## הקדמה

מטרות העבודה:

* היכרות עם שיטות תרשימים מתקדמות.
* למידת תכנות של משימות חזותיות קוגנטיביות.
* איסוף זמני תגובה באמצעות מטלאב.
* תרגול תכנות פונקציונאלי במטלאב.
* שימוש במבני נתונים שונים של מטלאב.

חיפוש חזותי הינה פעולה שאנו מבצעים כמעט בכל רגע, בין אם חיפוש קרטון החלב במקרר ובין אם זיהוי פושע במסדר זיהוי. מכיון שזוהי פעולה כל כך שכיחה היא גם נחקרת מאוד בתחום המדעים הקוגנטיבים. במהלך המחקר התגלו תופעות שונות ופותחו תאוריות שונות המתארות איך נעשה חיפוש חזותי.אחד האפקטים שהתגלה הוא אפקט הPop out אשר קורה כאשר ישנו אובייקט השונה משאר האובייקטים שמסביבו בתכונה אחת בלבד כמו צורה או צבע. האובייקט "יקפוץ" לנו לעניים בצורה מיידית, כלומר נשים אליו לב מהר מאוד ללא תלות בכמות האובייקטים הסובבים אותו. דוגמה לאפקט זה קוראת במהלך משחק כדורגל, שחקן יכול לזהות במהירות את חבריו לקבוצה בתוך כל שחקני הקבוצה היריבה המקיפים אותו. עוד התגלה כי כאשר אנו מחפשים אובייקט אשר על מנת למצוא אותו אנו צריכים לשים לב ליותר מתכונה אחת שלו, לדוגמה גם צורה וגם צבע, זמן החיפוש שלו יקח יותר זמן והוא נמצא במתאם חזק עם כמות הפריטים הסובבים אותו. דוגמה לכך היא חיפוש עט כחול בקלמר מלא בכלי כתיבה, אנו צריכים למצוא אובייקט בצורת עט בצבע כחול, ככול שיהיו יותר פריטים בקלמר הזמן שידרש למציאת העט יקח יותר זמן.

בעבודה זו אנו נשחזר ניסוי בסיסי מאוד בחקר החיפוש החזותי אשר בא לבדוק את ההבדלים בזמני התגובה כאשר מחפשים אובייקט הנבדל בתכונה אחת מהמסיחים הסובבים אותו - Feature לעומת חיפוש אובייקט הנבדל בשתי תכונות מהמסיחים - Conjunction. מלבד סוגי החיפוש בניסוי יבדקו גם שני תרחישים, תרחיש בו קיים אובייקט אשר מהווה מטרת חיפוש ותרחיש בו לא קיימת מטרת חיפוש. נצפה לראות כי תוצאות הניסוי ישקפו את ממצאי המחקר עד היום והם, זמני החיפוש בחיפוש מסוג Feature יהיו קצרים מאוד ולא תלויים בכמות המסיחים ואילו זמני החיפוש בחיפוש ממסוג Conjunction יהיו ארוכים יותר ויגדלו ככול שכמות המסיחים תגדל. נצפה לראות תוצאות אילו גם כאשר ישנו אובייקט מטרה וגם כאשר אובייקט המטרה נעדר.

על מנת לבדוק זאת נציג לנבדק שורה של תרשימים כאשר התרשימים השונים יהוו הדמיות של סוגי החיפוש השונים. בחיפוש מסוגFeature יופיעו המסיחים ומטרת החיפוש בצבע זהה אך המטרה תיבדל מהמסיחים בצורה. ואילו בחיפוש מסוג Conjunction יופיעו שתי צורות של מסיחים כאשר כל צורה בצבע שונה וצורת המטרה תיבדל מכל המסיחים בצבע ובצורה. בחלק מהתרשמים לא תופיע המטרה ובחלק מהם היא תופיע ועל הנבדק להחליט האם אובייקט המטרה מופיע בתרשים ולהגיב בהתאם. את זמן החיפוש נמדוד האמצעות זמן התגובה של הנבדק. בהתאם להשערה, נצפה שכמות גדולה של מסיחים תגרום לזמן תגובה ארוך יותר בהדמיות מסוג Conjunction ואילו על זמן התגובה בהדמיות מסוג Features כמות המסיחים לא תשפיע והוא ישאר נמוך.

