**<Mega Tublight :) >**

**מסמך עיצוב**

**<מספר גרסא : 1>**

**<13.01.16>**

**<מיכל & אליסף>**

**הסטוריית גרסאות המסמך**

|  |  |  |  |
| --- | --- | --- | --- |
| **תאריך** | **עורך/ת** | **תיאור / תקציר השינויים** | **הערות נוספות** |
| 24.11.15 | מיכל & אליסף | התחלה וזה | ממש ממש ממש לא סופי |
| 1.12.15 | מיכל & אליסף | המשך משהו | להלההלה |
| 04.12.15 | מיכל & אליסף | ממשיכים | לאליסף אין כוח |
| 07.12.15 | מיכל & אליסף | עדיין ממשיכים | נחשוב על זה בהמשך...O\_o |

1. **מבוא**

כאן יש לתאר את מטרת המסמך - לאיזה קהל יעד הוא מיועד.

בנוסף יש לצטט או לכתוב בעצמכם קישורים והסברים למושגים רלוונטים (כולל ראשי תיבות או מונחים מתקדמים שלא כולם בהכרח מכירים) שבהם נעשה שימוש בהמשך המסמך.

שימו לב - שני אתרים שימושיים (יש עוד כמוהם ברשת) שעשויים להיות שימושיים לסעיפים הבאים במסמך:

* <https://moqups.com> - עיצוב מהיר ונח של הדמיה ויזואלית לממשק המשתמש
* <https://cacoo.com> - יצירת דיאגרמות, כמו למשל workflow של פעולות משתמש / מצבים במערכת / סדר הודעות של פרוטוקול תקשורת (ברמה האפליקטיבית), או חלוקת תפקידים בין רכיבים וכדומה

מסמך זה מהווה דף הוראות לנו, מפתחי המוצר, ועל פיו נממש את הפרויקט.   
המסמך מחולק לשני חלקים : החלק החומרתי והחלק התוכנתי.

**חלק חומרתי :**

שניהם בעצם אותו רכיב

**16 כניסות\יציאות**

**CD74HC4067E**

**מושגים והגדרות בהם נשתמש בהמשך :**

* **Mux – קיצור של multiplexer (מרבב), תפקידו לבחור בין מספר קלטים לפלט יחיד.**
* **Dmux – קיצור של Demultiplexer (מפלג), תפקידו להוציא קלט יחיד למספר פלטים.**
* רצועות לדים **– רצועות שמכילות לדים (60 למטר). אנו נשתמש בצ'יפ WS2811.**
* **IR emitter – נורת לד המשדרת אור אינפרה אדום.**
* **IR receiver – קולט אור אינפרה אדום.**
* **ADC - ממיר קלט אנלוגי לפלט דיגיטלי בפרוטוקול . אנו נשתמש ב** 3004**MCP בפרוטוקול SPI.**
* **Raspberry pi2 – מחשב קטן (מוח הפרויקט).**
* **PCB – לוחות להלחמה. גודל :** 12ˣ18
* **Arduino – מיקרו-בקר שניתן לתכנות. מתאים לעבודה עם חומרה.**
* פינים **– לחיבור בין דברים.**

