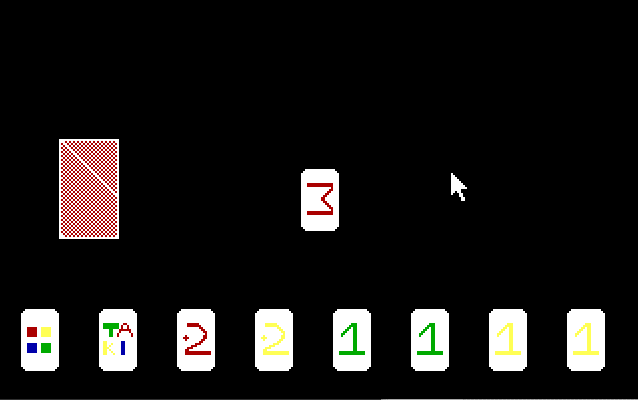


**פרויקט אסמבלי**



**טאקי**

מגיש: יונתן פרידקין

ת.ז: 209602564

כיתה י'11

מוגש למורה: הילה גורן ברנע

שנת לימוד: 2021 – 2022

**תוכן עניינים**

מבוא.................................................................................................................3

מבנה.................................................................................................................4

מדריך למשתמש..................................................................................................5

מדריך למפתח....................................................................................................6

משתנים.............................................................................................................13

רפלקציה............................................................................................................15

ביבליוגרפיה......................................................................................................16

**מבוא**

**שם המשחק** – טאקי

**שם הקובץ** – Fridkin.asm

**קבצים נלוויים** – 4 קבצי BMP ( מסך תחילת משחק, תמונת חוקים, תמונות ניצחון של השחקן הראשון ושל השחקן השני)

**שפת פיתוח** – Assembly

**סביבת העבודה** – Turbo Assembler

**סביבת הפיתוח** – NotePad++

**סביבת הרצה** – DosBox

**מבנה**

המשחק עצמו מורכב מלולאה שרצה כל עוד אף אחד מהשחקנים לא ניצח ( גודל שתי הידיים הוא יותר מאפס). בכל תור, הלולאה מזהה מי השחקן שתורו עכשיו. לאחר מכן, היא מציגה את יד הקלפים שלו ומחכה לקלט תקין (לחיצה על ערימת הקלפים או שיחוק קלף מתאים), לאחר מכן היא מתייחסת לאפקטים של הפעולה ומסיימת את התור הנוכחי.

הקוד למשחק מורכב כולו ממספר פרוצדורות שונות, שלכל אחת תפקיד שונה ובלעדי עליו היא אחראית. ניתן לחלק את הפונקציות שבמשחק לשתי קטגוריות: גרפיות ולוגיות.

להלן טבלה שמציגה את הקטגוריה של כל פרוצדורה:

|  |  |
| --- | --- |
| פרוצדורות גרפיות | פרוצדורות לוגיות |
| Sprite\_Draw | CanPlayCard |
| Show\_Card | PlayACard |
| Sprite\_Figure | DrawXCards |
| Show\_Card | Shuffle\_Deck |
| Special\_Sprite\_Figure | Main\_Game |
|  | Is\_On\_Card |

בהמשך, אפרט כל אחת מן הפונקציות, על תפקידה ואיך היא עובדת.

**מדריך למשתמש**

המשחק שכתבתי מיועד לשני שחקנים שמשחקים זה נגד זה. בכל תור אחד השחקנים משחק והשני מסיט את מבטו. בסוף כל תור, מופיעה ספירה לאחור על המסך והתור מתחלף.

**איך לשחק?**

ניתן לשחק קלף רק אם הצבע או הערך שלו זהים לאלו של הקלף האחרון ששוחק ( הקלף שמופיע במרכז המסך). נוסף על הקלפים שערכם הוא בין אחד לתשע קיימים מספר קלפים מיוחדים עם אפקטים שונים.

