST1	ST1_sister_	32
ST2	ST2_brother	32
ST2	ST2_sister_	32
ST3	ST3_brother	32
ST3	ST3_sister_	32

```
d %>% filter(trial_type=="test") %>%
  filter(distractor_x==525) %>%
  mutate(sibling_group=str_sub(bottom_right, 0, 11)) %>%
  count(distractor_image, sibling_group)
```

distractor_image	sibling_group	n
HT1	HT1_brother	24
HT1	HT1_sister_	32
HT2	HT2_brother	24
HT2	HT2_sister_	32
HT3	HT3_brother	24
HT3	HT3_sister_	32
ST1	ST1_brother	32
ST1	ST1_sister_	32
ST2	ST2_brother	32
ST2	ST2_sister_	32
ST3	ST3_brother	32
ST3	ST3_sister_	32

Objects

36 objects, 1 for each trial frame (3 + 3 + 30).

```
d %>% group_by(trial_type, object_image) %>% summarise(n())
```

trial_type	object_image	n()
example	fork.png	24
example	peach.png	24
example	pepper.png	24
practice	hotdog.png	24
practice	popcorn.png	24
practice	sandwich.png	24
test	apple.png	24
test	avocado.png	24
test	bacon.png	24
test	banana.png	24
test	bread.png	24
test	broccoli.png	24
test	carrot.png	24
test	cherries.png	24
test	chocolate.png	24
test	cookie.png	24
test	corn.png	24
test	egg.png	24
test	grapes.png	24
test	icecream.png	24
test	juice.png	24
test	kiwi.png	24
test	knife.png	24
test	lemon.png	24
test	onion.png	24
test	orange.png	24
test	pear.png	24
test	pineapple.png	24
test	pizza.png	24
test	plate.png	24
test	potato.png	24
test	pumpkin.png	24
test	spoon.png	24
test	strawberry.png	24
test	tomato.png	24
test	watermelon.png	24

Check that object starts on same side as target character.

```
d %<>% mutate(.after="object_x", object_x_check=case_when(
  target_x==275 ~ object_x==175,
  target_x==525 ~ object_x==725))
summary(d$object_x_check)
```

```
##    Mode    TRUE
## logical     864
```

Check that object moves up/down to the correct sibling.

```
d %<>% mutate(.after="sibling_image", sibling_pos=case_when(
    sibling_image==top_left ~ "top left",
    sibling_image==top_right ~ "top right",
    sibling_image==bottom_left ~ "bottom left",
    sibling_image==bottom_right ~ "bottom right"))
```

```
d %<>% mutate(.after="object_y4", object_y_check=case_when(
  str_detect(sibling_pos, "top") ~
    (object_y2==150 & object_y3==100 & object_y4==50),
  str_detect(sibling_pos, "bottom") ~
    (object_y2==250 & object_y3==300 & object_y4==350)))

summary(d$object_y_check)
```

```
##    Mode    TRUE
## logical     864
```

Counterbalance positioning

Target character

Target character position (left or right) counterbalanced across 3 example, 3 test, and 30 practice trials.

```
d %>% count(trial_type, target_x)
```

trial_type	target_x	n
example	275	24
example	525	48
practice	275	48
practice	525	24
test	275	360
test	525	360

```
d %>% count(character_set, target_x)
```

character_set	target_x	n
1	275	72
1	525	72
2	275	72
2	525	72
3	275	72
3	525	72
4	275	72
4	525	72
5	275	72
5	525	72
6	275	72
6	525	72

Sibling & object position

Sibling/object counterbalanced. (Just checking test trials b/c can't balance 3 test and 3 practice across 4 positions.)

```
d %>% filter(trial_type=="test") %>%
  group_by(character_set) %>%
  count(sibling_pos)
```

character_set	sibling_pos	n
1	bottom left	32
1	bottom right	28
1	top left	28
1	top right	32
2	bottom left	32
2	bottom right	28
2	top left	28
2	top right	32
3	bottom left	28
3	bottom right	32
3	top left	32
3	top right	28
4	bottom left	28
4	bottom right	32
4	top left	32
4	top right	28
5	bottom left	32
5	bottom right	28
5	top left	28
5	top right	32
6	bottom left	28
6	bottom right	28
6	top left	32
6	top right	32