**חלק תוכנתי :**

**מושגים והגדרות בהם נשתמש בהמשך :**

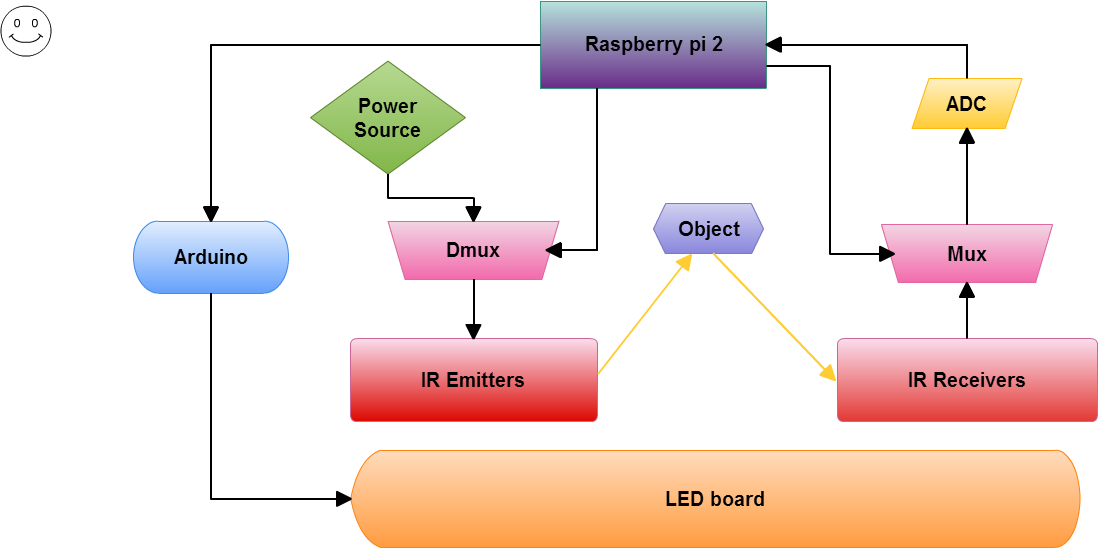
* Double Buffering :

שיטה שבה נעביר את הפלט אל המסך (מסך הלדים).  
לפי שיטה זו נשתמש בשתי מטריצות (תמונות) גלובליות, באחת יהיה הפלט העכשווי (t – 1), ובשנייה יהיה הפלט הבא, אותו נמלא עכשיו (t).

* Mutex :

1. ארכיטקטורת המערכת :
   1. "מבט על" :

**חלק חומרתי :**



* **Raspberry pi 2 –** הוא "מוח" הפרויקט.

**תפקידיו:** א. לשלוח מידע למסך הלדים להצגה (דרך הארדואינו).  
 ב. לשלוח לדימוקס איזה לד איפרה אדום להדליק.  
 ג. לשלוח למוקס מאיזה קולט לקבל מידע, ולקבל אותו דרך הADC.

* **Arduino –** תפקידו לקבל תמונה מהרספברי ולהתאים את השליחה למסך הלדים.
* **Led board –** תפקידו להציג את שקיבל מהארדואינו.
* **Dmux –** מקבל מהרספברי איזה לד להדליק בכל רגע נתון.
* **IR Emitters –** פולטים אינפרה אדום שיפגע באובייקט וממנו יחזור אל הקולט.
* **Object –** עצם מוחשי כלשהו במרחק קליטה מהלוח (ראה קובץ איפיון). בו יפגעו קרני האינפרה אדום ויחזרו אל קולט האינפרה אדום.
* **IR Receiver –** קולטים את האינפרה אדום שיוחזר מן האובייקט בתהליך הנאמר מקודם.
* **Mux –** יקבל מהרספברי מאיזה קולט לקלוט בכל רגע נתון.
* **ADC –** תפקידו להמיר את המידע שהעביר המוקס מאנלוגי לדיגיטאלי, כדי שהרספברי יוכל לקרוא אותו.
  1. פירוט לכל רכיב :
* **IR Emitters :**נורות אינפרה אדום בודדות, בעלות שני רגליים – מינוס (-) ופלוס (+).  
  את המינוסים נחבר בין כל הנורות (GND – ground), ואת הפלוס של כל נורה נחבר למקום המתאים לה בדימוקס. (מרחק.???)
* **IR Receivers :**

קולטים (נראה כמו נורות האינפרה אדום, אך בצבע שחור).  
בעלות שני רגליים – מינוס (-) ופלוס (+).  
את המינוסים נחבר ביחד לground (של הנורות אינפרא אדום).

את הפלוס בכל קולט נחבר למקום המתאים לו במוקס.  
(מרחק???)

* **Dmux :**

הדימוקס יורכב מ17 דימוקסים, כלומר דימוקס אחד של 16 אליו מחוברים 16 דימוקסים.

[\*\* הערה : נצטרך רק 225 יציאות ו 16×16 = 256 ולכן נוכל לחסוך 2 דימוקסים מכיוון ש 16×14 = 224 ונוכל להשתמש ביציאה שחסכנו בדימוקס הראשי ולכן יהיו לנו 225 יציאות].

