**פרויקט סיום בסייבר – ספר פרויקט**

**שם הפרויקט:** משחק BRUTETRIS

**מגיש:** גיא ברוטמן

**שם המנחה:** ערן רוזנק

**תומך הוראה:** אורי לוי

**מדריך למשתמש**

המשחק טטריס הוא משחק קלאסי שהושק בשנת 1985. במהלך המשחק, על השחקן לכוון צורות המורכבות מקוביות בצבעים שונים, אשר נופלות במהירות הולכת וגדלה מחלקו העליון של המסך. המטרה היא לסדרן בתחתית המסך. כשהקוביות יוצרות שורה שלמה ללא רווח, השורה נעלמת, והקוביות שנערמו מעליה נופלות "קומה" אחת. כשהקוביות נערמות עד לקצהו העליון של המסך, המשחק נגמר.

**מקשים:** Right/Left arrows – תזוזה לצדדים

Down arrows – תזוזה מהירה למטה

Up arrows – היפוך הצורה

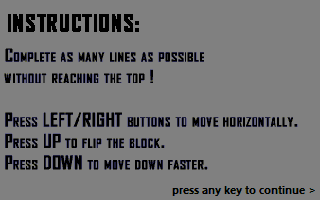
Ctrl key – השהיית המשחק (Esc key להפסקת השהייה)

Esc key – יציאה מהמשחק

**מסכים: מסך הפתיחה –** הצגת שם המשחק וקרדיטים. יש ללחוץ על מקש ה – Enter כדי להמשיך.

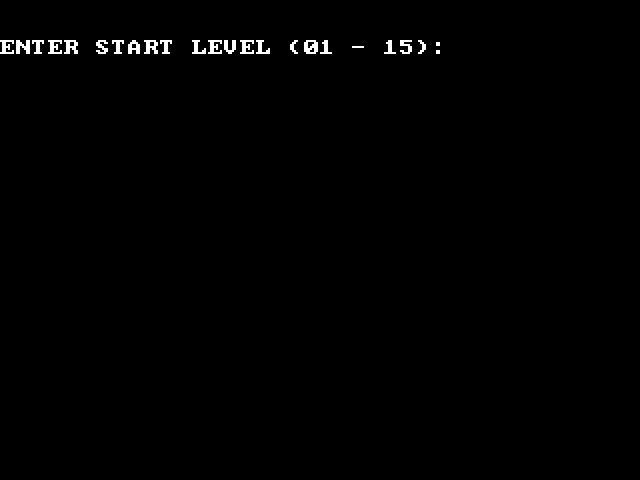
­­­­­­­­­

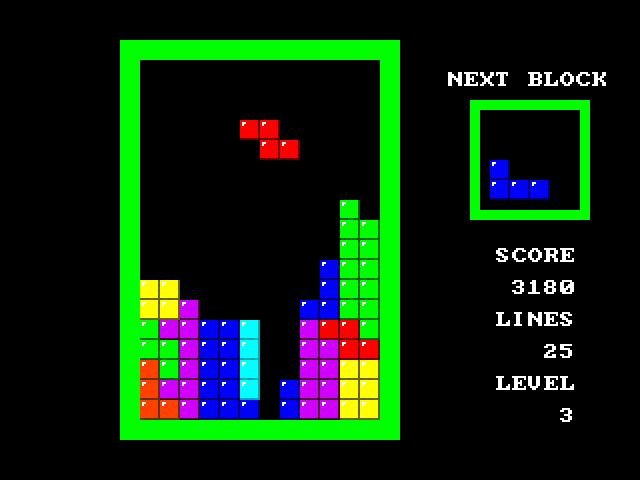
**מסך הוראות** **–** הצגת הוראות המשחק. ניתן ללחוץ על כל מקש בכדי להתחיל את המשחק.



