

Storyline Build Guide: Interchangeable Drag-and-Drop with Per-Blank Feedback

Activity goal: Learner drags 4 terms (Apple, Apple, Orange, Orange) into 4 blanks. Blanks 1 & 3 accept either Apple; Blanks 2 & 4 accept either Orange. On Submit, each blank shows a tick or cross, plus an overall Correct/Incorrect layer.

Why custom logic is needed: Storyline's built-in drag-and-drop grading maps each drag item to exactly one target, so it cannot accept "either Apple in either apple blank." We use the freeform only for drag/snap behavior and handle grading ourselves with variables and a custom Submit button.

STEP 1 — Create the slide objects

Name everything exactly as below (Home > arrange via the timeline panel):

Object	Name	Notes
Drag item 1	Apple_1	Text/shape reading "Apple"
Drag item 2	Apple_2	Text/shape reading "Apple"
Drag item 3	Orange_1	Text/shape reading "Orange"
Drag item 4	Orange_2	Text/shape reading "Orange"
Drop target 1	Blank_1	Expects Apple
Drop target 2	Blank_2	Expects Orange
Drop target 3	Blank_3	Expects Apple
Drop target 4	Blank_4	Expects Orange
Background catcher	BG_Catch	Transparent rectangle covering the whole slide, sent to back (right-click > Send to Back)
Submit button	Submit	Custom button (do NOT use the built-in player Submit)
Feedback icon 1	Icon_B1	Small shape next to Blank_1
Feedback icon 2	Icon_B2	Next to Blank_2
Feedback icon 3	Icon_B3	Next to Blank_3
Feedback icon 4	Icon_B4	Next to Blank_4

Also create **two slide layers**: **Correct** and **Incorrect** (the Incorrect layer contains a **Try Again** button).

STEP 2 — Set up the freeform (drag behavior only)

1. Insert > Convert to Freeform > **Drag and Drop**.
 2. In **Form View**, map (this mapping is arbitrary — it only registers items/targets; it is never used for grading):
 - Apple_1 → Blank_1
 - Apple_2 → Blank_3
 - Orange_1 → Blank_2
 - Orange_2 → Blank_4
 3. Open **Drag & Drop Options** and set:
 - ✓ Snap dropped items to drop target → **Snap to center**
 - ✓ Allow only one item in each drop target
 - ✓ Return item to start point if dropped outside a correct drop target
 4. In the Triggers panel, **DELETE the default "Submit interaction" trigger**. We replace it entirely.
-

STEP 3 — Create the variables

Manage Project Variables (the "(x)" icon in the Triggers panel):

Variable	Type	Default	Purpose
Apple1_Pos	Number	0	Which blank Apple_1 sits in (0 = none)
Apple2_Pos	Number	0	Which blank Apple_2 sits in
Orange1_Pos	Number	0	Which blank Orange_1 sits in
Orange2_Pos	Number	0	Which blank Orange_2 sits in
AppleOK	True/False	False	Both apple blanks correct
OrangeOK	True/False	False	Both orange blanks correct

STEP 4 — Add icon states

For each of **Icon_B1**–**Icon_B4**, open the State editor (Edit States) and create:

State	Content
Normal (initial)	Empty / fully transparent
Correct	Green tick ✓ image
Incorrect	Red cross ✗ image

Optional cosmetic states (no triggers required — Storyline handles these automatically):

- On drag items: **Drag Over** state for hover feedback.
- On blanks: **Hover** state (e.g., highlighted border) shown while an item is dragged over.

STEP 5 — Position-tracking triggers (20 triggers)

5A. Drop-on-blank triggers (16)

For every drag item × every blank, create:

Set [variable] to value [N] — When: User drops [item] on [Blank_N]

#	Trigger
1–4	Set Apple1_Pos = 1 / 2 / 3 / 4 when user drops Apple_1 on Blank_1 / Blank_2 / Blank_3 / Blank_4
5–8	Set Apple2_Pos = 1 / 2 / 3 / 4 when user drops Apple_2 on Blank_1 / Blank_2 / Blank_3 / Blank_4
9–12	Set Orange1_Pos = 1 / 2 / 3 / 4 when user drops Orange_1 on Blank_1 / Blank_2 / Blank_3 / Blank_4
13–16	Set Orange2_Pos = 1 / 2 / 3 / 4 when user drops Orange_2 on Blank_1 / Blank_2 / Blank_3 / Blank_4

5B. Drop-outside reset triggers (4)

Clears a stale position when the learner pulls an item OUT of a blank and drops it on empty space:

Trigger

17 Set (Apple1_Pos) = 0 when user drops (Apple_1) on (BG_Catch)

18 Set (Apple2_Pos) = 0 when user drops (Apple_2) on (BG_Catch)

19 Set (Orange1_Pos) = 0 when user drops (Orange_1) on (BG_Catch)

20 Set (Orange2_Pos) = 0 when user drops (Orange_2) on (BG_Catch)

⚠ **Test this in your Storyline version:** BG_Catch is not a registered drop target, so items still snap back to their start point — the trigger only clears the variable. If the "drops on" trigger does not fire on a non-target shape in your version, register BG_Catch as a target in Form View and switch placement to "Free."