הקלפים המיוחדים הם:

שנה צבע – ניתן לשחק את קלף זה בכל סיטואציה. לאחר ששיחק, השחקן לוחץ על מספר בין 1 ל-4 בשביל לבחור את הצבע החדש. בשביל לשנות לצבע אדום יש ללחוץ על 1, בשביל כחול על שתיים, בשביל ירוק על שלוש ובשביל צהוב על ארבע.

2+ - ברגע ששיחק שחקן קלף זה, תורו נגמר. אם לשחקן השני אין יכולת להגיב עם 2+ נוסף הוא נאלץ לשלוף מהקופה 2 קלפים. במידה ויש לו קלף, הוא מחויב לשחק אותו. אם לשחקן הראשון אין 2+ נוסף להגיב איתו, הוא מחויב לשלוף 4 קלפים. התהליך חוזר על עצמו עד שלאחד השחקנים אין 2+ להגיב איתו.

טאקי – לאחר ששחקן שיחק קלף זה הוא יכול לשחק מהיד כמות קלפים בלתי מוגבלת. שימו לב, אחרי טאקי ניתן לשחק רק קלפים בעלי אותו צבע.

סופר טאקי – קלף הסופר טאקי זהה לקלף הטאקי מלבד העובדה שהוא מקבל את צבעו של הקלף האחרון ששוחק לפניו.

עצור – קלף העצור מדלג על תורו של השחקן הבא בסבב. במשחק בעל שני שחקנים, שיחוק קלף העצור מזכה את השחקן בתור נוסף.

פלוס – קלף הפלוס מזכה שחקן בתור נוסף. במידה ולשחקן אין מה לשחק לאחר ששיחק פלוס, הוא צריך לשלוף קלף.

שינוי כיוון – קלף שינוי הכיוון משנה את כיוון המשחק. במשחק עם שני שחקנים, משחק של קלף העצור מזכה את השחקן בתור נוסף.

**מדריך למפתח**

כאמור, הקוד בנוי כולו מפרוצדורות עצמאיות בעלות תפקידים שונים. כעת אסביר את תפקידה של כל פונקציה, את הפרמטרים שלה ואת מה שהיא מחזירה. בנוסף, עבור כל פונקציה משמעותית יצורף תרשים זרימה המסביר את הלוגיקה מאחוריה.

הערות:

1. את הגרפיקה של מהלך המשחק ( קלפים וערימת הקלפים) אני מדפיס בשיטה בשם sprites ( ראה ביבליוגרפיה) . בשיטה זו, אני מגדיר מערך "דו מימדי" עם מספר שורות ועמודות כלשהו. בהתחלה, אני ממלא את כל התאים באפסים ואז מסמן תאים מסוימים בערכי 1 בצורה שאני רוצה שתתקבל.
2. ייצוג הקלפים ( צבע וערך ) מתבצע בשיטה הבאה: קלף כלשהו מיוצג על ידי שני מספרים: מספר שמציין את הערך שלו וספרה נוספת שמציינת את הצבע שלו. כך לדוגמה 61 הוא 6 אדום ו62 הוא 6 כחול. ניתוח המידע של הקלפים מתבצע באמצעות חלוקה בעשר של הערך. התוצאה השלמה היא המספר והשארית היא הצבע.
3. אני שומר את ערך הקלף האחרון ששוחק בשני מקומות: גם בערימת הקלפים ששוחקו ( DiscardPile) וגם בתוך שני משתנים בשם Last\_Color ו – Last\_Value

כעת אציג את הפונקציות השונות בהן אני נעזר בקוד. להלן רשימת הפרוצדורות:

**פונקציות ראשיות**

Shuffle\_Deck – פונקציה זו מערבבת את מערך הקלפים הראשי. היא עוברת בלולאה רגילה על המערך ובכל ריצה מגרילה אינדקס רנדומלי. לאחר מכן, היא מחליפה בין הערכים של האינדקס הנוכחי (מתוך הלולאה) והשל האינדקס הרנדומלי.