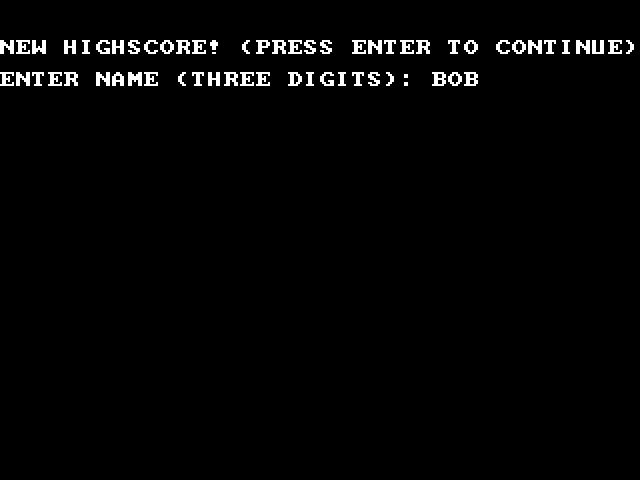
**מסך בחירת הרמה** **–** קליטת הרמה ההתחלתית. השחקן בוחר את הרמה בה יתחיל

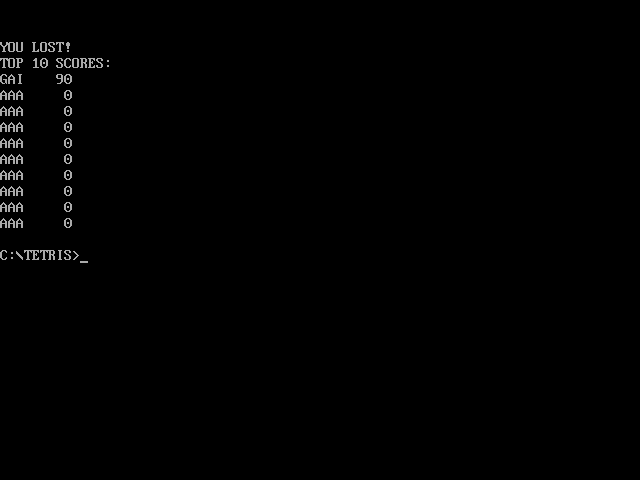
המשחק (מהירות נפילת הצורות תלויה ברמה).

