Evaluating Search and Selection Times for Linear, Grid, and Radial GUI Menus

Peter N. Caruana
Dept. of Electrical Engineering and Computer Science
York University
Toronto, Ontario, Canada M3J 1P3
caruana9@eecs.yorku.ca

ABSTRACT

It is well established that the layout of a menu affects selection time. This study examines these effects when button labels are randomly shuffled between trials. Twelve participants conducted a series of target selection trials for three types of menu layout: linear, grid and radial. It was found that grid menus significantly outperformed radial menus, with a difference of 0.51 seconds (22%) on average per task. Linear menus also performed better than radial menus in this regard, however no significance could be established. Comparing with participant preferences it appears likely that a meaningful difference could be found with a larger study. Data showed that a linear relationship exists between the average selection time and the mean squared radial displacement of a menus button positions. While this relationship fits well to the results of this experiment, it would be beneficial to confirm these results with a larger scale study.

Keywords

User interfaces, user study, menus, videogames

INTRODUCTION

One of the most fundamental and well-studied elements of graphical user interfaces (GUI) is the menu. Menus provide a simple way to structure information and controls on a display. When a player starts up a videogame, in most cases the first thing they are greeted with is a main menu or start menu. This screen serves as a hub for accessing all areas that the game has to offer. Typically comprised of various submenus for single player, multiplayer, settings etc., the main menu is what stands between gamers and what they want to do. Having a well-designed user interface (UI) is an important factor in cultivating overall user experience (UX) [4]. Ideally, designers of game main menus want to minimize time spent fiddling with options and maximize time spent actually playing the game.

Main menus differentiate themselves from in-game menus by not being a part of the core game loop. This leads to a number of different design considerations. In-game menus tend to prioritize having as few elements as possible. The reason for this is that during gameplay time is an important factor. Optimal acquisition times according to Miller [5] occur when the number of items in a menu is between four to eight choices, i.e. there are relatively few items. In contrast, main menu designs may have more elements than is optimal because of the absence of an associated time pressure.

Related Work

Menus are often used in computer applications to interact with related sets of items or data. Traditionally, linear menus are a common method for organizing elements. They do not take up much space and are simple. However, research has shown that radial menus are far more effective in terms of target seek time and error rate [1][2]. For this reason, many modern in-game menu designs use this style of menu. The game *Counter Strike: Global Offensive*(1) (*CS:GO*) makes use of a radial menu for in-game weapon selection, shown in Figure 1.

Figure 1. In-game screenshot of CS:GO weapon selection wheel

A large body of research regarding game menus focuses on studying effects of in-game menu layouts. In particular, a common area of study is the effectiveness of various menus using input devices specific to gaming. Chertoff et al. [2] evaluated the efficiency and user preference of three menu techniques when using a Nintendo Wii controller as input. They found that between linear, radial, and rotary menu techniques; radial menus had the best performance, least errors, and were most preferred by participants.

Grid menus can commonly be found on ecommerce platforms such as *Wish*. This style of menu tends to perform well for navigating large spaces of data [3] which is advantageous when elements are large, such as when using images of products. It can be difficult to find studies on the use of grid menus in video games, however this menu style has seen growing popularity in titles such as *CS:GO* or *Call of Duty*: *Black Ops 4*(2).

(1) Counter Strike: Global Offensive [Video game] (2012), Valve Software

(2) Call of Duty: Black Ops 4 [Video game] (2018), Activision

Recent trends in mainstream gaming have seen the design of main menu UI transition away from the use of linear lists in favor of grid and radial styles of menu. An excellent example of this can be seen in Figure 2, which shows how menu design has changed within the *Halo*(3) series. Comparing two installments eight years apart, the armor customization menu went from a linear text-based list (2a) to an image-based grid (2b). Radial styles are often employed in in-game menu design to improve interaction efficiency for time sensitive tasks because they are fast and facilitate the development of muscle memory over time [2]. In contrast, muscle memory is not much of a factor for main menus, where players only have a few interactions per game session (at the start and end of each session). To a larger degree main menu use will rely on searching for the desired item. With radial and grid styles of menu beginning to see more use in the design of main menu UI, this study seeks to evaluate the performance of these menu styles in this context.