Sprite\_Figure – פונקציה זו מקבלת sprite, מיקום התחלתי על המסך וצבע. היא עולה 8 פיקסלים למעלה וזזה 7 פיקסלים שמאלה ( גודלם של הsprites של הערכים הם 15 X 16 כך שמתקבל שהנקודה שנשלחה כפרמטר לפונקציה היא אמצע הצורה שמתקבלת על המסך). לאחר מכן היא מתחילה לקרוא את המערך מהנקודה הזו, מתעלמת מאפסים ומדפיסה פיקסלים עם ערך המערך הנוכחי הוא 1. הפונקציה בנויה מלולאה מקוננת בה הלולאה החיצונית היא מספר השורות והלולאה הפנימית היא גודל שורה. הערה: הפונקציה מדפיסה פיקסלים למסך באמצעות חישוב הפיקסל הנוכחי בזיכרון הגרפי וכותבת לתוכו ישירות את הצבע.

Card – הפונקציה הזו פועלת על אותו עיקרון כמו הפונקציה הקודמת. קריאה לפונקציה מדפיסה קלף ריק, במיקום שהתקבל כפרמטר. גודל הsprite של קלף ריק הוא 31 X 19 כך שלפני תחילת ההדפסה הפונקציה עולה 15 פיקסלים למעלה וזזה 9 פיקסלים שמאלה.

Show\_Card – הפונקציה הזו מקבלת מיקום על המסך וערך של קלף. תחילה היא עוברת על הקלף ומנתחת את רכיבי הצבע והערך שלו. לאחר מכן, היא בודקת את כל התנאים השונים להדפסת קלף ( לדוגמה: אם al = 6 ן ah = 1 הפונקציה תדפיס על המסך 6 אדום). לאחר ניתוח הנתונים הפונקציה קוראת לפונקציה Card ולפונקציה Sprite\_Figure

DrawXCards – פונקציה זו שולפת לאחד השחקנים כמות קלפים כלשהי. הפונקציה מקבלת כפרמטרים את יד השחקן, את הגודל הנוכחי שלה, את מערך הקלפים הראשי ואת גודלו. הפונקציה בונה לולאה שרצה ככמות הקלפים שצריך לשלוף ועל כל ריצה שלה היא מעתיקה מן המערך הראשי אל היד את הקלף הראשון שאינו אפס ואז מאפסת אותו.

CanPlayCard – פונקציה זו בודקת האם ניתן לשחק קלף כלשהו. הפרמטר היחיד שמתקבל הוא ההיסט בזיכרון של הקלף (offset). הפונקציה מנתחת את הנתונים של הקלף ומשווה אותם למשתנים Last\_Color ו – Last\_Value. הפונקציה מחזירה בתוך cx את הערך חמש במידה וניתן לשחק את הקלף.

Visual\_Deck - פונקציה זו מציגה על המסך יד קלפים של שחקן. היא מקבלת כפרמטרים את offset היד ואת מיקום התחלת ההדפסה. קודם כל, הפונקציה מחשבת ומחזירה בתוך dx את המרחק הקבוע שיהיה בין כל שני קלפים ( החישוב בנספחים). לאחר מכן היא עוברת על כל אחד מאברי המערך וקוראת לפונקציה Show\_Card. בסוף איטרציה אני מוסיף לערך ה-X סך של 18 ועוד המרווח שחושב ( קלף תמיד מודפס מהאמצע ולכן כדי להגיע לאמצע הקלף הבא יש להוסיף גודל חצי קלף ועוד המרווח בין שני קלפים ועוד 9 בשביל להגיע לאמצע הקלף הבא).

Re\_Show\_Screen – הפונקציה מאפסת ומדפיסה מחדש את כל רכיבי הגרפיקה השונים שמופיעים במסך ( יד השחקן, קלף אחרון ששוחק וערימת הקלפים). תחילה היא מנקה את המסך בכל שנכנסת מחדש למצב גרפי, לאחר מכן היא קוראת לפונקציה Visual\_Deck על יד השחקן ( הפונקציה מקבלת כפרמטר את יד השחקן בתוך si ואת גודל היד בתוך bx ). לאחר מכן היא קוראת לפונקציה Show\_Card על הקלף שבמרכז המסך ומדפיסה את ערימת הקלפים.