נחבר את מקורות המתח של כל הדימוקסים במקביל למקור מתח אחד, משם יגיע המתח אל הדימוקס הגדול. אל הדימוקס הראשון נחבר את ארבעת הביטים העליוניים של הסלקטים, ואל 14-ת הדימוקסים האחרים נחבר את ארבעת הביטי התחתונים של הסלקטים.   
את כל היציאות נחבר את הקולטי אינפרה אדום.

את הכניסה של הדימוקס (common), נחבר למקור חשמל שידליק את המשדר הנבחר (לפי הסלקטים)

* **Mux :**יורכב מאותם רכיבים ובאותו צורה של הדימוקס אך יתפקד כמוקס.  
  את הcommon נחבר לנגד ומשם לADC.  
  את 225 הכניסות נחבר לכל קולט אינפרה אדום.
* **ADC :**אליו נחבר את המוקסים. וזהו זה כל מה אנחנו יודעים כרגע. : P
* **Arduino :**יקבל מידע של תמונה להצגה ללדים. בפרוטוקול I2C ויתרגם אותו בהתאם ללדים (הכרחי לעניין התזמון).
* **Led Strips :**

רצועות לדים במרחקים 1.6 ס"מ.  
נעשה 30 שורות, כשכל שורה היא 0.5 מטר. השורות יהיו במרחק 1.6 ס"מ. (ראה סרטוט בסעיף 5 בקובץ אפיון).   
עם רצועות הלדים נשתמש כמסך, באמצעות הפרוטוקול שמותאם לרכיב ששולט על הלדים (WS2811).

* **Raspberry :**הרספברי אחראי על תקשור עם הארדואינו כדי לשלוט על מסך הלדים. ואחראי על שינוי הסלקטים ותפיסת המידע שהגיע מהADC והרכבת תמונת הINPUT. (עליו בעצם תהיה כל התוכנה והכל פחות או יותר (?)).
  1. דיון לגבי העיצוב הנבחר :
* **Raspberry pi 2 –**

1. לא רצינו להיות מוגבלים בשפות תכנות.
2. היינו צריכים מכשיר בעל כוח עיבוד מהיר כדי:  
   1. לקבל מספיק תמונות בפרקי זמן קצרים (כדי שיראה חלק ולא תקוע).  
   2. שנוכל להשתמש בכלים לעיבוד תמונה הדורשים כוח עיבוד טוב.  
   3. כדי לשדר מספיק תמונות בפרקי זמן קצרים.
3. הגבלת מקום – לא רצינו להיות מוגבלים בזיכרון (ראם ורום)[בשביל המשחקים, בין היתר].  
     
   חסרון – בגלל שהוא מחשב לא נוכל לחברו ישירות לרצועות הלדים (led board), מאחר והוא לא טוב בתזמונים (tight timing).  
   תחליף – ארדואינו.

* **Arduino –**   
  מטרתו לכפר על החיסרון של הרספברי.   
  הוא מקשר בין הרספברי לבין רצועות הלדים ומטפל בעיניין התזמון (tight timing).  
    
  לא חשבנו על תחליף ראוי.
* **Led board –**

ישמש כמסך.  
בחרנו להשתמש בדגם הספציפי: WS2811 משתי סיבות:  
מחיר וזמינות.  
  
חסרונות (במימוש הכללי) :   
א. רזולוציה נמוכה.  
ב. עבודה קשה....  
  
יתרונות : המחיר נמוך ביחס לגודל.

* **Mux / Dmux –**

כדי שנוכל לבחור איזה משדר IR להדליק בכל רגע נשתמש בDMUX.  
כדי שנוכל לקלוט תמונה שלמה של כל המסך נשתמש בMUX שיבחר מאיזה מהקולטים לקרוא את המידע.  
  
בחרנו ברכיב הספציפי – CD74HC4067E, מכיוון שהוא מוקס ודימוקס אנלוגי.  
E – כדי שנוכל להשתמש בו בצורה נוחה.  
  
חסרונות :   
עבודה קשה (הלחמות והרבה).

* **IR Emitters –**  
  חלק מהקונספט של קליטת מרחק.  
    
