**תיק פרויקט**

**Zombie Apocalypse**

מתכנת: יובל מנדל

שפת תכנות: אסמבלי 8086

תעודת זהות: 216496349

מורה מלווה: אמיל אברמוביץ’

כיתה: י'3

בית ספר: (ע"ש) הרצוג כפר סבא

תוכן עניינים

[מבוא 3](#_Toc136980752)

[נושא העבודה 5](#_Toc136980753)

[אופן הפעלה 6](#_Toc136980754)

[גרסאות המערכת 7](#_Toc136980755)

[תיעוד והסבר הפתרון 8](#_Toc136980756)

[רשימת פעולות 10](#_Toc136980757)

[תרשימי זרימה 14](#_Toc136980758)

[פרטים המאפשרים כניסה והפעלה 20](#_Toc136980759)

[דוגמאות הרצה 21](#_Toc136980760)

[סיכום אישי 22](#_Toc136980761)

# מבוא

שם העבודה: Zombie Apocalypse

שם הקובץ: “game.asm”

קבצים נלווים שבהם נעשה שימוש בעבודה

“BackGRND.bmp”, “HS.txt”, “KnightDe.bmp”, “KWalk1.bmp”, “KWalk2.bmp”, “KRoll1.bmp”, “KRoll2.bmp”, “KRoll3.bmp”, "Menu.bmp”, “ZMBWL1.bmp”, “ZMBWL2.bmp", "ZMBWR1.bmp”, “ZMBWR2.bmp”, “GameOver.bmp”

הסבר:

מסכים –

“Menu.bmp”, “GameOver.bmp”, “BackGRND.bmp”

תמונות של זומבים –

“ZMBWL1.bmp”, “ZMBWL2.bmp", "ZMBWR1.bmp”, “ZMBWR2.bmp”

תמונות של הדמות הראשית –

“KnightDe.bmp”, “KWalk1.bmp”, “KWalk2.bmp”, “KRoll1.bmp”, “KRoll2.bmp”, “KRoll3.bmp”

תוצאה הכי גבוהה –

“HS.txt”

סביבת עבודה

כדי ליצור קובץ קובץ הרצה, צריך לעשות לקובץ אסמבל (הופך את הקובץ ל- obj) ואז לינק (הופך את הקובץ לקובץ הרצה exe) כדי לעשות זאת השתמשתי בסביבת העבודה “Turbo assembler” ובשימוש בקבצים tasm (אסמבל) ו- tlink (לינק).

אסמבל:

Tasm /zi game.asm

לינק:

Tlink /v game.obj

סביבת פיתוח

סביבת הפיתוח בה השתמשתי היא “NotePad++”, סביבה זו מאפשרת כתיבה והתמצאות נוחה. הסביבה מכירה את שפת התכנות אסמבלי ולכן מדגישה פעולות/אלמנטים שקשורים לשפה.

בנוסף, השתמשתי בדיבאגר Turbo Debugger, כדי לעקוב ולדאבג את התוכנית שלי. כדי להריץ את התוכנית בדיבאגר צריך לכתוב את השורה הבאה בסביבת ההרצה:

Td game.exe

הדיבאגר מציג את הזכרון את הרגיסטרים וגם את המסך של המשתמש כך שניתן להבחין בקלות איזה רכיב גורם לבאג.

סביבת הרצה

סביבת ההרצה בה השתמשתי היא DOSBox. DOSBox הוא אמולטור למערכת ההפעלה DOS ומדמה את המעבד 8086 מה שגורם לה לזהות את שפת אסמבלי 8086 ולהריץ את קבצי ההרצה, דבר שלא אפשרי בWindows.

# נושא העבודה

המשחק שיצרתי הוא משחק מסוג Top Down Shoot, כלומר נקודת המבט של השחקן היא מלמעלה.