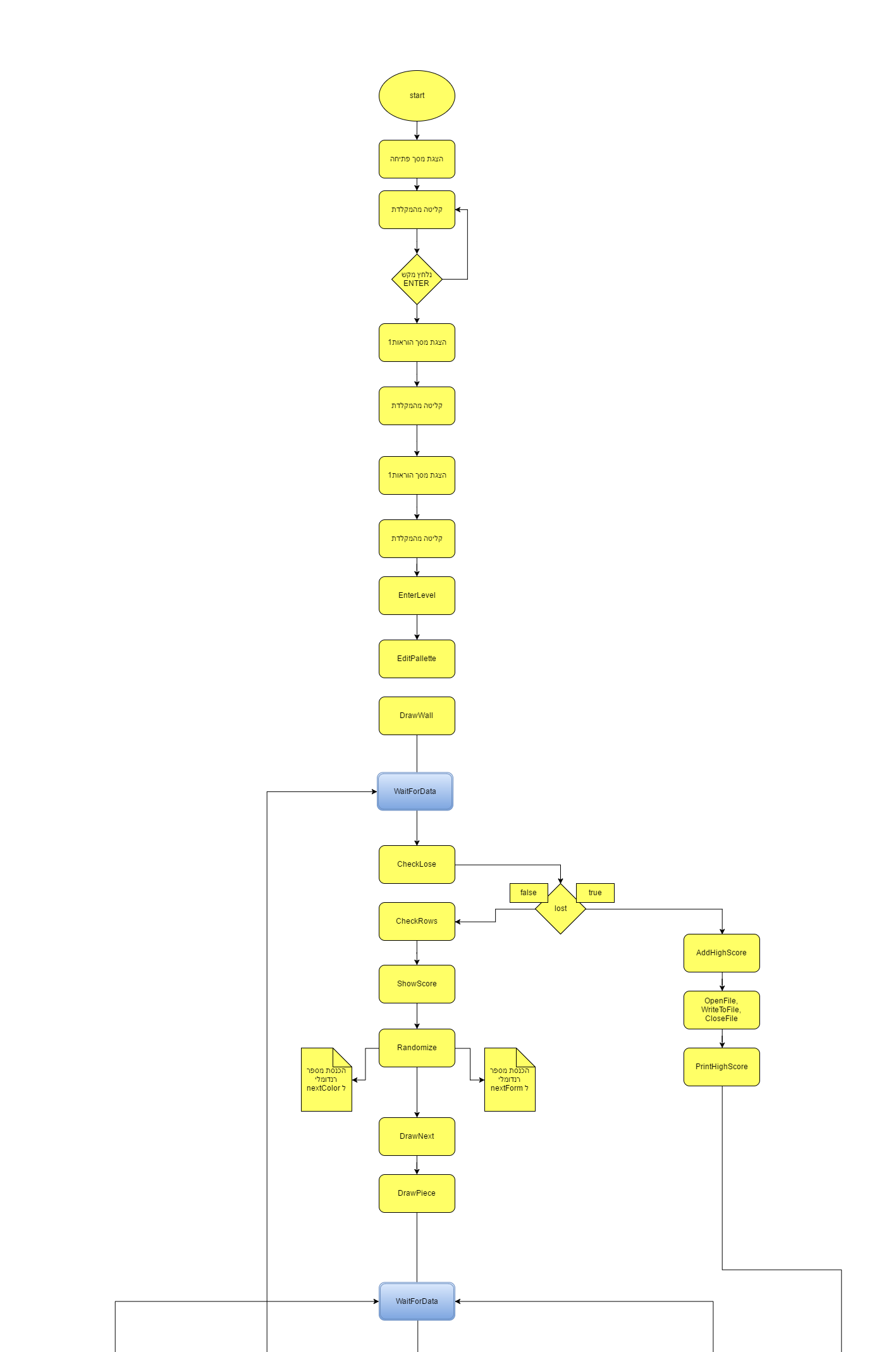


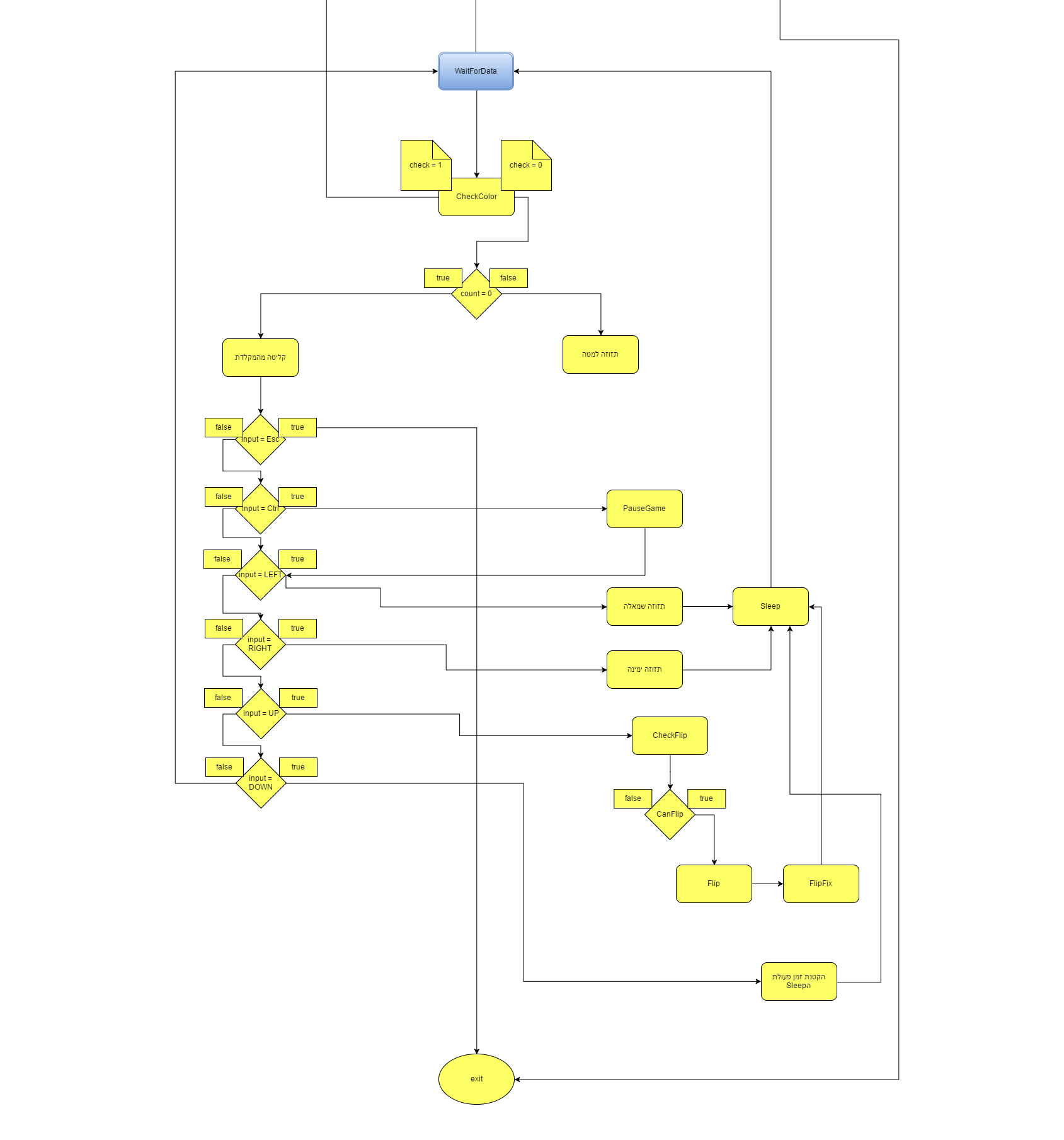
**מסך המשחק** **–** תצוגת המשחק. על המסך מוצג הניקוד, מספר השורות שהושלמות הרמה והצורה הבאה שהולכת לצאת