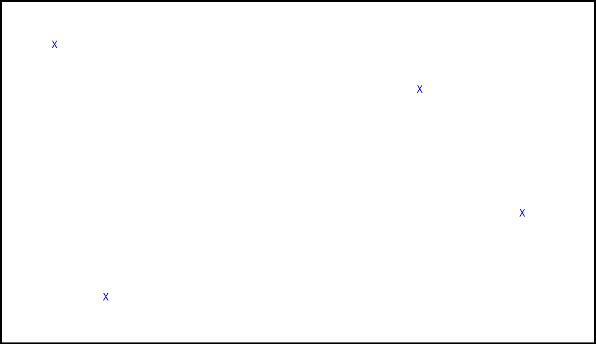
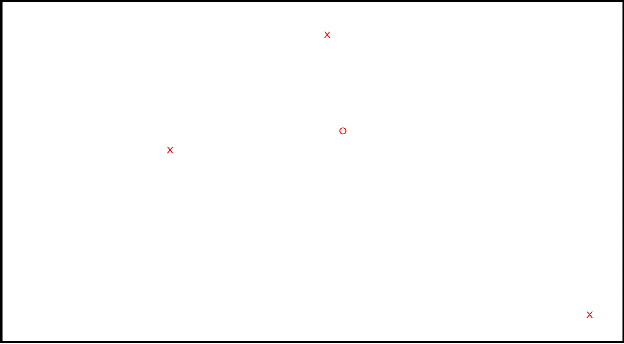
שיטות:

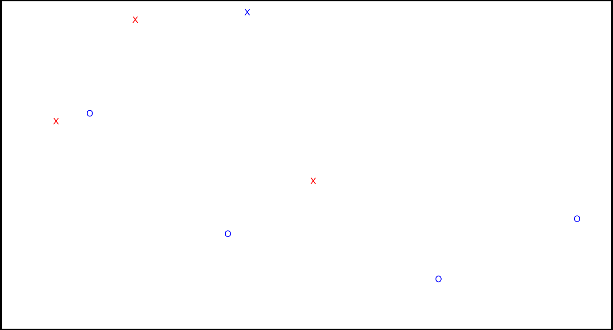
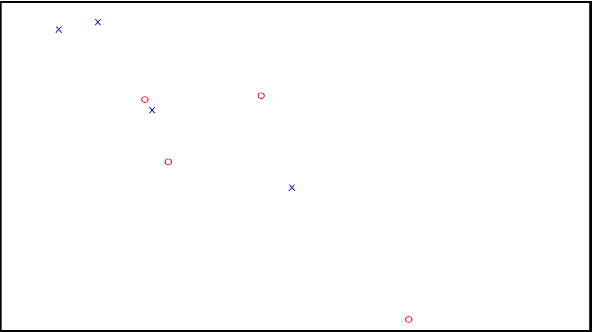
כלים:

על מנת לבדוק את זמני התגובה בהתאם לתנאי החיפוש – Feature או Conjunction, התרחיש – האם יש אובייקט מטרה או שלא ובהתאם לכמות האובייקטים המוצגים – גודל הסט (מטרה + מסיחים) 4,8,12,16. הניסוי מורכב מ 8 חלקים – בלוקים, וכל בלוק הוא בעל הפרמטרים הבאים: סוג החיפוש וכמות האובייקטים ובכל בלוק חצי מהתרשימים מכילים מטרה וחצי לא. בזמן ריצת כל בלוק יופיעו על המסך צורות 'X' ו 'O' בצבעים כחול ואדום. הצבעים וכמות הצורות תואמת את הפרמטרים של הבלוק – סוג החיפוש וגודל הסט.

בחיפוש מסוג Feature כאשר יש מטרה, תוצג אחת הצורות כמטרה פעם אחת בלבד ואילו כל שאר המסיחים יהיו מן הצורה השנייה כאשר גם המטרה וגם המסיחים יופיעו באותו צבע. כאשר אין מטרה יופיעו אך ורק מסיחים כולם מאותה צורה.