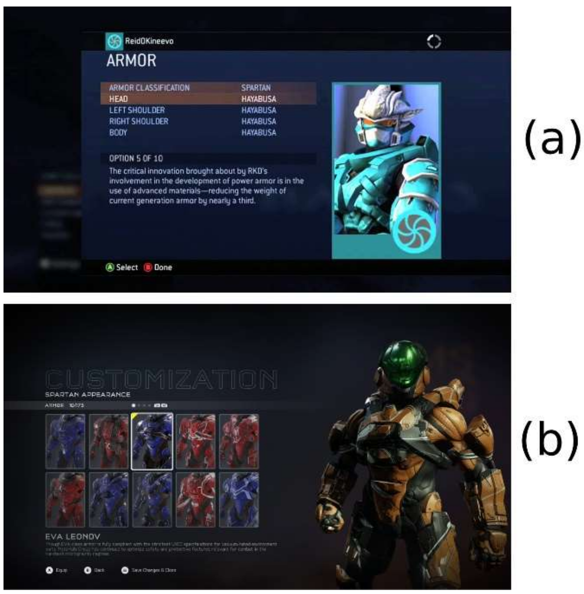

Figure 2. Armor customization menu in (a) Halo 3 (2007) and (b) Halo 5: Guardians (2015)

METHOD

This section describes an experiment to measure the selection speed and error rate of GUI menu styles using a computer mouse as the input method. A linear menu style will be tested to be used as a baseline, as well as in addition grid and radial menu styles. These menus were chosen as good representatives of common types of menu typically found in various applications.

Participants

Twelve participants were recruited for the study. Information on participant age and gender was collected. Participants were asked to indicate the typical number of hours per week they spend using a computer and whether they will be using a traditional mouse or using a trackpad to perform the experiment. Menu preference was recorded for each participant after completing the experiment.

Hardware

Due to the coronavirus pandemic, the tests were conducted in the participants' homes. This meant the experimental equipment will vary with each participant. It was assumed that all participants used 1080p resolution displays and a mouse or trackpad for input.

Software

The experimental software used by participants was a small GUI application written in Python. It was built primarily using the *pygame* and *pygame-gui* libraries. The application generates an 800×600 pixel window which runs a single experiment for a participant. The software divided a given experiment into three blocks (one for each menu style). A random target name was selected from a list of targets and gets displayed to the user at the start of each trial. Figure 3 shows the start screen, consisting of an instruction and a start button, located at the center of the window.

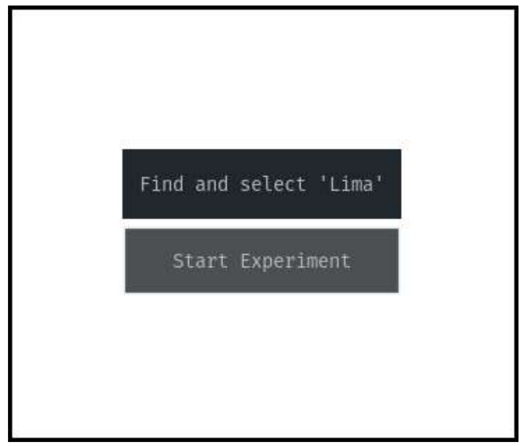

Figure 3. Start screen for a given trial.

Starting the experiment presents the user with a menu layout. The task was to select the option from the menu that has the same label as the target. Selecting an option was accomplished by clicking the button with the mouse. The user completes the trial when correct button was clicked. For each trial, the participant's start and end times (i.e. when they click to begin and when they select the correct button) were logged, as well as the number of miss-clicks. Menu style was determined by the current experimental block. Each menu style has a set of 12 fixed button locations to simulate a layout with many elements, akin to videogame main menus. For every trial, labels were randomly assigned to each button from the fixed set of labels, according to a uniform distribution. Layout randomization across trials was done to negate the effects of memorization and emphasizing the need to search for the target. The set of labels was the first 12 words of the NATO phonetic alphabet: Alpha, Bravo, Charlie, Delta, Echo, Foxtrot, Golf, Hotel, India, Juliet, Kilo, Lima. These were chosen as they are designed to be easily discerned from one another and are of similar word length.

