

Pay-With-Cash Subtask: User Walkthrough

Overview

1. Receipt Display

Shows an itemized list of four grocery items plus their prices, concluding with a target total.

2. Wallet Section

Displays a finite set of bills and coins that the user can drag into the pay area.

3. Pay Area

A designated grey box where the user drags currency to build a payment (unseen running total).

4. Buttons

- **Submit:** Finalizes the payment and checks if exact change was given.
- **Give Up:** User can signal they cannot figure out the exact change; the system then finishes for them.

5. Timed Assistance

If the user remains idle, a series of cues appear to help them (verbal, constructive, gestural, demonstration).

1 Launching the Task

1. Run the Python script (e.g., `python3 Shopping_PayWithCash_Subtask.py`).
2. A new window appears containing:
 - A **Receipt** panel at the top left.
 - A **Wallet** section along the bottom left.
 - A **Pay Area** on the bottom right.
 - Two buttons: **Submit** (to the right of the pay area) and **Give Up** (somewhere left or above the pay area).





Local brand Chicken Noodle soup \$0.69
Local brand Tomato soup \$0.39
Campbell's Tomato Rice soup \$0.79
Tomato sauce \$0.45

TOTAL: \$2.32

Drag Here To Pay

Give Up

Submit



Time: 119s Attempts: 3

2 Receipt Panel

- **Appearance**

A grey-shaded rectangle labeled “Receipt.” Inside, four lines list each grocery item name and price. Below those lines, “TOTAL: \$X.XX” shows the target amount to pay.

- **Purpose**

The user must provide **exact change equal to that total**.

3 Wallet Section

- **Appearance**

Multiple bill and coin sprites arranged in a compact grid at the bottom-left.

- Bills: One \$5 bill, five \$1 bills.

- Coins: Four quarters, ten dimes, ten nickels, ten pennies.

Each sprite clearly shows its denomination (e.g., “\$1”, “25¢”).

- **Interaction**

- **Click & Drag:** Press on any bill or coin to “pick it up.”

- **Move:** While holding the mouse button down, drag the sprite anywhere.

- **Release:** Let go to drop it. If you drop inside the pay area, it “counts” toward the hidden running total. If you remove a sprite from the pay area, the system registers it as an extraneous move.

4 Pay Area

- **Appearance**

A large grey box labeled “Drag Here To Pay.” It sits on the bottom-right.

- **Purpose**

Any bill or coin released within this box is counted toward the hidden payment total.

There is **no on-screen display** of how much has been accumulated; the user must infer by matching it to the receipt’s total.

- **Interaction**

- Drag a sprite into this box and release: that sprite’s value adds to the hidden total.

- Drag a sprite out of this box: the sprite’s value is removed, and the system logs it as an extraneous move.

5 Buttons

5.1 Submit Button

- **Appearance**
A green button labelled **Submit** located just below or adjacent to the pay area.
- **Function**
Clicking **Submit** invokes `_handle_submit()`.
 - If the hidden running total exactly matches the receipt total, the task ends successfully.
 - Otherwise, it ends as a failure, and the system calculates a quality score based on how close the payment was.

5.2 Give Up Button

- **Appearance**
A red button labelled **Give Up** located somewhere to the left or above the pay area.
- **Function**
Clicking **Give Up** invokes `_handle_surrender()`.
 - The system compares how far off the payment is from the target.
 - Depending on that difference, it sets `assist_level_used` to 7, 8, or 9 (all “take-over” levels), sets `process_score = 0`, and ends the task as a failure.
 - No further hints are shown once this button is clicked.

6 Drag-and-Drop Mechanics

Users move bills and coins between the wallet and the pay area to build their payment. Everything happens visually—no numbers appear during these moves.

- **Picking Up an Item**
 - Hover over a bill or coin in the wallet until you see the cursor change (indicating it can be grabbed).
 - Click and hold to “pick up” that currency. The sprite follows your cursor naturally.
- **Dragging Into the Pay Area**
 - While still holding down the mouse button, drag the item into the grey “Pay” box.

- When you release, that bill or coin stays inside the pay area. Its value now contributes to your total (though you can't see the running sum).

- **Dragging Out of the Pay Area**

- If you change your mind, click a bill or coin that's already in the pay area.
- Drag it back outside the grey box and release. That removal is counted as an extra, "trial-and-error" move.
- The system quietly subtracts that value from your hidden total.