  יתרונות של הקונספט של הIR:  
  א. נורות הIR (קולטים + משדרים) יהיו ממש על הלוח, ולא יבלטו ממנו.  
  ב. מבחינת חישובים ועיבוד תמונה יהיה יחסית פשוט.  
    
  חסרונות :   
  א. הרבה הלחמות.  
  ב. הרבה נורות, מה שמצריך הרבה רכיבים.  
  ג. רזולוציה נמוכה (יחסית).  
  ד. תמונת הקלט – רק שחור-לבן ( grey-scale).  
  ה. יקר (יחסית [לי]).
* **IR Receiver –**

חלק מהקונספט של קליטת מרחק (ראה IR Emitters).  
  
חסרונות :   
קולט גם אינפרה אדום מהסביבה ולכן נצטרך לטפל בזה.

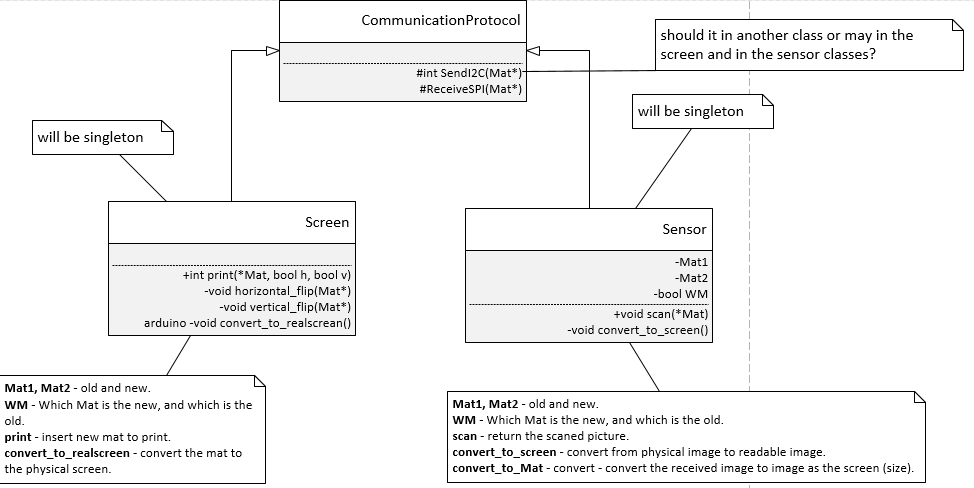
* **ADC –**

רכיב שיאפשר לנו לקרוא את המידע (האנלוגי) מהקולטים ברספברי (דיגיטלי).  
בחרנו ברכיב הספציפי: MCP3004 מכמה סיבות:  
א. יש לו מהירות דגימה שתספיק לנו (ואף יותר ממה שנצטרך).  
ב. מתקשר בפרוטוקול נתמך ברספברי (SPI).  
ג. יש לו מספיק כניסות.

1. עיצוב הנתונים (data)

**תיעוד מחלקות**

התקני קלט ופלט כלליים



**CommunicationProtocol :**

המחלקה אחראית על תקשורת עם הארדואינו (לדים) ועם הADC (קולטים)

**: Methods**

* #**int SendI2C (Mat\*)**   
  מתודה זו מתקשרת בין הרספברי לארדואינו,   
  היא שולחת את המתריצה שקיבלה (תמונה) אל הארדואינו בפרוטוקול .I2C
  + מקבלת: פוינטר למטריצה שתישלח לארדואינו בפרוטוקול I2C
  + מחזירה: מספר המיצג ERROR\SUCCESS
* **#int ReceiveSPI(Mat\*)**  
  מתודה זו מתקשרת בין הרספברי לADC ומקבלת ממנו את המטריצה שקלט.
* מקבלת: פוינטר למטריצה שתמלא בתמונה שתתקבל.
* מחזירה: מספר המיצג ERROR\SUCCESS

**: Screen**

המחלקה תממש מסך, אחראית על קבלת התמונה לשליחה, ושליחתה.  
המחלקה תהיה יחידה בכל המערכת (singletone).  
 **: Methods**

* **+int print(\*Mat, bool h, bool v)**

אחראית על הדפסת תמונה למסך

* מקבלת : פוינטר למטריצה – המטריצה להדפסה.  
   דגל – האם להפוך את התמונה לאורך.  
   דגל – האם להפוך את התמונה לרוחב.
* מחזירה : מספר המיצג ERROR\SUCCESS
* **-void horizontal\_flip(Mat\*)**  
  מתודה להפיכת תמונה לרוחב.
* מקבלת: פוינטר למטריצה להפיכה.
* מחזירה:כלום.
* **-void vertical\_flip(Mat\*)**  
  מתודה להפיכת תמונה לאורך.
* מקבלת: פוינטר למטריצה להפיכה.
* מחזירה: כלום.