במשחק שלי השחקן הוא אביר שנלחם במספר זומבים כאשר המטרה היא לשרוד כמה שיותר זמן תוך כדי חיסול הזומבים וצבירת נקודות (על כל זומבי שנהרג השחקן מקבל נקודה). הוא יכול לזוז ל- 8 כיוונים ויכול לעשות Dash לכיוון אליו הוא הולך (במהלך ה- Dash השחקן לא פגיע). האביר יכול גם לירות יריות לכל נקודה שהוא בוחר. במהלך המשחק נוצרים זומבים מ- 4 מקומות במסך שהופכים להיות יותר מהירים ככל שהמשחק מתקדם. זומבים מורידים חיים לשחקן כאשר הם נוגעים בו והשחקן יכול להרוג זומבים בכך שהוא פוגע בהם יריות. המשחק נגמר כאשר השחקן נהרג. אם השחקן שבר את שיא הנקודות אז הניקוד שלו נשמר בתור השיא החדש.

# אופן הפעלה

כדי להפעילה את המשחק השחקן נדרש להריץ את סביבת ההרצה DOSBox ושם לעשות mount למיקום של קבצי המשחק ואז לרשום את שם קובץ ההרצה Game.exe וללחוץ אנטר.

תחילה, השחקן מוצב מול מסך פתיחה (שם גם מופיע שיא הנקודות הנוכחי), כדי להתקדם על השחקן ללחוץ אנטר ואז המשחק מתחיל. האביר יכול לנוע ל- 8 כיוונים באמצעות הכפתורים wasd (כל לחצן הוא כיוון וכל שילב של לחצנים לא נגדיים הוא כיוון). כדי באופן רגעי לנוע במהירות גבוהה יותר ולהיות בלתי פגיע על השחקן ללחוץ רווח (Dash) (השחקן מוותר על היכולת לירות ולשנות כיוון בזמן הזה). כדי לשחרר יריה על השחקן ללחוץ מקש שמאלי והירייה תתקדם לכיוון הסמן של העכבר. כמות החיים של השחקן מופיעה בפינה השמאלית למטה וכמות הנקודות של השחקן מופיעות בצד ימין למטה. כדי לצאת מהמשחק השחקן יכול למות או ללחוץ esc. כשהשחקן יוצא מהמשחק יופיע לו מסך סיום עם מספר הנקודות הסופי שלו ואם הוא שבר את שיא הנקודות זה יצוין במסך. כדי לצאת מהתוכנית על השחקן ללחוץ אנטר והשליטה תחזור ל- DOSBox.

# גרסאות המערכת

בגרסה זו של המשחק הדרך היחידה של המשחק להתקיף זה באמצעות יריות. הייתי רוצה שבגרסאות אחרות של המשחק לשחקן יהיה גם אפשרות להשתמש בחרב (מכה מקרוב). בנוסף אפקטים של שמע בזמן יריות או כזומבים פוגעים בשחקן זה אלמנט שיכול להוסיף הרבה לחוויית המשחק של השחקן. משהו נוסף שהייתי מוסיף למשחק זה עוד פונקציות שהשחקן יכול להשיג כלל שהוא הורג זומבים כמו יריות מקפיאות שעוצרות את הזומבים או יריות חשמל שקופצות מזומבי לזומבי.

# תיעוד והסבר הפתרון

התוכנית שלי מחזיקה נתונים על האביר בזיכרון על ידי משתנה ייעודי על כל תכונה. לכל משתנה יש שם וכדי לשנות את התכונות של האביר אני צריך לפנות למשתנה עם האופסט הספציפי שלו. לעומת זאת, נתונים על הזומבים והירויות שמורים בצורה שונה. יש מעיין calss ליריות ולזומבים. אני מניח שכאשר אני פונה לאופסט שבו אני יודע שיש זומבי או יריה הנתונים שלו מסודרים בסדר מסוים אחרי האופסט הזה ולכן אני יכול להכין פעולות שעובדות על זומבים ויריות באופן כללי בלי ליצור פעולה לכול יריה וזומבי.  
לדוגמה באופסט של הזומבי יש בוליאני שמציין האם הוא פעיל ואז באופסט שלו +1 יש את ערך ה- x ובאופסט + 3 יש את ערך ה- y שלו אז בפעולות שבהן אני מזיז זומבי אני אדע לשנות את האופסט +1 ו- +3, או אם הזומבי מת אז אני אשים 1 באופסט של הזומבי.