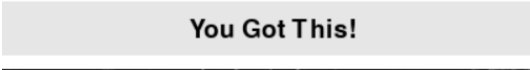
7 Timed Assistance Cues

If the user remains inactive (no dragging or clicking) for a few seconds, the game steps in to offer help:

- **After about 3 seconds of inactivity ("verbal supportive cue"):**

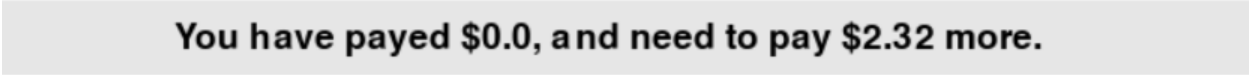
A banner appears at the bottom saying,
"You Got This!"

This gentle encouragement indicates the system is noticing the user hasn't interacted yet.

- **After another seconds total;**  **couple of seconds (≈ 8 "verbal directive cue"):**
The banner updates to something like,

"You have paid \$A.BC, and need \$D.EF more."

Here, \$A.BC is the amount the user has already placed into the pay area, and \$D.EF is what remains to match the receipt. This tells the user precisely how much more money they need to drag to the payment area

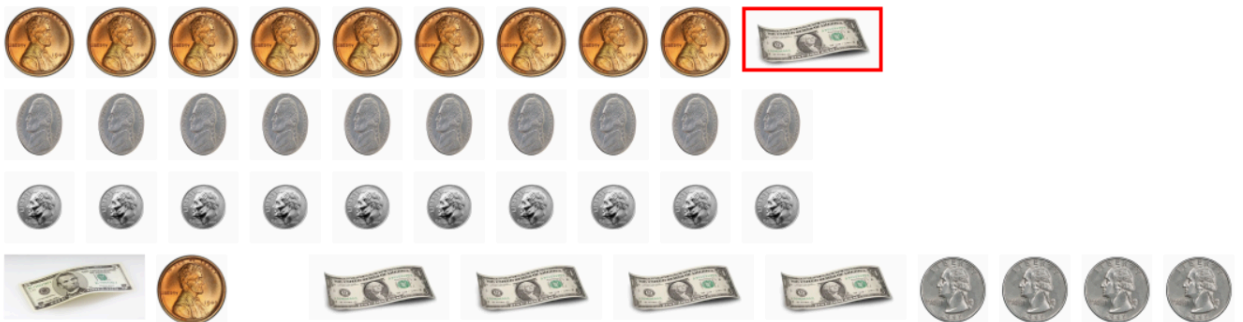


- **After another couple of seconds more (≈ 13 seconds total; "gestural supportive cue"):**

One specific bill or coin is outlined in red. That highlighting shows, "This denomination will help you get closer to the target." For example, if \$1.13 still remains, you might see the \$1 bill or the quarter highlighted in red, suggesting it's a helpful next step.



- **After another couple of seconds (≈ 18 seconds total; “gestural directive cue”):**
The highlighted bill or coin jumps to the spot in the wallet that sits closest to the pay area. This physical movement is a visual nudge, making it easier for the user to drag that item next.