**: Sensor**המחלקה אחראית על קריאת המידע מהADC ולשמש כהתקן לקריאת תמונות.  
המחלקה תהיה יחידה בכל המערכת (singleton).  
 **: Fields**

* -Mat1, -Mat2 - תמונות לכתיבה ולקריאה.
* bool WM – אם True אז Mat1 לקריאה וMat2 לכתיבה,

אם False אז Mat2 לקריאה וMat1 לכתיבה.

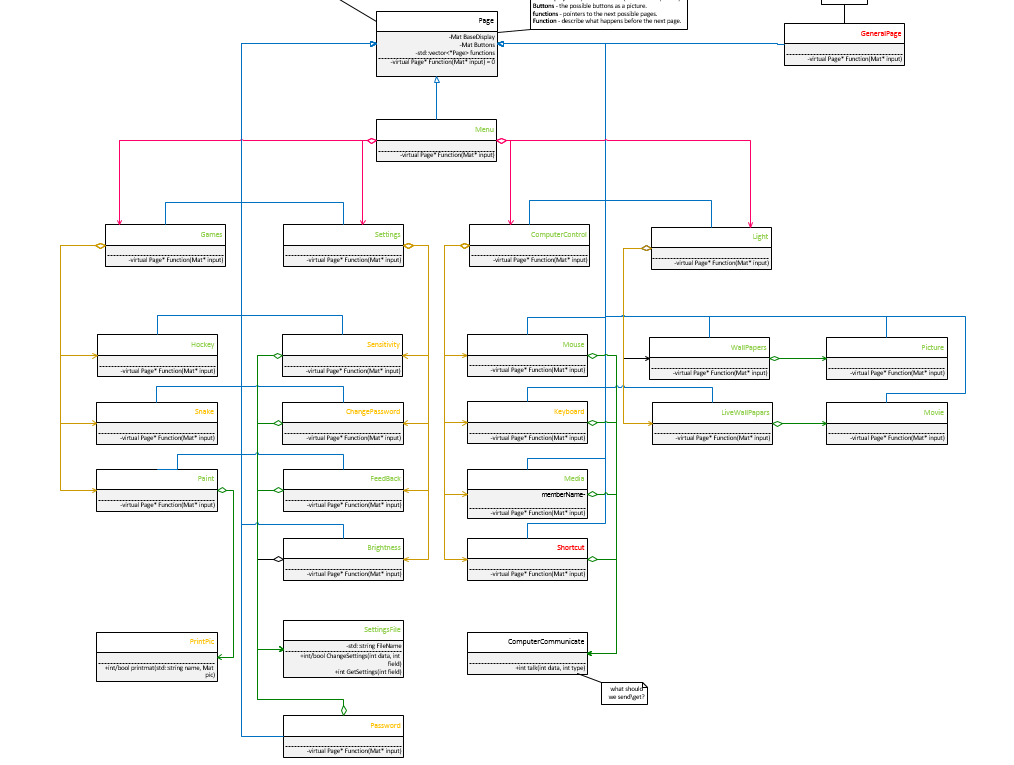
**: Methods**

* **+void scan(\*Mat)**  
  מתודה לקליטת תמונה.
* מקבלת: פוינטר למטריצה בה תאוכסן התמונה.
* מחזירה : כלום
* **-void convert\_to\_screen()**

מתודה להמרת המטריצה לגודל המסך.

* מקבלת: כלום
* מחזירה : כלום

מערכת התפריט ועיבוד המידע



**Page :**מחלקה אבסטרקטית כללית ליצוג המסך הנוכחי.

**: fields**

* **-Mat BaseDisplay** **:** תמונת הפלט של המסך הנוכחי.
* **-Mat Buttons** **:** תמונת הקלט של המסך הנוכחי.
* **-std::vector<\*Page> functions** : פויינטר לדפים האפשריים הבאים.

