Ballistic Correcting Sticky Cursor

Members:

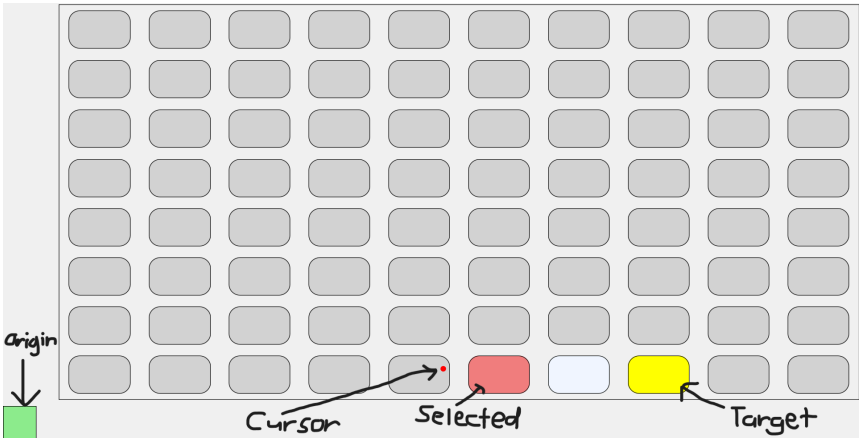
Thai La - vtl932 & Randall Sungcang - rts063

Interaction Problem:

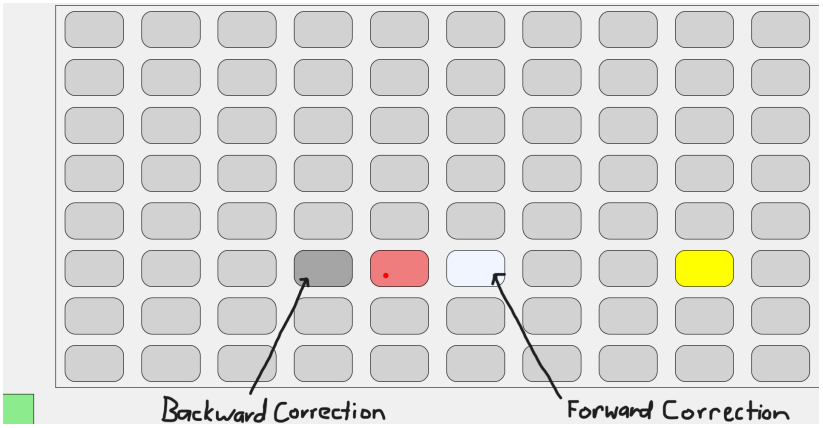
- The goal of this project is to solve the problem of incorrect ballistic motion using sticky targets and target correction. The domain will involve a physical mouse device and a virtual mouse cursor that will represent the user's mouse movements as well as items of interest present in the system.

Interaction Techniques:

- The interaction techniques that we've implemented are Ballistic Motion alongside Sticky Targets and Predictive Corrections in our program
- **Ballistic Motion & Sticky Targets**:
    - The items in the trial will have a "sticky" property to it that will decrease the movement speed of the simulated cursor, making it difficult for the cursor to leave the item's area.
        - To offset the sticky property, the user can use the Ballistic Motion technique to temporarily turn off the item's sticky property, this property will turn back on once the user exits the Ballistic Motion phase.
    - To use the Ballistic Motion, the user will have to move the cursor in a "flick" like motion or simply move the mouse at a fast speed, when the cursor's speed exceeds a certain specified threshold, then the program will enter the Ballistic Motion phase.
        - The cursor's speed is calculated by taking distance divided by time where distance is the current cursor position subtracted by previous position and time is a constant interval defined by the hyper parameter PERIOD_MS.
    - The two hyper parameters that determines the Ballistic Motion Phase are PERIOD_MS and BALLISTIC_SPEED:
        - PERIOD_MS is a unit interval we use to calculate quantized distance
            - Higher PERIOD_MS = requiring more of a "flick" movement to enter the Ballistic Motion Phase.
        - BALLISTIC_SPEED is the threshold for determining ballistic motion phase

        - Lower BALLISTIC_SPEED = lower threshold to enter the Ballistic Motion Phase.
- **Predictive Correction**:
    - Whenever the user selects an item that is not the target, the system will then highlight two items if possible.
    - The Forward Prediction item will be highlighted sky blue while the Backward Prediction item will be highlighted dark gray.
    - 