Each menu was differentiated by two factors: button dimensions and button placement as described by the

(3) Halo [Videogame series], Xbox Game Studios

position of the button's geometric center. Horizontal values are from the left side of the window, vertical values are from the top of the window (i.e. (0, 0)) is top left of the screen. Each of the three menus implemented for this experiment can be seen in Figure 4: linear (4a), grid (4b) and radial (4c).

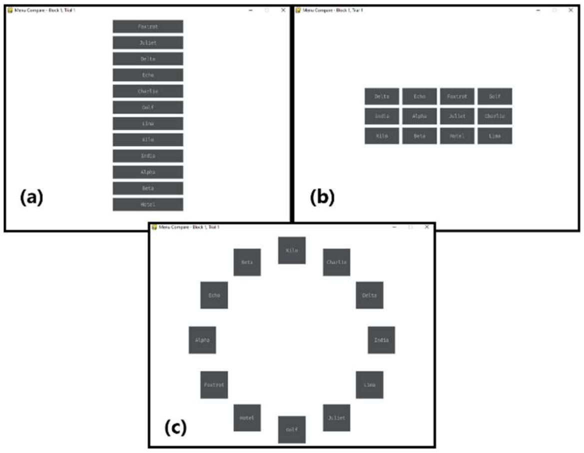

Figure 4. Menu styles: (a) Linear, (b) Grid, (c) Radial

Mean Squared Radial Displacement (MSRD)

A useful metric for summarizing a menu is the mean squared radial displacement. This is defined as the average squared distance of all buttons from the center of mass of the layout. For $P(i)$ being the (x, y) position of the i^{th} button in the menu, and N being the total number of buttons, the center of mass is calculated by:

$$\bar{\mu} = \frac{1}{N}\sum_{i=0}^{N-1} P(i) \rightarrow \bar{\mu} = (\bar{x}, \bar{y}) \qquad (1)$$

This is the average position of all buttons i.e. center of mass. The MSRD, using Eq. (1) is then given by:

$$\mu_d^2 = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{|P(i) - \bar{\mu}|}{\lambda}\right)^2 \qquad (2)$$

λ is a normalizing factor, to account for different units of measurement. It should be set to the largest distance possible within the scope of the experiment. The λ used in this study was 600 pixels. μ_d^2 is a measure of how clustered a set of points are together. Table 1 below shows μ_d^2 values for the linear, grid and radial menu types. The position functions for each menu are defined in the next three sections.

Table 1. Normalized mean squared radial displacements (μ_d^2) for three menu types

Menu Type	μ_d^2
Linear	0.067
Grid	0.044
Radial	0.174

Linear Menu

The linear layout uses buttons of size 200 × 40 pixels. The dimensions of the buttons were chosen to be five times wider than tall so that the primary factor in selection was vertical positioning of the cursor. Buttons were centered horizontally on screen at 300 pixels. Vertical button placement was done such that there is a 5-pixel division between all buttons. For buttons with a 40-pixel height, this meant each was placed 45 pixels from the location of the button above it. The positions can be described by the following expression:

$$P_{\text{lin}}(i) = (300, 10 + i \cdot 45) \text{ for } i = [0,11]$$

A factor of 10 was included in the formula to account for a 10-pixel buffer from the top of the window.

Grid Menu

To emphasize 2D movement, buttons for this layout were set to the dimensions of 100 × 50 pixels making them more square compared to the linear layout. The 12 buttons were placed in a 4x3 grid, and aligned such that the space between buttons was 5 pixels. The expression for the position is:

$$P_{\text{grid}}(i,j) = \big(200 + i \cdot (105), 200 + j \cdot (55)\big)$$
$$\text{for } i = [0,3], j = [0,2]$$

Radial Menu

The radial style of layout was defined by having all of the elements placed in a circle. For this experiment, buttons were placed evenly around a circle with a radius of 250 pixels. The dimensions of each button were made 80 × 80 pixels to be completely square. This was done so that the radial distance to each button was approximately the same from the center of the window. Positions were generated according to the expression,

$$P_{\text{rad}}(i) = (250 \sin\frac{i\pi}{6} + 400, 250 \cos\frac{i\pi}{6} + 300)$$
$$\text{for } i = [0,11]$$

An offset of (400,300) was added to center the circle in the window. By making the angular argument in multiples of $\pi/6$, the twelve positions generated by $P_{\text{rad}}(i)$ are spaced evenly.