Open\_Picture – הפונקציה מציגה על המסך את תמונת הפתיחה. לחיצה על 1 מתחילה משחק חדש ולחיצה על 2 עוברת לתמונת החוקים.

Rules – פונקציה שמדפיסה על המסך את תמונת החוקים. יוצאת רק לאחר הקשת הלחצן ESC.

GraphicMode – הפונקציה עוברת למצב גרפי ובכך גם מנקה את המסך מרכיביו השונים.

**פונקציות משניות**

WaitForLeftClick – הפונקציה מחכה שהמשתמש ילחץ לחיצה שמאלית על העכבר

WaitForEnterKey – הפונקציה מחכה שהמשתמש ילחץ על מקש האנטר

WaitForColor – הפונקציה מחכה שהמשתמש יכניס מספר בין 1 ל-4 בשביל לבחור את הצבע לאחר שיחוק שינוי צבע.

Current\_Turn – הפונקציה קוראת את תור השחקן הנוכחי ומדפיסה על המסך 1 או 2 בהתאם בצבעים אדום או כחול.

Mouse – הפונקציה מאתחלת את העכבר.

Initialize\_Mouse – הפונקציה מאתחלת את העכבר.

Limit\_Mouse – הפונקציה מגבילה את טווח התנועה של העכבר לגבולות המסך.

Hide\_Mouse – הפונקציה מסתירה את העכבר.

Print\_Draw\_Pile – הפונקציה מדפיסה על המסך את ערימת הקלפים.

**פרוצדורות גרפיקה וBMP**

Openfile – פתיחת קובץ BMP

ReadHeader – קריאת 54 הבתים הראשונים של קובץ BMP

ReadPalette – קריאת פלטת הצבעים של התמונה

CopyPal – העתקת הפלטה לזיכרון הגרפי

CopyBitMap – הדפסת התמונה על המסך

CloseFile – סגירת הקובץ

**תרשימי זרימה**

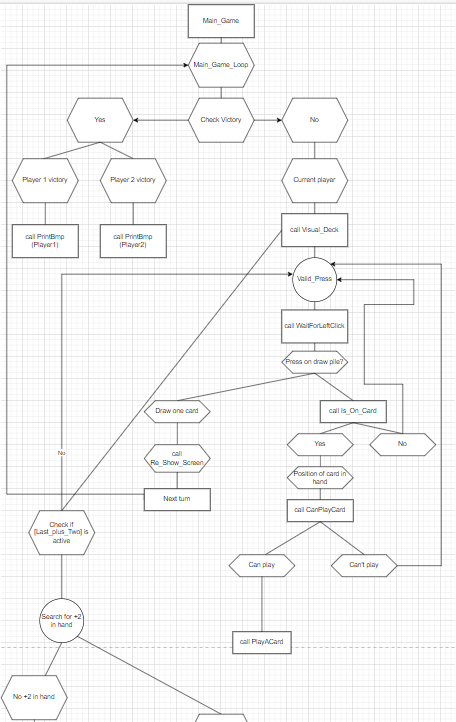
בחלק זה אציג תרשים זרימה של המשחק הראשי.

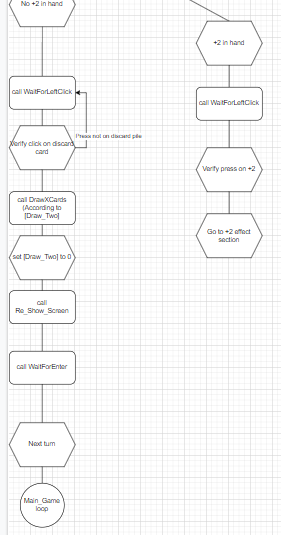
מקרא תרשים הזרימה:

לולאה

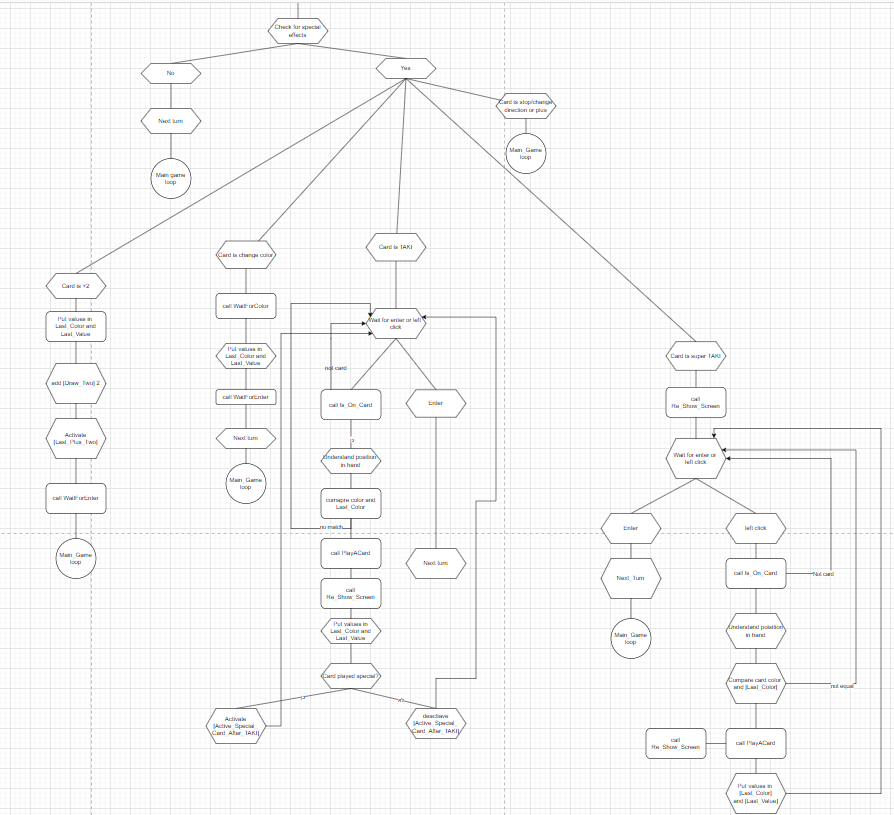
פעולה המתבצעת בתוך הפרוצדורה

קריאה לפרוצדורה





חלק זה מציג את האפשרות של שליפת קלף ושיחוק קלף מתאים, כעת אציג את ההתחשבות באפקטים המיוחדים של הקלפים.



**משתנים**

משתנים שהגדרתי ובהם נעזרתי לקוד הם:

Global\_arr – מערך הקלפים הראשי המכיל את כל קלפי המשחק השונים.

Global\_arr\_size – משתנה המתאר את גודלו הנוכחי של המערך Global\_arr. לאחר קריאה לפונקציה DrawXCards ערכו קטן בהתאם למספר הקלפים שנשלפו.

Player1 – מערך זה מחזיק את ערכי הקלפים של השחקן הראשון. המערך מתחיל מלא באפסים וכל שליפה של קלפים ממלאת אותו בהתאמה.

Player1Size – משתנה המתאר את גודלה הנוכחי של יד השחקן הראשון

Player2 – מערך זה מחזיק את ערכי הקלפים של השחקן השני. המערך מתחיל מלא באפסים וכל שליפה של קלפים ממלאת אותו בהתאמה.

Player2Size – משתנה המתאר את גודלה הנוכחי של יד השחקן השני.

Player\_Turn – משתנה שערכו הוא 0 או 1. ערכו קובע לפונקציה Main\_Game מי השחקן שכעת תורו. בסוף כל תור ערכו משתנה בצורה מעגלית: מ-0 ל-1 ומ-1 ל-0.

Draw\_Two – משתנה המתפקד למקרי שיחוק 2+. המשתנה עולה בערכו ב2 בכל פעם שמשוחק קלף זה וקובע כמה קלפים צריך לשלוף בסוף כל סבב של קלפי 2+. אחרי שליפת הקלפים באירוע ה-2+ ערכו מתאפס.