כדי שאוכל לבטא מספרים בצורה מדויקת אני מתייחס לכל מספר שקשור למיקום של משהו על המסך או לאורך של משהו או המהירות שלו כמספר בעל נקודה עשרונית סטטית (Fixed Decimal Point), זה אומר ש- 4 הביטים הראשונים של המספר מייצגים שבר, מה שמקטין את המספר הכי גדול שאפשר לייצג אך נותן לייצג מספרים עם שברים (רמת הדיוק בשבר היא עד 1/16 כלומר השברים יתעגלו).

כדי שהיריות יעברו על קו ישר לסמן של העכבר ויהיו באותה המהירות, והזומבים ילכו לאביר בקו ישר ובאותה המהירות אני חישבתי את וקטור היחידה בכיוון במתאים. עשיתי זאת סדרך הבאה:

נסמן a הפרש ה- y של שני הנקודות, נסמן b הפרש ה- x של שני הנקודות.

ערך המישור של וקטור היחידה ( x שאני מוסיף כל פעם לערך ה- x של היריה/זומבי) =

b/sqrt(b^2+a^2)

ערך האנך של וקטור היחידה ( y שאני מוסיף כל פעם לערך ה- y של היריה/זומבי) =

a/sqrt(b^2+a^2)

בתוכנית שלי בחרתי לשנות את צורת העכבר כפי שנראית על המסך. מראה עכבר עובד בצורה בינארית, כל ביט מייצג פיקסל על המסך. בכדי לשנות את צורת העכבר צריך ליצור שתי "מטריצות" אחת אחרי השנייה (כל אחת היא 16 וורדים – לכן זה כמו לומר 16\*16). המטריצה הראשונה תסמן את צורת העכבר המבוקשת, המטריצה השנייה תסמן איפה צריך לשים צבע.

כדי להוסיף תמונת רקע פעולות המחיקה של היריות אביר וזומבים הן פעולות שפותחות את ה- bmp של הרקע וקוראות את ה- Header וה- Pallete של קובץ ה- bmp. אז, הן מדלגות לשורה הרלוונטית שממנה צריך להעתיק חלק מה- bmp ומשם הן מעתיקות רק את החלק הרלוונטי מה- bmp לרקע כדי שלא יהיה צורך להעתיק את כל ה- bmp למסך כל פעם שמשהו זז במסך.

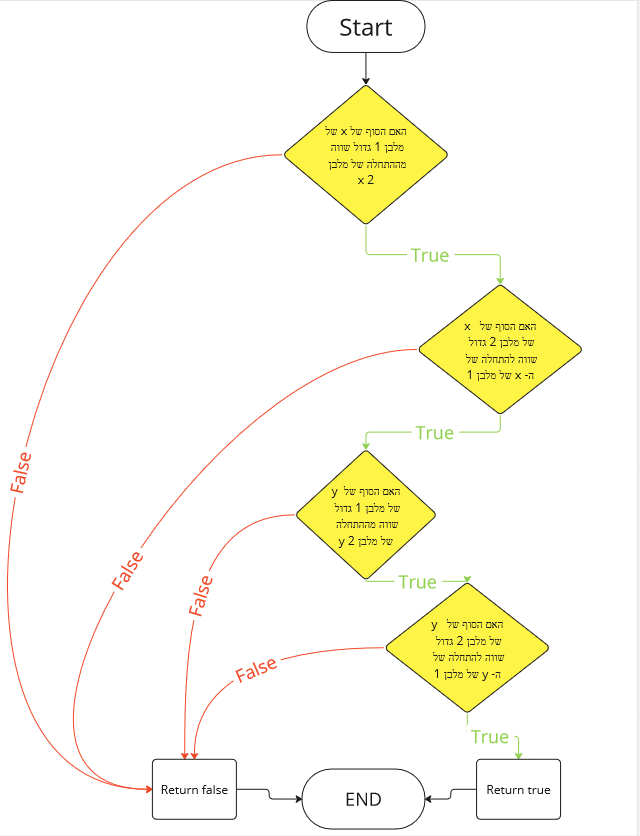
כל bmp בהן אני משתמש בלולאה הראשית (חוץ מהרקע) מומרות למטריצות בתחילת הקוד והצגה שלהן מתבצעת באמצאות המטריצה.