**מסך קליטת השם** **–** במקרה והניקוד שלך בוא בין העשרה הכי טובים, על המסך מוצג

הודעה ונקלט מהמשתמש קיצור שמו

**מסך לוח הניקוד** **–** על המסך מוצגים שמות עשרת השחקנים הכי טובים והניקוד שלהם.

**תרשים זרימה**

****

**רשימת פרוצדורות**

|  |  |  |  |
| --- | --- | --- | --- |
| **שם הפרוצדורה** | **מה היא מבצעת** | **קלט** | **פלט** |
| OpenShowBMP | מבצעת הדפסה של תמונה על המסך |  | הודעת שגיאה במקרה ויש תקלה בקריאת הקובץ |
| OpenBmpFile | מבצעת קריאה של קובץ התמונה |  | התרעה על שגיאה במידה ויש שגיאה בקריאת הקובץ |
| CloseBmpFile | סוגרת את קובץ התמונה |  |  |
| ReadBmpHeader | קוראת את ה – Header של התמונה |  |  |
| ReadBmpPalette | קוראת את פלטת הצבעים בתמונה |  |  |
| CopyBmpPalette | מעתיקה את הפלטה לזיכרון המסך |  |  |
| ShowBMP | מציגה את התמונה | מיקום התמונה, גודל התמונה והשם שלה |  |
| PrintBmp | קוראת להדפסת תמונת BMP |  | מיקום התמונה, גודלה והשם שלה |
| Sleep | מבצעת השהייה בתכנית תוך שימוש בלולאה |  |  |
| PauseGame | משהה את המשחק בעזרת לופ, מציירת על המסך ריבוע בו רשום Pause. הפרוצדורה משתמשת בPauseSave וב PauseDraw כדי לא לאבד את הצורות מתחת לריבוע |  | ריבוע כחול בו רשום Pause |
| PauseSave | שומרת את הצבעים מתחת לריבוע הPause |  | מערך צבעים מתחת לריבוע הPause |
| PauseDraw | מדפיסה את הצבעים מתחת לריבוע הPause השמורים במערך | מערך צבעים מתחת לריבוע הPause |  |
| DrawWall | מדפיסה חומה מלבנית |  | חומה מלבנית |
| CheckLose | בודקת אם צורה הגיעה לחלק העליון של המסך ויוצאת מהמשחק אם אכן צורה הגיעה. |  |  |
| CheckRows | סורקת את המסך לבדוק האם יש שורות שהושלמו. אם שורה הושלמה, הפרוצדורה קוראת לDeleteRow כדי למחוק את השורה |  |  |
| DeleteRow | מוחקת את השורה שהתמלאה, מורידה את השורות שמעל השורה המלאה למטה, וקוראת לSound שמשמיעה צליל |  | מחיקת שורה, הזזת השורות שמעליה והשמעת צליל |
| Randomize | מכניסה לרגיסטר DX מספר פסאודו־אקראי |  |  |
| Sound | משמיעה תו נתון באמצעות הרמקול של המחשב | תו | השמעת צליל |
| DrawNext | מדפיסה את הצורה הבאה שתצא בתוך ריבוע | הצורה הבאה והצבע הבא | ריבוע ובתוכו הצורה הבאה |
| ShowScore | מדפיס את נתוני המשחק (רמה, מספר שורות שהתמלאו וניקוד)  הפרוצדורה משתמשת בPrintNum כדי להפוך את הנתונים המספריים למערך בתים ולהדפיסם |  |  |
| BitPlacer | יוצרת מספר עם ביט אחד ששווה לאחד במקום מסויים כדי לשוות עם הצורה המיוצגת בעזרת מספר ולבדוק האם יש התנגשות | מיקום הביט | מספר עם ביט דולק במקום מסויים |
| CheckColor | בודקת את הצבע במקומות שצריך לבדוק וכך בודוקת האם יש פגיעה עם צורה אחרת. הפרוצדורה משתמשת בBitPlacer כדי לבדוק באיזה נקודות צריך לבדוק | צורה המיוצגת על ידי מספר | משתנה check ששווה לאחד אם יש פגיעה ושווה לאפס אם אין |
| CheckFlip | בודקת לפני היפוך הצורה האם הצורה ההפוכה עולה על צורה אחרת ומשנה את משתנה canFlip בהתאם | צורה הפוכה המיוצגת על ידי מספר | משתנה canFlip ששווה לאחד אם יש פגיעה ושווה לאפס אם אין |
| Flip | מקבלת מספר המייצג את הצורה הקיימת ומבצעת היפוך "מטריצה" ובכך מסובבת את הצורה ימינה. הפרוצדורה משתמשת בFlipFix כדי להחזיר את הצורה לאותו הציר ובכך לא לשנות את המיקום שלה בעת הסיבוב | צורה המיוצגת על ידי מספר | צורה הפוכה המיוצגת על ידי מספר |