STEP 6 — Submit button triggers (ORDER IS CRITICAL)

Triggers run top-to-bottom. Arrange the Submit button's triggers in **exactly this order** using the up/down arrows in the Triggers panel:

6A. Reset evaluation variables (2)

1. Set (AppleOK) = **False** — When user clicks (Submit)
2. Set (OrangeOK) = **False** — When user clicks (Submit)

6B. Evaluate pairs (4)

3. Set (AppleOK) = **True** — When user clicks (Submit) — **IF** (Apple1_Pos) == 1 **AND** (Apple2_Pos) == 3
4. Set (AppleOK) = **True** — When user clicks (Submit) — **IF** (Apple1_Pos) == 3 **AND** (Apple2_Pos) == 1
5. Set (OrangeOK) = **True** — When user clicks (Submit) — **IF** (Orange1_Pos) == 2 **AND** (Orange2_Pos) == 4
6. Set (OrangeOK) = **True** — When user clicks (Submit) — **IF** (Orange1_Pos) == 4 **AND** (Orange2_Pos) == 2

6C. Per-blank tick/cross icons (8)

Blank 1 (expects Apple): 7. Change state of (Icon_B1) to **Correct** — When user clicks (Submit) — **IF** (Apple1_Pos) == 1 **OR** (Apple2_Pos) == 1 8. Change state of (Icon_B1) to **Incorrect** — When user clicks (Submit) — **IF** (Apple1_Pos) ≠ 1 **AND** (Apple2_Pos) ≠ 1

Blank 2 (expects Orange): 9. Change state of (Icon_B2) to **Correct** — **IF** (Orange1_Pos) == 2 **OR** (Orange2_Pos) == 2 10. Change state of (Icon_B2) to **Incorrect** — **IF** (Orange1_Pos) ≠ 2 **AND**

≠ 2

Blank 3 (expects Apple): 11. Change state of to **Correct** — IF == 3 OR == 3 12. Change state of to **Incorrect** — IF ≠ 3 AND ≠ 3

Blank 4 (expects Orange): 13. Change state of to **Correct** — IF == 4 OR == 4 14. Change state of to **Incorrect** — IF ≠ 4 AND ≠ 4

Note the logic: **Correct** uses *OR with ==*; **Incorrect** uses *AND with ≠* (the exact logical opposite). The Incorrect condition also catches an **empty blank** or a wrong-fruit drop.

6D. Show feedback layer (2 — must be LAST)

15. Show layer **Correct** — When user clicks — IF == True AND == True
 16. Show layer **Incorrect** — When user clicks — IF == False OR == False
-

STEP 7 — Layer settings (so icons stay visible)

For BOTH the and layers, open Layer Properties (gear icon on the layer) and:

- **Uncheck "Hide objects on base layer"** — otherwise the tick/cross icons vanish when the layer appears.
 - Design each layer as a feedback banner (e.g., at the bottom of the slide) that does NOT cover the blanks/icons. Avoid a full-screen dim overlay.
-

STEP 8 — Try Again button (on the Incorrect layer)

In this order:

1. Change state of to **Normal** — When user clicks
2. Change state of to **Normal** — When user clicks
3. Change state of to **Normal** — When user clicks
4. Change state of to **Normal** — When user clicks
5. **Hide layer** — When user clicks

Dropped items stay where they are, so the learner only needs to fix the crossed blanks.
(Alternative: trigger "Jump to this slide" with slide properties set to **Reset to initial state** for a full clean restart.)

STEP 9 — Test checklist

Preview the slide and verify:

- ☐ Each item snaps into a blank; only one item per blank.
 - ☐ Apple in Blank 1 + Apple in Blank 3 (either order) + Oranges in 2 & 4 → all ticks + Correct layer.
 - ☐ Swap the two Apples or the two Oranges → still fully Correct (interchangeability works).
 - ☐ Any wrong placement → cross on that blank + Incorrect layer; correct blanks still show ticks.
 - ☐ Leave a blank empty and Submit → that blank shows a cross.
 - ☐ Drag an item out of a blank onto empty space, Submit → that blank shows a cross (verifies BG_Catch reset).
 - ☐ Try Again clears all icons, keeps items in place, learner can re-submit.
-

Optional extensions

- **LMS scoring:** This custom-graded slide won't report to an LMS by itself. Add an off-stage "Pick One" freeform with two hidden buttons; trigger-select the "correct" button when the Correct layer shows, then submit that interaction.
 - **Attempts limit:** Add an `Attempts` number variable, add 1 on each Submit, and show a "Review answer" layer instead of Try Again when `Attempts` == 2.
 - **States on blanks instead of icons:** The same conditions can drive Correct/Incorrect states (green/red border + corner tick/cross) directly on the Blank shapes.
 - **Audio feedback:** "Play media when state of Icon_Bx changes to Incorrect."
-

Totals: 14 objects (+2 layers) · 6 variables · ~40 triggers (20 tracking, 16 on Submit, 5 on Try Again — minus any optional items).