# רשימת פעולות

|  |  |  |  |
| --- | --- | --- | --- |
| **שם פעולה** | **קלט** | **פלט** | **תיאור** |
| DelayTheSameAmountEachCycle | משתנים: "ActiveZombieCounter"  "ActiveBulletCounter" | אין | החלפה של הרקע גורמת לדיליי אז כדי שתמיד יהיה אותו הדיליי הפעולה גורמת לדיליי לפי כמות בזומבים והיריות הפעילות |
| DisplayHighScore | משתנה:  "HighScore" | המסך | כותב את התוצאה הכי גבוה בתחתית המסך |
| GetHighScore | קובץ: HS.txt | משתנה: HighScore | שם את התוצאה הכי טובה מהקובץ למשתנה |
| CheckHighScore | משתנים:  HighScore, PlayerScore | קובץ:  HS.txt | מעדכן את הקובץ עם התוצאה הכי גדולה אם התוצאה הנוכחית יותר גבוהה ממנה |
| PutBackgroundInSpot | קובץ: BackGRND.bmp  Stack: 1.X, Y, Len, Width | מסך | שם את הקטע הרלוונטי מתמונת הרקע על המסך |
| PrintBackground | קובץ: BackGRND.bmp | מסך | מצייר את תמונת הרקע על המסך |
| SetMouseShape | מטריצה: MouseMask | צורה של עכבר | משנה את צורת הסמן של העבר לפי MouseMask |
| UpdateRoll | משתנים:  KRolling  KCanRoll | משתנים:  KNeedDraw  KRollStage  KRolling  KAbleToBeHit  KCanMove  RollCoolDownCounter  KCanRoll | מעדכן את ה- cooldown/stage של הדאש (תלוי במצב של הדאש) |
| AddXYPlayerRoll | KXToAddWhileRolling  KYToAddWhileRolling | YPlayer  XPlayer | מוסיף את ה- x y בשביל הדאש |
| StartRoll | דגלים של מקשים + KRolling  KCanRoll | KRollStage  KAbleToBeHit  KCanMove  KRolling  KYToAddWhileRolling  KXToAddWhileRolling | מתחיל את הדאש |
| ActivateZombiesRandomly | אין | אין | קורא לפעולה שמפעילה זומבים - בסיכוי כלשהו |
| ShowScore | משתנה: PlayerScore | מסך | מראה את הניקוד הנוכחי בצד ימין למטה של המסך |
| DisplayFinalScore | משתנה: PlayerScore | מסך | מראה את הניקוד הסופי |
| FindAndActivate | משתנה: ZombieOffsetArray | אין | מוצא זומבי לא פעיל במערך זומבים וקורא ופעולה שמפעילה אותו |
| DisplayMainMenu | קובץ: Menu.bmp | מסך | מדפיס את הMenu על המסך |
| DisplayGameOver | קובץ: GameOver.bmp | מסך | מדפיס את המסך סגירה על המסך |
| WaitForEnter | אין | אין | עוצר את התוכנית על ללחיצה על מקש אנטר |
| ClearScreen | אין | מסך | מוחק את המסך |
| DisplayKHP | משתנה: KnightHP | מסך | מציג את החיים של השחקן בצד שמאל למטה |
| DrawZombie | Stack: offsetzombie | מסך | מדפיס זומבי |
| SetZombieBehavior | Stack: offsetzombie  XPlayer  YPlayer | תכונות של זומבי | קובע את הכיוון של הזומבי אל איפה שהשחקן נמצא עכשיו |
| UndrawZombie | Stack: offset zombie | מסך | מוחק את הזומבי |
| CheckandUpdateallZombies | ZombieOffsetArray | אין | בודק האם כל זומבי פעיל וקורא לפעולה שתעדכן אותו עם כן |
| UpdateZombie | Stack: offset zombie | תכונות של הזומבי | מעדכן את כל התכונות של הזומבי (x, y, alive) מעדכן התנהגות בודק האם זומבי מתנגש באביר מעדכן לו את ה- cooldown |
| ZMBMoveBack | Stack: offset zombie | תכונות של הזומבי | גורם לזומבי ללכת צעד אחורה |
| MoveZombie | Stack: offset zombie | תכונות של זומבי | מוזיז את הזומבי (משנה לו את ה- xy ) |
| ActivateZmbRnd | Stack: offset zombie | תכונות של זומבי | מפעיל זומבי ואחד מ- 4 מקודות במסך וקובע לו התנהגות (מאתחל אותו) |
| UpdateShootCoolDown | משתנים:  KShootCoolDown  KCanShoot | משתנים: KShootCoolDown  KCanShoot | מעדכן את הcooldown על היריות של האביר |
| ShowCurser | אין | סמן עכבר | מראה את הסמן של העכבר |
| HideCurser | אין | במן עכבר | מסתיר את הסמן של העכבר |
| SetAsyncMouse | אין | טבלת וקטורים של הפסיקות בתחילת הזיכרון | גורם ללחיצה על מקש שמאלי של העכבר לקרוא לפעולה MouseHandle |
| MouseHandle | KCanShoot  KRolling  BulletOffsetArray  XPlayer  YPlayer | Bullet | אם השחקן יכול לירות יאתחל ירייה |
| DrawBullet | Stack: Bullet offset | מסך | מדפיס ירייה |
| UndrawBullet | Stack: offset bullet | מסך | מוחק ירייה |
| Update\_activated\_Bullets | BulletOffsetArray | Bullet | עובר על מערך יריות ומפעיל אחת |