    - For **Forward** Correction:
        - The Forward Prediction makes use of how the items are stored in a grid like structure to highlight the nearest available item relative from the selected item's location.
        - When the user jumps to the Forward Prediction item, the item will be selected and the simulated cursor will move to the item, the previous item will then be added to the Backward Correction stack.
    - For **Backward** Correction:
        - After the user moves the cursor, the items that were passed over by the simulated cursor are then stored in a stack like structure.
        - When the user jumps to the Backward Correction item, the item will be selected and will be then taken off the stack, the simulated cursor will move to the item.
            - The new Backward Correction item chosen will be the new item on top of the stack.

- 

Technical Overview:

- **Model Composition**
  - Our program is coded using the JavaFX language and using Gitlab for repository.
  - It consists of a controller, multiple views and classes that represent the objects used in the program.
- **Software Approach**
  - We've decided to assign separate tasks to each group member, group members working together for debugging and the more difficult problems.
  - Consistent communication in regards to the project in terms of progress or issues uncovered.
  - Meeting for project planning, co-coding to improve progress.
- **Algorithms Used**
  - Forward prediction: we used atan2 to determine the direction of the user's mouse movement from current position minus previous position. Then the direction is quantized into 45 degree increments and converted into coordinates of the predicted item.
  - Back prediction: we used a specialized modified stack that adds an item to the stack when the user passes over the item with the cursor. The top item in the stack is the previous item. There is also another special internal stack variable called queued which represents the current item that the mouse is on, which is added to the top of the stack once another item is queued.
- **Libraries Used**
  - Mainly JavaFX libraries and Java Util libraries

Evaluation:

- **Evaluation Goals:**
    - To see if our program and the interaction techniques implemented works as intended.
    - To uncover possible bugs in the program.
    - To figure out what else could be done to improve the program.
- Our test evaluation makes use of triangulation, which consists of experiments, questionnaires and observational evaluation methods
- **Experiments:**
    - Record the Fitts' ID of each trial.
    - Track the total time it took for the user to select the correct item.
        - Report the time it took the user to select the targeted item each trial.
        - Track the average time the user took to complete all trials.
    - Track the usage of the Correction technique.
        - Report the number of times the user had to use the Correction Technique each trial.
        - Track the average number times the user required the use of Correction for all trials.
    - Report the wrong selections before the user is able to reach and select the targeted item.
        - Track the average number of errors committed by the user
    - Track the user's performance over time.
        - Use the information from the previous experimental methods to learn the program's learning curve.
- **Questionnaires:**
    - Questions on task difficulty (1 being the easiest, 5 being the hardest).
        - Ask the user of their perceived effort on the trials when using the system.
    - Learning difficulty (1 being the easiest, 5 being the hardest)
        - Ask the user whether it became easier or more difficult to understand the program's functions as the trials went on.
        - If the trials became easier once they became comfortable with the program's interaction techniques.
    - Comments on usability and improvements

        - Ask the user if there were anything apart from the interaction techniques that hindered them from completing the trials properly.
        - Their opinions on what could improve their performance in completing the trials.
        - What could help support the specified interaction techniques in the program.
        - What could improve on the overall program and the layout of the trials.
        - If the assistance of the specified interaction techniques were noticeable.
    - Open-ended comments for the system
- **Observational:**
    - Observing the user as they complete the trials, record the actions taken by the user and how they interacted with the system.
    - Observe the effort that the user exerted and their facial expressions during the trials.

Conclusion:

- **Conclusion From Evaluation**
    - The learning curve of the program is not steep and the user understands the program's mechanics rather quickly.
        - However when the user is trying out the program for the first time, there is confusion and the user visibly struggles especially when trying to maneuver through the sticky targets.
        - But once the user has a better understanding of the program's techniques, the trials progress much smoother and completion time improves.
    - We noticed that users tend to only use correction in the first few trials as they adjust to the system's ballistic motion and sticky target feature. Once they adjust, they do not use the correction feature as often.
- Overall the program works as expected, the "sticky" property makes it difficult for the cursor to escape the item while the ballistic motion allows the user to ignore the "sticky" property of items.
    - The Backwards Correction properly selects from a previously passed over item.
    - The Forward Correction properly predicts the correct item based on the cursor's velocity.