Last\_Color – משתנה השומר את צבעו של הקלף האחרון ששוחק. הפונקציה CanPlayCard משווה את קלף הבדיקה עם משתנה זה ועם Last\_Value

Last\_Value – משתנה השומר את ערכו של הקלף האחרון ששוחק. הפונקציה CanPlayCard משווה את קלף הבדיקה עם משתנה זה ועם Last\_Color

DiscardPile – מערך הקלפים ששוחקו. כל קריאה לפונקציה PlayACard מעתיקה מן היד אל DiscardPile

Discard\_Pile\_Size – משתנה המתאר את גודלו הנוכחי של המערך DiscardPile. מאפשר גישה נוחה לקלף האחרון שקיים במערך זה.

Current\_Index – בכל לולאה של המתנה לשיחוק קלף תקין, המשתנה מקבל את מיקומו של קלף היד במטרה לאפשר חופשיות בשימוש עם הרגיסטרים למטרות אחרות.

Current\_Value – המשתנה נועד בשביל להחזיק את ערכו של קלף שנבחר, במידה ואפשר לשחק אותו.

Interval – משתנה השומר את המרחק הקבוע בין כל קלף לקלף על המסך, לאחר כל קריאה לפונקציה Visual\_Deck, ערכו מתעדכן למרווח הנוכחי בין הקלפים.

Active\_Special\_Card\_After\_Taki – משתנה שערכו הוא 0 או 1 ומתאר האם הקלף האחרון ששוחק במצב הטאקי הוא מיוחד ( עצור, שנה כיוון, פלוס, 2+, טאקי או סופר טאקי). אם הוא פעיל ( ערכו הוא 1) בעת יציאה מלולאת הטאקי מתבצעת קפיצה לאפקט המיוחד המתאים.

Active\_Special\_Card\_After\_Super\_Taki – משתנה שערכו הוא 0 או 1 ומתאר האם הקלף האחרון ששוחק במצב הסופר טאקי הוא מיוחד (עצור, שנה כיוון, פלוס, 2+, טאקי או סופר טאקי). אם הוא פעיל ( ערכו הוא 1) בעת יציאה מלולאת הסופר הטאק

**רפלקציה**

כפרויקט האמסבלי של מגמת הנדסת תוכנה בחרתי ליצור את המשחק טאקי באמסבלי. תהליך יצירת המשחק הייתה עבורי מהנה בחלקים מסוימים וגם מעייפת וקשה בחלקים אחרים. בהתחלה, לא ממש התמצאתי בשפת האמסבלי וקראתי הרבה חומרים באינטרנט במטרה לשפר את הידע. לאחר מכן, ככל שעבר הזמן התחלתי לצבור תאוצה בעבודה ולעבוד עוד ועוד שעות במהלך היום, עד שלקראת הסוף עבדתי בסביבות 5 שעות ביום על הפרויקט.

קשיים מרכזיים שהיו לי הם: להתחיל את העבודה ולהבין את הכיוון הכללי שלה. בזבזתי הרבה זמן בניסיון לחשוב מראש על הרבה פונקציות ופעולות במקום להתחיל לכתוב דברים פשוטים ולהבין אחר כך דברים אחרים. נוסף על כך, לקראת הסוף, גיליתי שהפונקציה שאחראית על לחיצה שמאלית על העכבר מחזירה את אותם הערכים בקריאה שנייה בדיבגר, מה שמנע ממני לדבג את הקוד בצורה יעילה. בנוסף, גיליתי שהרצה רגילה של הקוד ודיבוג שלו מחזירים תוצאות שונות ללא שום סיבה ברורה.

ברצוני להודות לחבריי והוריי שתמכו בי מאוד והעלו לי את המוטיבציה ברגעים קשים.

**ביבליוגרפיה**

נעזרתי באתרים ובקטעי הקוד הבאים בשביל לכתוב את הפרויקט שלי:

1. Stack Overflow
2. ספר האסמבלי של תוכנית גבהים ממנו צברתי את רוב הידע שלי וממנו הפונקציות להדפסת תמונה על המסך.
3. הפורום Piazza
4. קוד בקלאסרום ליצירת מספרים רנדומליים