| MoveBullet | Stack: offset bullet | תכונות Bullet | מוזיז את הירייה ובודק האם הוא יצא מהזירה |
| CheckBulletHitZombies | משתנים: ZombieOffsetArray  Stack: offset Bullet | תכונות Bullet וזומבי | בודק האם יריה ספציפית פוגעת בזומבי כלשהו והאם כן מעדכנת אותם |
| KnightShotZombie | Stack: offset Bullet, offset Zombie | משתנים של ירייה ושל זומבי Player Score | פועלה שמקבלת זומבי ויריה ומעדכנת אותם (נקרא כאשר היריה פוגעת בזומבי) |
| CheckCollision | Stack: x1 y1 len1 height x2 y2 len2 height2 | Stack: bool | בודק האם שני מלבנים פוגעים אחד בשני |
| ActivateBullet | Stack: 1.xstart 2.ystart 3.xtarget 4.Ytarget 5.bullet | תכונות של ירייה | מאתחל יריה (מפעיל אותה) |
| LoopDelay | אין | אין | עודה דיליי לתוכנית |
| printAxDec | ax | screen | מדפיס את המספר ב- ax על המסך |
| SetAsyncKeyboard | Keyboard\_handler | Oldintruptoffset  Oldintruptsegment  VectorInterupt table | הופך את המקלדת לאסינכורנית |
| RestoreKeyboardInt | Oldintruptoffset  Oldintruptsegment | Vector Interupt table | מחזיר את המקלדת ללא אסינכרונית |
| Keyboard\_handler | מקלדת | דגלי מקשים | מדליק ומכבה את דגלי המקשים |
| KMoveUp | אין | YPlayer | האביר זז למעלה |
| KMoveDown | אין | YPlayer | האביר זז למטה |
| KMoveRight | אין | XPlayer | האביר זז ימינה |
| KMoveLeft | אין | XPlayer | האביר זז שמאלה |
| CheckKeys | דגלי מקשים | מסך, תכונות אביר | בודק את הדגלים של הקשים |
| DrawKnight | KRolling  FrameNumber | מסך | מדפיס אביר למסך |
| UndrawKnight | LastXPlayer  LastYPlayer | מסך | מוחק את האביר מהמסך |
| RemoveFixedDecimalPoint | Stack: num | Stack: num | מוריד את הנקודה העשרונית הסטטית |
| XYToMemory | Stack: 1.x 2.y | Stack: 1.memory | מעביר ערך x y למיקום במסך |
| Sqrt | Stack: 1.HighOrder 2.LowOrder | 1.Answer | שורש למספר 32 ביט |
| XYtoAdd2Dots | Stack: 1.x1 2.y1 3.x2 4.y2 5.speed | Stack: 1.VectorX 2.VectorY | מוצא את וקטור היחידה בין 2 הנקודות ומכפיל אותו במהירות |
| XYtoAdd2DotsWithNeg | Stack: 1.x1 2.y1 3.x2 4.y2 5.speed | Stack: 1.VectorX 2.VectorY | פעולה עוטפת ל - XYtoAdd2Dots גורם לזה לעבוד כשה- y או ה- x של המטרה קטנים |
| ConvertBMPtoMatrix | Stack: 1.Bmp Offset 2. BMP len 3. BMP Height 4. Matrix offset | Matrix | ממיר bmp ל- מטריצה |
| ConvertAllBMPToMatrix | אין | כל המטריצות בזיכרון | ממיר את כל המטריצות בזיכרון |
| RandomByCs | Bl- מינימום bh- מקסימום | al | מגריל מספר רנדומלי בין bl ל- bh |
| MakeMask | Bh- גודל מסכה | si | מכין מסכה של 1 לפי bh |
| RandomByCsWord | BX- מינימום DX- מקסימום | AX | מגריל מספר בין bx ו- dx |
| MakeMaskWord | dx- גודל מסכה | si | מכין מסכה של 1 לפי הגודל של dx |
| OpenShowBmp | dx- bmp offset  OneBmpLine ScrLine FileHandle Header Palette ErrorFile BmpLeft BmpTop BmpColSize BmpRowSize | מסך | מציג את ה- bmp על המסך |
| OpenBmpFile | dx- offset bmp file name | FileHandle | פותח קובץ bmp |
| CloseBmpFile | FileHandle | אין | סוגר קובץ bmp |
| ReadBmpHeader | FileHandle | Header | קורא את הHeader של קובץ bmp ל- Header |
| ReadBmpPalette | FileHandle | Palette | קורא את הפלטה של קובץ bmp Palette |
| CopyBmpPalette | Palette | פלטה של DOSBox | מעתיק את הפלטה לדוסבוקס |
| ShowBMP | BmpLeft BmpTop BmpColSize BmpRowSize ScrLine | מסך | מציג את המטריצה של ה- BMP על המסך |
| SetGraphic | אין | מסך | הופך את DOSBox למצב גרפי |
| putMatrixInScreen | Matrix – offset matrix  dx- length cx – height di – Location to print | מסך | מציג מטריצה על המסך |
| OpenMyBmpFile | dx- offset bmp file name | MyFileHandle  מסך | פותח קובץ bmp |
| CloseMyBmpFile | MyFileHandle | אין | סוגר קובץ bmp |
| ReadMyBmpHeader | MyFileHandle | MyHeader | קורא את הHeader של קובץ bmp ל- MyHeader |
| ReadMyBmpPalette | MyFileHandle | MyPalette | קורא את הפלטה של קובץ bmp Palette |
| CopyMyBmpPalette | MyPalette | פלטה של DOSBox | מעתיק את הפלטה לדוסבוקס |