**: methods**

* **-virtual Page\* Function(Mat\* input) = 0**  
  פונקציית הפעולה. קובעת מה ייקרה בעת לחיצה / קלט מסויים.
* מקבלת : תמונת קלט
* מחזירה : פויינטר לעמוד הבא

**Menu :**

המחלקה מממשת את התפריט הגרפי למשתמש.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : light , settings, games, computer control.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Light**

המחלקה מממשת תפריט גרפי למשתמש.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : wallpaper, live wallpaper

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: ComputerControl**

המחלקה מממשת תפריט גרפי למשתמש.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : mouse, keyboard, media, shortcuts.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Settings**

המחלקה מממשת תפריט גרפי למשתמש.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : sensitivity, brightness, ChangePassword, Feedback

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Games**

המחלקה מממשת תפריט גרפי למשתמש.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : snake, hokey

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Wallpapers**

המחלקה מממשת רשימת שמות של קבצים (תמונות).  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : Picture, self (next list)

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Picture**

המחלקה מממשת תמונה להצגה.

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: LiveWallpapers**

המחלקה מממשת רשימת שמות של קבצים (סרטים).  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחד מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון : Movie, self (next list)

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Movie**

המחלקה מממשת סרטון, ותשתמש בPicture כדי להציג כל פריים.

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**: Mouse**

המחלקה מממשת עכבר, תפקידה להמיר את תמונת הקלט למקום גיאומטרי של העכבר ולהעבירו אל ComputerCommunicate.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:  
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -  
ב. מיקום של היד ושליחה ל ComputerCommunicate.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**Keyboard :**  
המחלקה מממשת מקלדת, תפקידה להמיר את תמונת הקלט ללחיצה על לחצני מקלדת (אותיות\מספרים) ולשלוח ל ComputerCommunicate.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על אותיות\מספרים ושליחה ל ComputerCommunicate.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**Media :**המחלקה מממשת מקלדת של מדיה (לחצני מדיה) תפקידה להמיר את תמונת הקלט ללחיצה על לחצני מדיה (ווליום, פלי, סטופ) ולשלוח ל ComputerCommunicate.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על לחצני המדיה ושליחה ל ComputerCommunicate.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**Shortcut :**  
המחלקה מממשת מקלדת לחצני קיצור, תפקידה להמיר את תמונת הקלט ללחיצה על לחצני קיצור (בהירות, שינה, צילום מסך), ולשלוח ל ComputerCommunicate.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על לחצני קיצור ושליחה ל ComputerCommunicate.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא

**ComputerCommunicate :**המחלקה תתקשר עם המחשב דרך פרוטוקול קיים (לא ידוע איזה...).   
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* פונקציות לתקשורת..
* יקבלו : מה לשלוח.
* יחזירו : האם הצליח.

**Sensitivity :**המחלקה אחראית על תצוגת וקליטת מסך שינוי רגישות התגובה.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על המד ושליחת המידע לSettingsFile.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**ChangePassword :**המחלקה אחראית על תצוגת וקליטת מסך שינוי הסיסמא.   
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על הכנסת הסיסמא ושליחת המידע לSettingsFile (אחרי הכנסת כל הסיסמא).

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**FeedBack :**המחלקה אחראית על תצוגת וקליטת מסך שינוי הגדרות הfeedback.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על "כן" או "לא" ושליחת המידע לSettingsFile.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**Brightness :**המחלקה אחראית על תצוגת וקליטת מסך שינוי הבהירות.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תבדוק אם תמונת הקלט מייצגת:   
א. לחיצה על אחת מהאפשרויות (עמודים הבאים), ותחזיר את העמוד הנכון self -

ב. לחיצה על "כן" או "לא" ושליחת המידע לSettingsFile.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**SettingsFile :**המחלקה אחראית על פתיחת קובץ ההגדרות, כתיבתו, קבלת מידע ועדכון הקובץ, והצפנת הקובץ.

**Fields:**

* **-std::string FileName :** שם קובץ ההגדרות.