| FlipFix | דוחפת את הביטים של הצורה ימינה עד שהצורה נמצאת על הציר השמאלי למטה | צורה הפוכה מיוצגת על ידי מספר | צורה הפוכה מתוקנת מיוצגת על ידי מספר |
| Draw | מקבלת מיקום, צבע, אורך ורוחב ומדפיסה מלבן עם התכונות האלה | מיקום המלבן, צבע המלבן, אורך ורוחב המלבן | מלבן מודפס על המסך |
| DrawBlock | מדפיסה ריבוע עם מסגרת וגרפיקה בצבע נתון במיקום נתון. משתמשת בDraw כדי להדפיס את הריבוע | מיקום הריבוע, צבע הריבוע | ריבוע מודפס על המסך |
| DrawPiece | הפרוצדורה מקבלת מספר שמייצג את הצורה. הפרוצדור עוברת על הביטים של המספר ומתייחסת עליו כמטריצה. לכל ביט, אם הוא דלוק הפרוצדורה מדפיסה ריבוע בעזרת DrawBlock ואם הוא כבוי היא ממשיכה הלאה | צורה המיוצגת על ידי מספר, מיקום הצורה, צבע הצורה | צורה מודפסת במיקום המתאים על המסך |
| UnDraw | משנה את משתנה הצבע לשחור וקוראת לDrawPiece ובכך מוחקת את הצורה | צורה המיוצגת על ידי מספר, מיקום הצורה | מוחקת את הצורה במיקום המתאים על המסך |
| EditColor | מכניסה לתוך פלטת הצבעים של המחשב את ערך RGB במערך הצבעים במיקום מסויים | מערך הצבעים RGB, מיקום הצבע | פלטת צבעים מתוקנת |
| EditPallette | מכניסה את כל ערכי הRGB במערך הצבעים לתוך פלטת הצבעים של המחשב. משתמשת ב EditColor להכנסת הצבעים. | מערך צבעים RGB | פלטת צבעים מתוקנת |
| Text\_Printdec | מדפיסה מערך שמייצג מספר דצימלי | מערך שמייצג מספר דצימלי | הדפסת המערך |
| HEX2DEC | ממירה מספר מהקסדצימלי ודצימלי | מספר הקסדצילי | מספר דצימלי |
| OpenFile | פותחת קובץ txt | שם הקובץ, סיבת הפתיחה(קריאה/כתיבה) |  |
| ReadFile | קוראת מקובץ txt ומכניסה מספר בתים מהקובץ לתוך מערך | מספר בתים מהקובץ | מערך עם הבתים מהקובץ |
| WriteToFile | כותבת לתוך הקובץ ממערך נתון | מערך בתים | קובץ עם הבתים מהמערך |
| CloseFile | סוגרת את הקובץ |  |  |
| PrintHighScore | משתמשת בOpenFile ובReadFile כדי להכניס לתוך מערך את הנתונים בקובץ, ומדפיסה את הקובץ בצורה מסויימת על המסך. משתמשת בנוסף בPrintNum כדי להפוך את הנתונים המספריים במערך לבתים ולהדפיסם | קובץ txt | הנתונים בקובץ |
| AddHighScore | משתמשת בOpenFile ובReadFile כדי להכניס לתוך מערך את הנתונים בקובץ. הפרוצדורה מקבלת ניקוד ובודקת האם הוא יותר גדול מאחד הניקודם במערך. אם הוא יותר גדול, היא מקבלת את השם של השחקן מכניסה אותו ואת הניקוד לתוך המערך במיקום המתאים ומשתמשת ב CloseFile כדי לסגור את הקובץ |  |  |
| PushList | דוחפת את הניקוד הקטן מהניקוד הנתון למטה ומוחקת את הניקוד הנמוך ביותר. בכך היא מאפשרת לניקוד הנתון להיכנס למקום המתאים לו בלוח הניקוד | ניקוד | ניקוד מתוקן |
| PrintNum | משתמשת ב HEX2DEC וב Text\_Printdec כדי לקחת מספר ולהפוך אותו למערך בתים שמוכן להדפסה | מספר | מערך בתים |
| EnterLevel | מקבלת מהמשתמש את הרמה ההתחלתית בה הוא רוצה להתחיל את המשחק ומכניסה את הקלט לתוך המשתנה level | רמה התחלתית | המשתנה level השווה לקלט |
| CheckLevel | בודקת האם המשתמש עבר שלב ואם כן קוראת ל AdjustSpeedכדי למהר את מהירות המשחק |  |  |
| AdjustSpeed | מכפילה את משתנה המהירות delayT ב0.8 | המשתנה delayT | המשתנה delayT המתוקן |