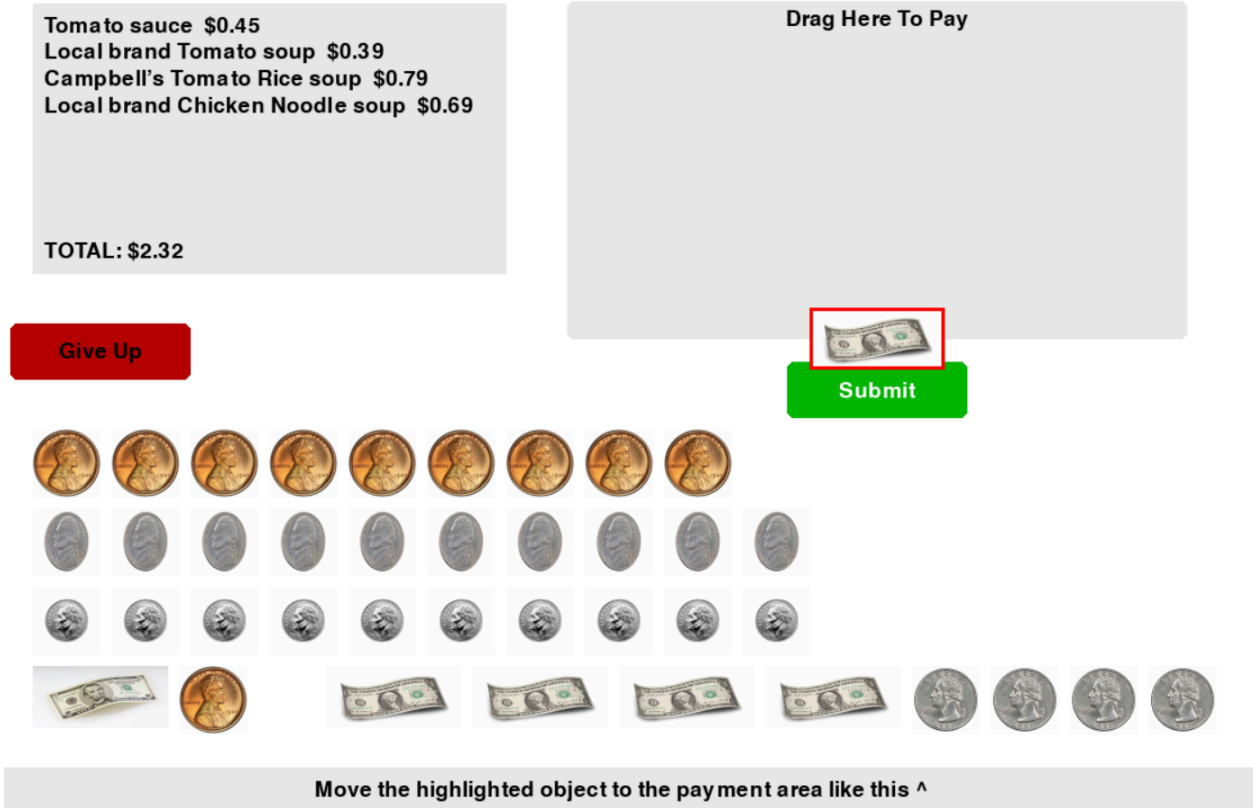


- **After another couple of seconds (≈ 23 seconds total; “partial task demonstration”):**
A banner appears reading,

“Move the highlighted object to the payment area like this ↑.”

At the same time, the highlighted coin or bill begins sliding partway toward the pay area—but it doesn’t drop in yet. This demonstrates the motion without completing it,

inviting the user to finish the action themselves.



- **After another couple of seconds (≈ 28 seconds total; “complete task demonstration”):**
That same bill or coin automatically slides fully into the pay area and stays there, as if the system has done it for the user. At this point, the assistance levels have gone from mere suggestions to actually performing the task step.

8 User-Driven Actions

- **Dragging Currency In**
 1. Click on any bill or coin in the wallet.

2. Hold down the mouse button and drag it into the grey pay area.
 3. Release the mouse button to drop it. That item's value is now counted toward the hidden payment total.
- **Dragging Currency Out**
 1. Click on a bill or coin that is already inside the pay area.
 2. Drag it back outside the pay area.
 3. Release to remove it from the hidden total. Each removal is also recorded as an "extra move," since it indicates the user is adjusting their choices.
 - **Clicking Submit**

When the user believes they have exactly matched the receipt's total, they click the green **Submit** button. At that moment:

 1. The game checks whether the sum of all bills and coins in the pay area exactly equals the receipt's total.
 2. If it matches perfectly, the task ends as a success.
 3. If it does not match, the task ends as a failure, and the game calculates how close the payment was.
 - **Clicking Give Up**

At any point—whether the pay area is empty, partially filled, or even exactly correct—the user can choose to click the **Give Up** button in red. When that happens:

 1. The game compares how far off their current payment is from the target total.
 2. It then assigns a "take-over" level of assistance (depending on that difference) and sets the process score to zero, since the user has asked for help rather than completing the task.
 3. The task ends as a failure, and the result data reflects that the user gave up rather than submitting their own solution.

9 Result Data & Timings

When the subtask finishes—either by **Submit**, **Give Up**, or **Timing Out**—the program packages several pieces of information to summarize what happened:

- **Payment Given:** The final sum of bills and coins that landed in the pay area (hidden from the user until now).
- **Target Total:** The receipt's required total that the user was aiming for.
- **Drag Events:** How many times the user picked up or clicked on a bill/coin.
- **Extraneous Moves:** The count of how many times a bill/coin was placed into the pay area and then removed (indicating trial-and-error).
- **Duration:** The elapsed time from when the task started until it ended.
- **Independence Level:** A number (0–9) showing the highest tier of assistance that occurred—0 means no help was ever given; 7–9 means the user gave up or the system took over.
- **Quality Score:** A 0–3 rating reflecting how close the final payment was (3 if exact, lower if off by small to large amounts).
- **Process Score:** A 0–3 rating based on how many hints or demonstrations took place; it drops to zero if the user clicked **Give Up**.
- **Success Flag:** A simple “yes/no” indicator of whether the final payment exactly matched the receipt.