Correct answer

Equal number of his/her/their brother/sister correct answers per list.

```
d %<>% mutate(.after="sibling_image", sibling_type=case_when(
  str_detect(sibling_image, "brother") ~ "brother",
  str_detect(sibling_image, "sister") ~ "sister"))

d %>% group_by(character_set) %>%
  count(target_pronoun, sibling_type)
```

character_set	target_pronoun	sibling_type	n
1	he/him	brother	24
1	he/him	sister	24
1	she/her	brother	24
1	she/her	sister	24
1	they/them	brother	24
1	they/them	sister	24
2	he/him	brother	24
2	he/him	sister	24
2	she/her	brother	24
2	she/her	sister	24
2	they/them	brother	24
2	they/them	sister	24
3	he/him	brother	24
3	he/him	sister	24
3	she/her	brother	24
3	she/her	sister	24
3	they/them	brother	24
3	they/them	sister	24
4	he/him	brother	24
4	he/him	sister	24
4	she/her	brother	24
4	she/her	sister	24
4	they/them	brother	24
4	they/them	sister	24
5	he/him	brother	24
5	he/him	sister	24
5	she/her	brother	24
5	she/her	sister	24
5	they/them	brother	24
5	they/them	sister	24
6	he/him	brother	24
6	he/him	sister	24
6	she/her	brother	24
6	she/her	sister	24
6	they/them	brother	24
6	they/them	sister	24

Double check correct description ("[Name] gave the [object] to pronoun.")

```
d %<>% mutate(.after="correct_description",
  correct_description_check=
  str_c(str_to_sentence(target_name),
        " gave the ",
        str_sub(object_image, end=-5),
        " to ",
        case_when(
          target_pronoun=="he/him" ~ "his ",
          target_pronoun=="she/her" ~ "her ",
          target_pronoun=="they/them" ~ "their "),
        case_when(
          str_detect(sibling_image, "brother") ~ "brother.",
          str_detect(sibling_image, "sister") ~ "sister."))) %>%
```

```
  mutate(.after="correct_description_check",
         correct_description_compare=
        (correct_description==correct_description_check))
summary(d$correct_description_compare)
```

```
##    Mode    TRUE
## logical     864
```

Correct file names

Target

```
d %<>% mutate(.after="target_file", target_path=
          str_c(target_image, "_",
          case_when(
              target_pronoun=="he/him"    ~ "he",
              target_pronoun=="she/her"   ~ "she",
              target_pronoun=="they/them" ~ "they"),
          "_", target_name,
          case_when(
              condition=="both" | condition=="nametag"  ~ "_nametag",
              condition=="intro" | condition=="neither" ~ ""),
          ".png")) %>%
       mutate(.after="target_path",
              target_path_check=(target_file==target_path))

summary(d$target_path_check)
```

```
##    Mode    TRUE
## logical     864
```

Distractor

```
d %<>% mutate(.after="distractor_file", distractor_path=
          str_c(distractor_image, "_",
          case_when(
              distractor_pronoun=="he/him" ~ "he",
              distractor_pronoun=="she/her" ~ "she",
              distractor_pronoun=="they/them" ~ "they"),
          "_", distractor_name,
          case_when(
              condition=="both"|condition=="nametag" ~ "_nametag",
              condition=="intro"|condition=="neither" ~ ""),
          ".png")) %>%
  mutate(.after="distractor_path",
         distractor_path_check=(distractor_file==distractor_path))

summary(d$distractor_path_check)
```

```
##    Mode    TRUE
## logical     864
```

Sibling

```
d %<>% mutate(.after="sibling_image", sibling_path=str_c(
            target_image, "_", sibling_type, "_",
            target_name, ".png")) %>%
       mutate(.after="sibling_path", sibling_path_check=
            (sibling_image==sibling_path))

summary(d$sibling_path_check)
```

```
##    Mode    TRUE
## logical     864
```

Audio

```
d$audio %<>% as.factor() %>% na_if("")
```

144 example and practice trials have audio; 720 test trials don't.

```
d %<>% mutate(.after="audio", audio_path=case_when(
       trial_type=="example" | trial_type=="practice" ~
         (str_c(trial_type, "_",
                target_name, "-",
                str_sub(object_image, end=-5), "-",
                case_when(
                    target_pronoun=="he/him" ~ "his",
                    target_pronoun=="she/her" ~ "her",
                    target_pronoun=="they/them" ~ "their"),
                "-", sibling_type, ".mp3")),
       trial_type=="test" ~ ""))
d$audio_path %<>% na_if("")

d %<>% mutate(.after="audio_path", audio_path_check=
       (audio==audio_path))
summary(d$audio_path_check)
```

```
##    Mode    TRUE    NA's
## logical     144     720
```

List of stimuli

Images

Full list of image files called in the stimuli list:

```
image_list <- d %>%
  select(target_file, distractor_file, object_image,
    sibling_image, top_left, top_right, bottom_left, bottom_right) %>%
  pivot_longer(cols=everything(),
               names_to="temp",
               values_to="file_name")

image_list2 <- d_char %>%
  select(target_file, brother, sister) %>%
  pivot_longer(cols=everything(),
               names_to="temp",
               values_to="file_name")

image_list %<>% bind_rows(image_list2) %>%
  arrange(file_name) %>%
  select(-temp) %>%
  unique()
```