בחיפוש מסוג Conjunction בתרחיש בו יש מטרה חצי מהגירויים על המסך יהיו מהצורה 'X' באחד הצבעים וחצי 'O' בצבע השני מלבד צורת המטרה אשר תופיע בצבע של הצורה שאינה מטרה. (ראו תרשים). כאשר אין מטרה יופיעו בדיוק חצי מהגירויים כ 'X' בצבע אחד ובדיוק חצי כ 'O' בצבע השני.

Feature – no target – set size = 4 feature – target shape: O - set size = 4

set size = 8 - Conjunction – no target - set size = 8 shape: X Conjunction – target

על הנבדק להחליט האם הוא רואה את המטרה או לא ולהגיב בהתאם. זמני התגובה נמדדים מהרגע שמופיע הגרף ועד הרגע שבו לחץ על המקש המתאים.

כל נתוני הניסוי נשמרים במבנה (structure) Expirament, מבנה זה מכיל 8 מבנים נוספים אשר מייצגים כל בלוק בניסוי. בכל אחד מ8 המבנים הללו נשמרים כל הפרמטרים של אותו בלוק וכל תוצאות הבלוק – זמן התגובה, נכונות התגובה עבור כל תרשים שהוצג והאם הוצגה מטרה או לא. בנוסף ישנו שדה בכל בלוק השומר את הממוצע וסטיית התקן של זמני התגובה לפי תרחיש המטרה. המבנה Expirament מכיל עוד מבנה נוסף – All\_results אשר בו נשמרים כל הנתונים הרלוונטים עבור כל הבלוקים ביחד כמו המתאם וה p-valueשבהם נשתמש בניתוח הנתונים.

## 

this function creates the structure where all data is saved  
%in this structure there is a structure for every block including  
%parameters, data and results. there is another structure holds  
%all results from all blocks  
function S = build\_struct(size,cond,scenario,set\_sizes,trial\_num)  
 mean\_sd = ["mean","SD"];  
 num\_of\_sizes = length(set\_sizes);  
 trailZzer = zeros(trial\_num,1);  
 bool = true(trial\_num,1);  
  
 generic\_res\_table1 = build\_double\_table(mean\_sd,scenario);  
 generic\_res\_table2 = build\_cell\_table(cond,scenario,num\_of\_sizes);  
 generic\_res\_table3 = build\_double\_table(cond,scenario);  
 passed\_trails\_table = build\_double\_table(cond,"size\_"+string(set\_sizes));  
  
 GenericBlock = struct('condition',"",'set\_size',0,'has\_target',bool,...  
 'acc',bool,'rt',trailZzer,'passed',bool,'block\_results',generic\_res\_table1);  
  
 %creating structure for eache block  
 cur\_cond = 0;  
 for i = 1:size  
 if mod(i - 1,num\_of\_sizes) == 0 %every num\_of\_size blocks switch cond  
 cur\_cond = cur\_cond + 1;  
 end  
 cur\_block = char("B" + i);  
 S.(cur\_block) = GenericBlock;  
 S.(cur\_block).condition = cond(cur\_cond);  
 S.(cur\_block).set\_size = set\_sizes(mod(i-1,num\_of\_sizes)+1);  
 end  
  
 %creating all results structure  
 S.All\_results = struct('passed\_trails',passed\_trails\_table,...  
 'mean',generic\_res\_table2,'sd',generic\_res\_table2,...  
 'p\_val',generic\_res\_table3,'rho',generic\_res\_table3,'fit',generic\_res\_table2);  
end

מהלך הניסוי:

ראשית מוצגות ההוראות לנבדק, על הנבדק להחליט בכל צעד של הניסוי האם הוא רואה את המטרה או לא. במידה והוא רואה את המטרה עליו ללחוץ על המקש 'a' ובמידה ואינו רואה עליו ללחוץ על המקש 'l'. בנוסף הוא מתבקש לנסות לעשות זאת בצורה המהירה ביותר אך מבלי לפגוע בדיוק התשובה.