Procedure

Participants were sent an executable version of the experimental software and were assigned to either group 1, 2 or 3. Upon running the application, they were prompted to complete a pre-test questionnaire, requiring them to enter information which was specified in *Participants,* as well as their group number.

Participants were instructed on the use of the software and given an opportunity to ask questions. The experiment consisted of three blocks, one for each menu style. Each participant completed 20 task trials for each block, totaling to 60 trials per person. Block order was based on the participant's group number. Breaks were allowed at any time between trials.

Figure 5 illustrates how a participant would run the experiment. For each trial, the participant would read the label of the target button and click the "Start Experiment" button to begin. They then located and clicked the target button, ending the trial. Participants were told not to rush, in order to emphasize the non-time sensitive aspect of the tasks. After completing the experiment, participants were asked to send back the data files generated by the software.

Figure 5. A participant performing the experiment

Design

This user study followed a 3 × 20 within-subjects design. Independent variables were:

- Menu style (linear, grid, radial)
- Trials $(1, 2, \ldots 20)$

Dependent variables were search and selection time (SST), and error rate. SST represents the total time to complete a trial, comprised of the time to locate the target (search) and a time to move the cursor to it (selection). Error rate represents the number of clicks which miss the target.

There were three groups of four participants. To counterbalance the order of testing, the block order per group was based on a 3 × 3 Latin Square. The total number of trials was 12 participants × 3 menu types × 20 trials = 720 trials.

RESULTS AND DISCUSSION

Post-study analysis yielded interesting findings on how the different menu layouts affected the average SST per trial. Qualitative analysis also showed interesting results regarding user preference. The preference of menu type among users was misaligned with the fastest menu type, but overall supported the experimental data.

Search and Selection Time

The primary dependent variable of this study was search and selection time, i.e., the time it takes for a user to complete one trial. Over a total of 720 trials, the mean time to complete a single trial was 2.00 seconds. The mean times for each menu type were: 1.96 seconds for linear; 1.77 seconds for grid; 2.27 seconds for radial. A plot of mean times can be seen in Figure 6. Performing a two-way ANOVA, it was found that the effect of menu type on SST was statistically significant ($F_{2,18}$ = 12.62, $p < .0005$). Post-hoc analysis (Fisher-LSD) determined the difference between grid and radial menu types to be significant. On average there was a difference of 0.51 seconds (22%) per trial between these two menu types.

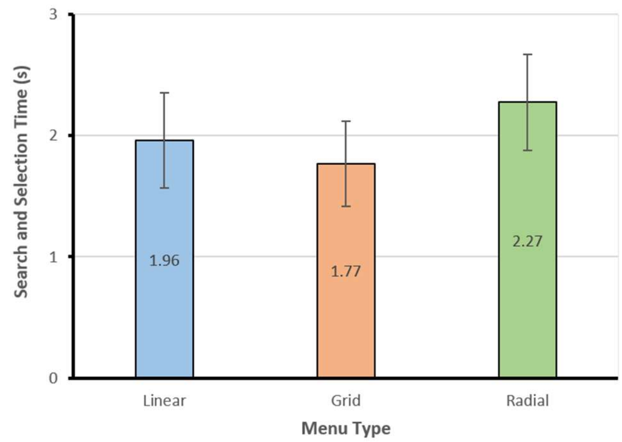

Figure 6. Mean SST (s) by menu type. Error bars show $\pm$ 1 SD. From left to right SDs are [0.39, 0.35, 0.39]

This manifested in quite substantial differences in the total time to complete each block. The mean completion time for the grid menu was 35.86 seconds versus 42.41 seconds for radial.

