Developing An Algorithm for Runtime Gameplay Adjustments in A Mobile Cognitive Assessment

Penn Memory Center
University of Pennsylvania Health System

Authors: Williams, A., Vadala, J., Tim D., Mechanic-Hamilton D.

Research Intern: Amehja Williams, SEAS 2026, amehjaw@seas.upenn.edu

Faculty Mentor: Dr. Dawn Mechanic-Hamilton, Hospital of the University of Pennsylvania, Department of Neurology

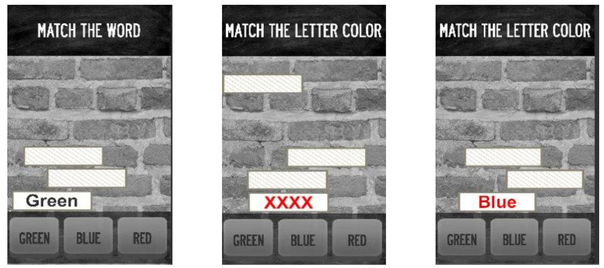
Program Sponsor: Penn Undergraduate Research Mentoring Program (PURM)

Penn CURF Center for Undergraduate Research & Fellowships
PURM Penn Undergraduate Research Mentoring Program

Introduction

- The mCAPP study –
  - **The What**: A mobile cognitive app performance platform delivering cognitive assessments to those at risk for Alzheimer's Disease and Related Dementias (ADRDs) in order to detect associated cognitive changes.
  - **The How**: The mCAPP mobile app contains three minigames aimed to assess memory and executive functioning skills, one of which is the Brick Drop Game.
    - Brick Drop: A processing speed and executive functioning Stroop-like task with three variations:

Word Reading | Color Identification | Color-word Mismatch



Research Objective: Design and optimize an algorithm to adjust game speed (and thus difficulty) of Brick Drop during runtime according to user interaction such that the adjustment is challenging but not jarring.

Methods

- Step 1 - **Mockup & Integration**: Developed Unity demo which calculates basic average reaction time over "N" number of clicks and integrated mockup into Brick Drop code base.

Hypothesis: Updating queue speed based on an average reaction time calculated over 4 clicks will be the least noticeable whereas updating it based on an average reaction time calculated over 6 or more clicks will be the most noticeable.

- Step 2 – **UX Testing**: 2 research co-workers (testers) played this new version of Brick Drop six different times (6 trials), each time a different number of clicks were used to calculate average reaction time. Trials were delivered in random order to avoid bias.

- Step 3 – **UX Questionnaire**: After each trial, testers answered a questionnaire to rate the difficulty and noticeability of the change in gameplay.

Methods Cont.

Mockup & Integration Breakdown

- **Mockup Development**
  - Mockup algorithm calculates average reaction times as the average of the differences between consecutive mouse click time stamps.

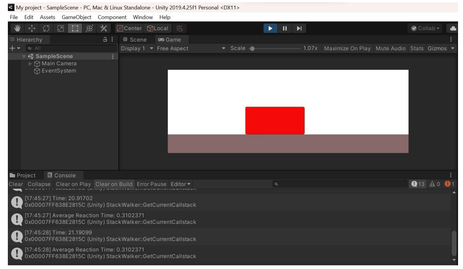


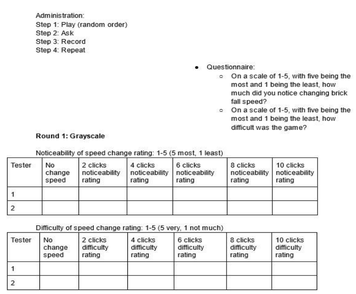
Photo of Mockup Unity Demo

- **Implementation of Runtime Adjustments**
  - The rate of change of the last two average reaction times are mathematically analyzed and brick queue speed is assigned a discrete value depending on analysis results.
  - The configuration file functionality revealed the mechanism by which game object properties (i.e queue speed) can be accessed and modified during runtime.

Config Page Showing Brick Properties

UX Testing & Design Breakdown

- **UX Test Design Process**
  - Testers played the updated Brick Drop version on a laptop with a mousepad. The questionnaire was delivered verbally after each round.



Administration:
Step 1: Play (random order)
Step 2: Ask
Step 3: Record
Step 4: Repeat

- Questionnaire:
  - On a scale of 1-5, with five being the most and 1 being the least, how much did you notice changing brick fall speed?
  - On a scale of 1-5, with five being the most and 1 being the least, how difficult was the game?

Round 1: Grayscale

Noticeability of speed change rating: 1-5 (5 most, 1 least)

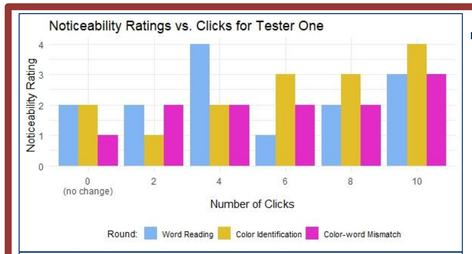
Tester	No change speed	2 clicks noticeability rating	4 clicks noticeability rating	6 clicks noticeability rating	8 clicks noticeability rating	10 clicks noticeability rating
1						
2						

Difficulty of speed change rating: 1-5 (5 very, 1 not much)

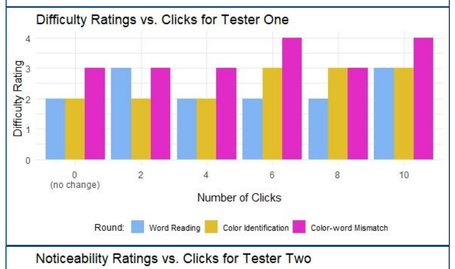
Tester	No change speed	2 clicks difficulty rating	4 clicks difficulty rating	6 clicks difficulty rating	8 clicks difficulty rating	10 clicks difficulty rating
1						
2						

UX Questionnaire

Results

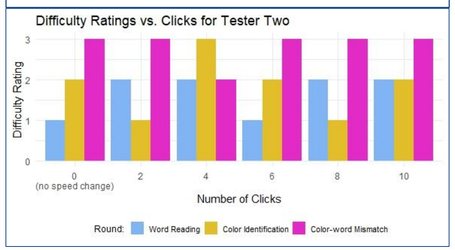


- Tester One rated the highest noticeability during Word Reading utilizing 4 clicks and Color Identification utilizing 10 clicks.



- Tester One rated the highest difficulty during Color-word Mismatch utilizing 6 and 10 clicks.

- Tester Two rated the highest noticeability during Color-word Mismatch utilizing 6 clicks and Word Reading utilizing 8 clicks.



- Tester Two rated the highest difficulty during Color-word Mismatch utilizing any click number and Color Identification utilizing 4 clicks.

Future Steps

- Conduct additional UX tests via mobile phone with at least 6-10 more testers to reduce the impact of outliers and confounding variables, such as mouse pad movement difficulties, on reaction time.
- Evaluate efficacy of the updated Brick Drop version by collecting UX and performance data from mCAPP study participants.

References & Acknowledgments