התרשימים יופיעו בזה אחר זה על פי הבלוק שמופעל באותו הרגע. סדר הבלוקים נקבע בצורה אקראית וכך גם גירוי המטרה והוא קבוע לאורך כל הבלוק. סדר התרחישים נקבע בצורה אקראית כאשר בדיוק חצי מהתרשימים יציגו מטרה וחצי מהם לא, כמו כן צבע הגירויים ומיקומם על המסך נקבעים בצורה אקראית ומוגרלים מחדש בין תרשים לתרשים. בתחילת כל בלוק מצויין בפני הנבדק המטרה אשר עליו לחפש 'X' או 'O' לאורך כל הבלוק.

block\_order = randperm(num\_of\_blocks); %determining bolck order randomizingly  
  
for i = block\_order %running through blocks by block\_order  
 cur\_block\_name = (char("B"+i));  
 Expirament.(cur\_block\_name) = run\_block(Expirament.(cur\_block\_name),...  
 stimuli\_shape,color\_vec,num\_of\_trails,fontsize);  
end  
% save('raw data' , 'Expirament');

בחירת סדר אקראי של הבלוקים והרצתם על ידי הפונקציה run\_block

%this function run all trails of the specified block,  
%it determines the target shape and the figure parameters(locations,colors),  
%builds the figures and collects the data  
function Block = run\_block(Block,shapes\_vec,color\_vec,num\_of\_trails,fontsize)  
 [target\_shape,none\_target\_shape] = rand\_two\_elements(shapes\_vec); %choose target randomly  
 scenario\_vec = rand\_elements\_eq(num\_of\_trails,2); %randomize scenario order  
 Block.has\_target = scenario\_vec';  
  
 for i = 1:num\_of\_trails %run all block trails  
 [x\_vec,y\_vec,target\_color,none\_target\_color] =...  
 fig\_parameters(Block,color\_vec); %determining locations and target color  
  
 build\_fig(Block,scenario\_vec(i),target\_shape,none\_target\_shape,...  
 target\_color,none\_target\_color,i, x\_vec,y\_vec,fontsize); %show trail figure  
  
 [acc,rt] = trail\_data(scenario\_vec(i)); %collecting data from subject  
 Block.acc(i) = acc;  
 Block.rt(i) = rt;  
 end  
end

קביעת הפרמטרים האקראיים של הבלוק והצגת התרשימים.

ניתוח נתונים:

לאחר שלב איסוף הנתונים ולפני שלב ניתוח הנתונים עצמו נבצע שלב סינון הנתונים: כל תגובה שאינה נכונה או שחורגת מגבולות זמני התגובה שנבחרו לפני הניסוי (בין 0.1 ל- 3 שניות) יגדרו כלא תקינים ואינן ישומשו בניתוח הנתונים. ערכים שאינם תקינים יסומנו בערך false כדי שנדע לא להתייחס אליה בשלב ניתוח הנתונים. נדרוש מספר מינימאלי של תגובות תקינות עבור כל בלוק (20) ובנוסף מספר מינמאלי של תגובות תקינות עבור כל הניסוי (160). אם נמצא כי אחד התנאים לא מתקיים נפסיק את הניסוי לא נמשיך לשלב האנליזה ולנבדק תוצג הודעת שגיאה. כמות התשובות התקינות עבור כל תנאי וכל גודל סט מוכנסת לטבלה Expirament.All\_results.passed\_trails.

%checking for wrong/bad results, in case not enough good data will not  
%move on to analysis  
count\_valid = 0;  
for i = 1:num\_of\_blocks  
 cur\_block\_name = (char("B"+i));  
 char\_size = char("size\_"+ Expirament.(cur\_block\_name).set\_size);  
 char\_cond = char(Expirament.(cur\_block\_name).condition);  
  
 [Expirament.(cur\_block\_name),has\_passed,num\_of\_valid\_ans] = ...  
 is\_valid\_block(Expirament.(cur\_block\_name),min\_correct\_ans\_per\_block,max\_rt\_in\_sec,min\_rt\_in\_sec);  
 count\_valid = count\_valid + num\_of\_valid\_ans;  
 Expirament.All\_results.passed\_trails.(char\_size)(char\_cond) = num\_of\_valid\_ans;  
  
% if has\_passed == 0  
% error('not enough correct answer per block');  
% end  
end  
% if count\_valid < min\_correct\_ans  
% error('not enough correct answer for expirament');  
% end