There was a difference of 0.31 seconds (14%) between the mean time the of the linear and radial menus. This difference was not statistically significant. The same was true for linear and radial menus, which had a non-significant difference of 0.19 seconds (10%). Given a larger sample size to reduce variability, significant differences could still be found between linear and radial layouts. This idea is further supported when considering which menu types users liked and disliked the most, which will be discussed in greater detail in *User Preferences*. The standard deviation for each mean was quite large, at about 20% in all cases. This leaves plenty of room for improvement in future studies.

The randomization of target locations was successful in its intention to eliminate the effects of learning. The effect of trial number on SST was not statistically significant ($F_{19,171}$ = 1.435, $p > .05$), which indicates that participants did not meaningfully improve as they completed more trials. This strengthens the claim that the primary factor measured in this experiment was visual search time. One can see from Figure 7 that there was no detectable effect of learning on radial and linear times. There is a slight negative trend in the plot for the grid menu type, however as stated above this was not significant. Generally, it can be seen that participants did the worst on the first trial of a given block. This was expected as they got adjusted to what they were doing and understand how exactly the given menu was laid out.

One surprising discovery was the apparent relationship between mean squared radial displacement (MSRD) and SST. Figure 8 shows that plotting the normalized MSRD for each menu against their mean SST yields a linear relationship. The fit linear function is highly accurate for the data ($R > .95$).

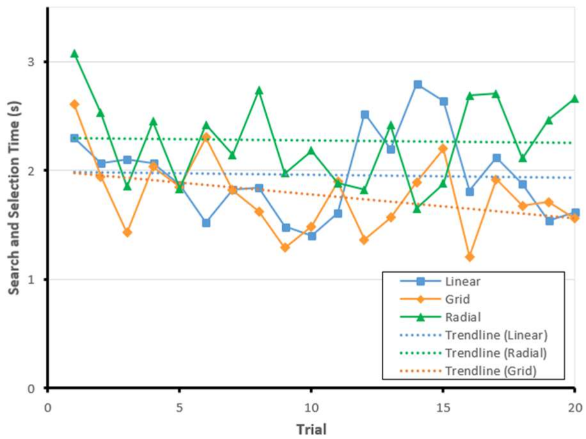

Figure 7. SST as a function of trial number for each menu type with trendlines.

An intuitive explanation for this is that the human eye can only focus on a small area at once. More compact positioning of elements allows people to compare items more efficiently. This finding invites future study in testing selection times with respect to variations in mean displacement. "How sensitive is this linear relationship to the inclusion of outliers?" is an interesting question to explore but will require more testing outside of the scope of this paper. With only three data points it is difficult to assign high confidence on the accuracy of this model.

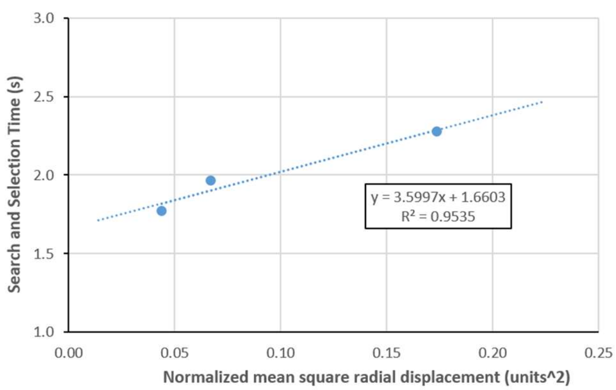

Figure 8. Plotting mean squared radial displacement vs SST with linear trendline

Participant Preferences

It was interesting comparing the objective test results with how participants self-reported their most and least preferred methods. Looking at Figure 9, one can see that the most disliked method was radial, as reported by 8 participants. When looking at linear and grid however, one can see they were nearly tied, with 6 participants preferring linear, and 5 participants preferring grid. Overall, linear was the second-most disliked menu type.

Every participant who preferred the linear menu also disliked radial the most. When looking at the least preferred menu by those who preferred the grid menu, it was almost evenly split, with 3 disliking linear the most, and 2 disliking radial the most.