**מקורות חיצוניים**

**אתר הסייבר הישראלי** cyber.org.il

**פורום StackOverflow** stackoverflow.com

**פורום Piazza** piazza.com

**בעיות ופתרונות**

בפרוייקט נתקלתי במספר בעיות, כגון היפוך הצורה ועבודה עם קבצי טקסט.

כדי לפתור את בעיות אלו נעזרתי בחברי לכיתה ובמנחים, וכמו כן באתרי פורומים כמו **StackOverflow** ובספר **gvahim\_assembly\_book**.

**בנימה אישית**

אני מאוד נהניתי לבנות את הפרויקט. בהתחלה לא הייתי בטוח איזה אלגוריתם אני רוצה לכתוב, והיו לי רעיונות שונים לפרויקט. אך בסופו של דבר התחלתי לעבוד על טטריס – הפרויקט הנוכחי, וכל פעם שיפרתי אותו יותר וחשבתי מה אני יכול להוסיף. במבט לאחור אני נזכר כשהמשחק היה רק קובייה שנופלת לנצח מהמסך ואני מתרגש לראות לאן הפרויקט הגיע בסופו של דבר.

הפרויקט לא היה שווה חצי ממה שהוא יצא מבלי ערן המנחה, שתמך בי וכמובן נתן לי את הידע המקדים הדרוש לבניית הפרויקט, אורי תומך ההוראה, שליווה אותי לכל אורך הדרך ועזר לי, וכמובן חברי לכיתה ששיתפו אותי עם הרעיונות שלהם.