Check to make sure they're all there:

```
files <- list.files("Images/Characters_300px/", pattern="*.png")
files %<>% append(list.files("Images/Objects_300px/", pattern="*.png"))

image_list %<>% mutate(exist=file_name %in% files)

image_list
```

file_name	exist
HT1_he_brian.png	TRUE
HT1_he_brian_nametag.png	TRUE
HT1_he_sam.png	TRUE
HT1_he_sam_nametag.png	TRUE
HT1_they_dan.png	TRUE
HT1_they_dan_nametag.png	TRUE
HT2_he_alex.png	TRUE
HT2_he_alex_nametag.png	TRUE
HT2_he_dan.png	TRUE
HT2_he_dan_nametag.png	TRUE
HT2_they_brian.png	TRUE
HT2_they_brian_nametag.png	TRUE
HT3_he_brian.png	TRUE
HT3_he_brian_nametag.png	TRUE
HT3_he_dan.png	TRUE
HT3_he_dan_nametag.png	TRUE
HT3_they_sam.png	TRUE
HT3_they_sam_nametag.png	TRUE
ST1_she_alex.png	TRUE
ST1_she_alex_nametag.png	TRUE
ST1_she_jess.png	TRUE
ST1_she_jess_nametag.png	TRUE
ST1_they_emily.png	TRUE
ST1_they_emily_nametag.png	TRUE
ST2_she_emily.png	TRUE
ST2_she_emily_nametag.png	TRUE
ST2_she_sam.png	TRUE
ST2_she_sam_nametag.png	TRUE
ST2_they_jess.png	TRUE
ST2_they_jess_nametag.png	TRUE
ST3_she_emily.png	TRUE
ST3_she_emily_nametag.png	TRUE
ST3_she_jess.png	TRUE
ST3_she_jess_nametag.png	TRUE
ST3_they_alex.png	TRUE
ST3_they_alex_nametag.png	TRUE
apple.png	TRUE
avocado.png	TRUE
bacon.png	TRUE
banana.png	TRUE
bread.png	TRUE
broccoli.png	TRUE
carrot.png	TRUE
cherries.png	TRUE
chocolate.png	TRUE
cookie.png	TRUE
corn.png	TRUE
egg.png	TRUE
fork.png	TRUE
grapes.png	TRUE
hotdog.png	TRUE
icecream.png	TRUE
juice.png	TRUE
kiwi.png	TRUE
knife.png	TRUE
lemon.png	TRUE
onion.png	TRUE

Audio

List of audio files for example and practice trials.

```
audio_list <- d %>% select(audio) %>%
  drop_na() %>% unique()
```

Check to make sure they're all there:

```
recordings <- list.files("Audio/", pattern="*.mp3")

audio_list %<>% mutate(exist=audio %in% recordings)

audio_list
```

	audio	exist
1	example_alex-fork-his-sister.mp3	TRUE
2	example_jess-pepper-her-sister.mp3	TRUE
3	example_dan-peach-their-sister.mp3	TRUE
4	example_sam-fork-his-brother.mp3	TRUE
6	example_brian-peach-their-brother.mp3	TRUE
7	example_brian-fork-his-brother.mp3	TRUE
8	example_emily-pepper-her-sister.mp3	TRUE
9	example_sam-peach-their-brother.mp3	TRUE
10	example_dan-fork-his-brother.mp3	TRUE
11	example_sam-pepper-her-sister.mp3	TRUE
12	example_emily-peach-their-brother.mp3	TRUE
14	example_alex-pepper-her-sister.mp3	TRUE
15	example_jess-peach-their-brother.mp3	TRUE
16	example_brian-fork-his-sister.mp3	TRUE
18	example_alex-peach-their-brother.mp3	TRUE
73	practice_alex-sandwich-his-brother.mp3	TRUE
74	practice_jess-popcorn-her-brother.mp3	TRUE
75	practice_dan-hotdog-their-brother.mp3	TRUE
76	practice_sam-sandwich-his-sister.mp3	TRUE
78	practice_brian-hotdog-their-sister.mp3	TRUE
79	practice_brian-sandwich-his-sister.mp3	TRUE
80	practice_emily-popcorn-her-brother.mp3	TRUE
81	practice_sam-hotdog-their-sister.mp3	TRUE
82	practice_dan-sandwich-his-sister.mp3	TRUE
83	practice_sam-popcorn-her-brother.mp3	TRUE
84	practice_emily-hotdog-their-sister.mp3	TRUE
86	practice_alex-popcorn-her-brother.mp3	TRUE
87	practice_jess-hotdog-their-sister.mp3	TRUE
88	practice_brian-sandwich-his-brother.mp3	TRUE
90	practice_alex-hotdog-their-sister.mp3	TRUE