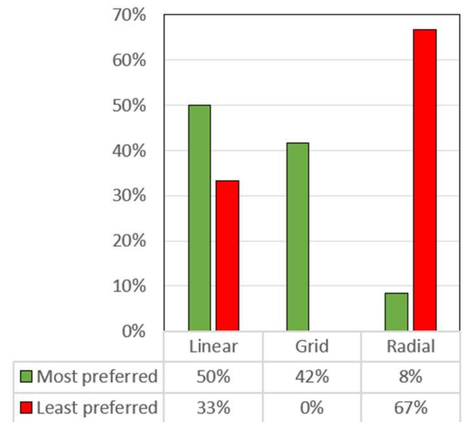

Figure 9. Participant menu preferences (N=12)

Even more bizarre was that all the participants who listed linear as their most disliked menu type had linear as their fastest or second fastest menu. Indicated by the results in the previous section, 10 of the 12 participants performed the fastest on the grid menu type, in stark contrast of their preferences.

Participants were also asked to comment on their choices of menu preference. Those who preferred linear stated that it was simple to do a vertical scan to find their target. The consensus on the radial menu was that the layout required treating it as a sort of transformed linear list, where they sequentially searched in a clockwise or counterclockwise fashion. If strategy was the defining factor in task times it would be expected that radial menus might perform more closely to linear menus. However, this was not the case. This information helps lend credibility to the idea that the MSRD was likely the defining factor between menu types, rather than spatial orientation.

These preference results appear to suggest that with more testing, a meaningful difference may be found between the performance of linear and radial menu types. They also help affirm what was suggested in the *Search and Selection Time* section, namely that more testing may provide more accurate data.

Participant Error

The effect of menu type on error was not statistically significant ($F_{2,18}$= 1.090, $p > .05$). Most participants (58%) had no errors at all while testing. Of the 5 (42%) who did have errors, the largest mean error by an individual was 0.25 errors per trial. The most errors in a single trial was 11 errors, during trial 15 when testing the grid menu type. This can be considered an outlier, because the participant likely had forgotten the target label and arbitrarily clicked buttons until they found the correct one. The relative non-existence of errors can likely be attributed to the lack of time pressure. Participants were able to take their time with each task.

CONCLUSION

The results of this experiment demonstrate the context-dependence of the performance of various menu types for selection tasks. Results showed that grid menus outperformed radial menus by a difference of 0.51 seconds (22%) per task when the location of elements was randomized. This contrasts with the results found by Chertoff et al. [2], where they reported that radial menus were the fastest and least error prone; in a context of static label locations. When memorization is removed (and thus visual search emphasized), the spatial localization of elements becomes the more dominant factor in predicting overall task time.

It was shown that there was a linear relationship between the mean squared radial displacement (MSRD) of menu elements and the search and selection time (SST) per task. More testing is required to validate this model, but it does provide a simple way to estimate the performance of a menu when no guarantees on target locations can be made. Applications where users are unlikely or unable to memorize the layout of buttons such as videogame main menus or e-commerce platforms may be able to benefit by taking this information into account.

ACKNOWLEDGMENT

Many thanks go out to Prof. MacKenzie as well as peers in EECS 6329 who all gave excellent advice and guidance.

REFERENCES

1. J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. 1988. An empirical comparison of pie vs. linear menus. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '88).* New York, ACM, 95–100.

2. D. B. Chertoff, Ross W. Byers, and Joseph J. LaViola. 2009. An exploration of menu techniques using a 3D game input device. In *Proceedings of the 4th International Conference on Foundations of Digital Games (FDG '09).* New York, ACM, 256–262

3. F. Knip et al. 2014. A field study on the usability of a nearby search app for finding and exploring places and events. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia (MUM '14).* New York, ACM, 123–132.

4. D. P. Kristiadi et al., The effect of UI, UX and GX on video games," *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*, 58-163

5. DP. Miller. The Depth/Breadth Tradeoff in Hierarchical Computer Menus. *Proceedings of the Human Factors Society Annual Meeting*, 296-300.