בשלב האנליזה נבדקים רק הנתונים שאינם מסומנים ב false לאחר שלב הסינון. בתחילה מחושבים הממוצע של זמן התגובה בשניות וסטיית התקן עבור כל בלוק בנפרד. הממוצעים וסטיות התקן מסווגים על פי תנאי הניסוי, גודל הסט והתרחיש. נחשב את המתאם בין גודל הסט לכמות הגירויים וכמו כן נחשב גם את ה p – value (לפי כל תנאי ניסוי וכל תרחיש). בשלב הבא מותאמת לנתונים (ציר x -גודל הקבוצה , ציר y -ממוצע זמני התגובה) פונקציית שתייצג בצורה הטובה ביותר את ממוצע זמני התגובה כתלות בגודל הקבוצה וזאת עבור כל תנאי וכל תרחיש. ההנחה היא כי זמן התגובה תלוי לינארית בגודל הקבוצה (על פי מה שהתבקש בשיעור) ולכן נעשה שימוש בפולינום ממעלה ראשונה על ידי הפונקציה polyfit. הפלט שנוצר הוא השיפוע של הפולינום שהותאם. את השיפוע נכניס לפונקציה polyval על מנת לחשב את הערכים של הפונקציה הנלמדת כתלות בגודל הקבוצה.

%calculating mean & SD  
for i = 1:num\_of\_blocks  
 cur\_block\_name = (char("B"+i));  
 Cur\_block = Expirament.(cur\_block\_name);

% get valid trails by scenario  
 inter1 = get\_sub\_vec\_intersect(Cur\_block.passed, 1,Cur\_block.(scenario(1)), 1);  
 inter2 = get\_sub\_vec\_intersect(Cur\_block.passed, 1,Cur\_block.(scenario(1)), 0);  
  
 Expirament = calc\_mean\_sd\_per\_cond(Expirament,i,inter1,scenario(1),num\_of\_sizes);  
 Expirament = calc\_mean\_sd\_per\_cond(Expirament,i,inter2,scenario(2),num\_of\_sizes);  
  
end  
  
% calculating correlation and p-value  
[r1,p1] = calculat\_pval\_cor(set\_sizes,Expirament.All\_results.mean.(scenario(1)){cond(1)});  
[r2,p2] = calculat\_pval\_cor(set\_sizes,Expirament.All\_results.mean.(scenario(2)){cond(1)});  
[r3,p3] = calculat\_pval\_cor(set\_sizes,Expirament.All\_results.mean.(scenario(1)){cond(2)});  
[r4,p4] = calculat\_pval\_cor(set\_sizes,Expirament.All\_results.mean.(scenario(2)){cond(2)});  
  
Expirament.All\_results.rho(cond(1),scenario(1)) = {r1};  
Expirament.All\_results.rho(cond(1),scenario(2)) = {r2};  
Expirament.All\_results.rho(cond(2),scenario(1)) = {r3};  
Expirament.All\_results.rho(cond(2),scenario(2)) = {r4};  
  
Expirament.All\_results.p\_val(cond(1),scenario(1)) = {p1};  
Expirament.All\_results.p\_val(cond(1),scenario(2)) = {p2};  
Expirament.All\_results.p\_val(cond(2),scenario(1)) = {p3};  
Expirament.All\_results.p\_val(cond(2),scenario(2)) = {p4};  
  
% fitting function for each condition and target  
Expirament.All\_results.fit.(scenario(1)){cond(1)} = ...  
 linear\_fit(set\_sizes,Expirament.All\_results.mean.(scenario(1)){cond(1)});  
Expirament.All\_results.fit.(scenario(2)){cond(1)} = ...  
 linear\_fit(set\_sizes,Expirament.All\_results.mean.(scenario(2)){cond(1)});  
Expirament.All\_results.fit.(scenario(1)){cond(2)} = ...  
 linear\_fit(set\_sizes,Expirament.All\_results.mean.(scenario(1)){cond(2)});  
Expirament.All\_results.fit.(scenario(2)){cond(2)} = ...  
 linear\_fit(set\_sizes,Expirament.All\_results.mean.(scenario(2)){cond(2)});