**: methods**

* **+int/bool ChangeSettings(int data, int field)**  
  הפונקציה תעדכן קובץ הגדרות קיים.
* מקבלת : data – המידע החדש.  
   field – השדה שאותו צריך לעדכן.
* מחזירה : האם הצליח או שלא הצליח.
* **+int GetSettings(int field)**  
  הפונקציה מחזירה את תוכן השדה המבוקש מתוך קובץ ההגדרות.
* מקבלת : שדה מתוך קובץ ההגדרות.
* מחזירה : את המידע מתוך השדה המבוקש.

**Password :**המחלקה אחראית על קליטת הסיסמא מהמשתמש ובדיקתה.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תקלוט מתמונות הקלט את הסיסמא שהמשתמש הקליד, אם היא נכונה היא תחזיר את Menu ובכל מקרה אחר תחזיר self.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעמוד הבא.

**Paint :**תבדוק איפה היד נמצאת ותצבע את המקום בוא היא נמצאת אם היא נמצאת במרחק המוגדר לצביעה (המרחק יגדיר את גודל "המברשת"). בצד תיהיה תצוגה של צבעים ונוכל לבחור את הצבע הרצוי.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תקלוט את היד ותצייר איפה שהיא לפי עוצמת הקליטה.

* מקבלת : תמונת קלט.
* מחזירה : פויינטר לעצמו.

**PrintPic :**המחלקה תקבלת תמונה ותשמור אותה לתקיית הWallPapers כדי שנוכל להציגה כשנרצה.

**: methods**

* **+int/bool printmat(std::string name, Mat pic)**  
  הפוקציה תכתוב את התמונה שקיבלה לקובץ.
* מקבלת : תמונה לכתיבה, שם הקובץ.
* מחזירה : האם הצליחה.

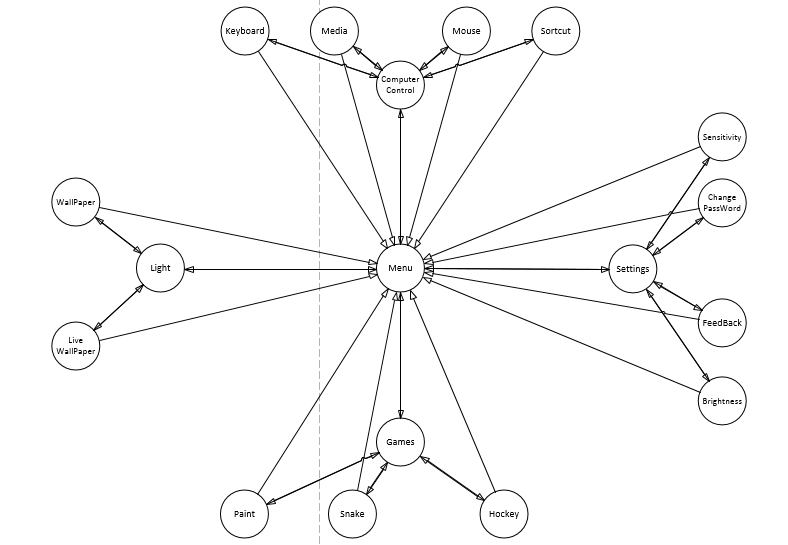
**Hockey :**תריץ משחק הוקי ותתרגם אותו לממשק הלוח על ידי קלט של הלוח ופלט של הלוח.  
המחלקה תהיה יחידה בכל המערכת (singletone).

**: methods**

* **-virtual Page\* Function(Mat\* input)**

הפונקציה תחשב האם האובייקטים נמצאים במרחק לחיצה מעל ל'דסקיות'.

* תקבל : תמונת קלט.
* תחזיר : תחזיר את עצמה.

4 **ממשק המשתמש** :

1. **נספחים**  
   כל דבר נוסף שרציתם לציין ולא היה בשבילו מקום ייעודי מוגדר :)

**חשיבות נוספות :**

השפות שנשתמש בהן :

**C++** (שימוש בתרדים, שימוש בתמונות (openCV או משהו אחר))

התרדים :

תרד – לפלט

תרד – לקלט

**תרד – לחישובים, בעצם יעשה את כל התוכנה ויסנכרן בין שני התרדים הראשונים.**