# תרשימי זרימה

CheckCollision:

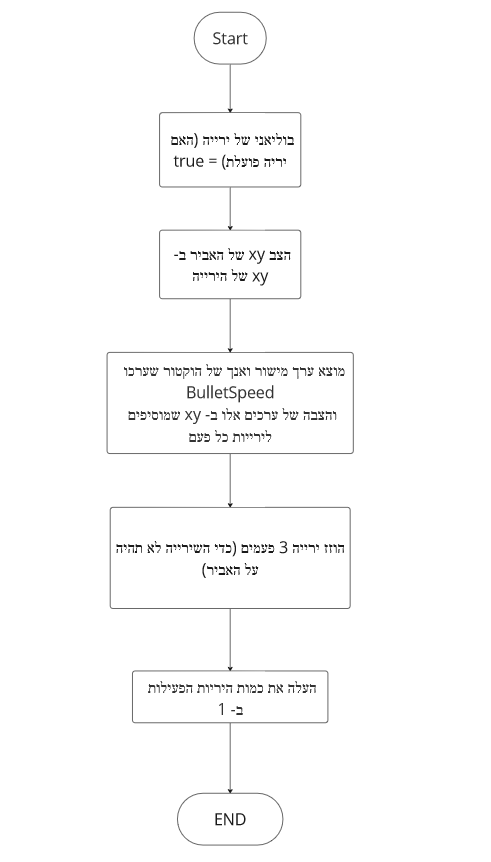


UpdateZombie:

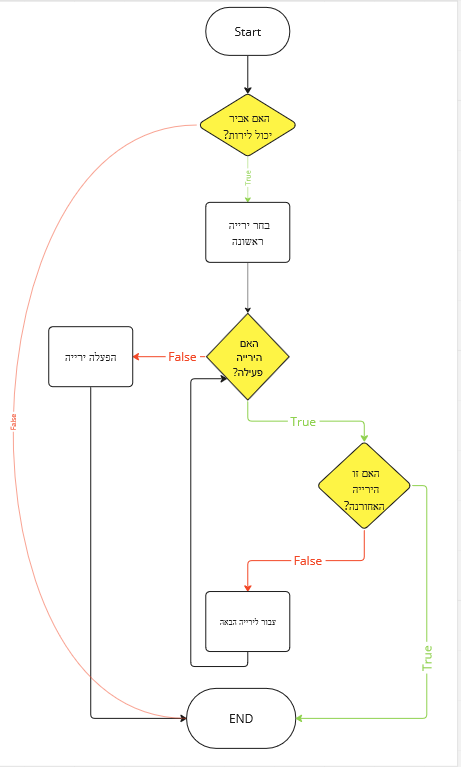
תמונה שמכילה טקסט, תרשים, תוכנית, קו

התיאור נוצר באופן אוטומטי

ActivateBullet:



MouseHandle:



Main Loop: תמונה שמכילה טקסט, תרשים, צילום מסך, קו

התיאור נוצר באופן אוטומטי

Update Roll:

תמונה שמכילה טקסט, תרשים, צילום מסך, קו

התיאור נוצר באופן אוטומטי

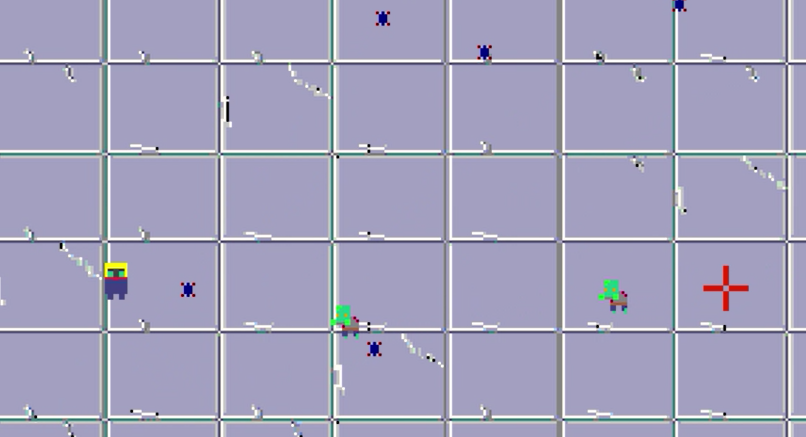
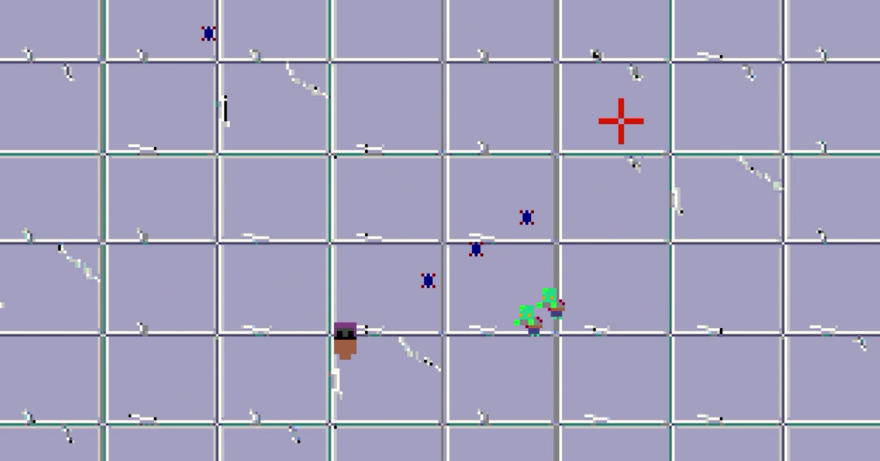
# פרטים המאפשרים כניסה והפעלה

כדי להריץ את המשחק יש לכתוב ב- DOSBox את שם הקובץ game.exe כשדוסבוקס נמצא על 20000 cycles (פקודה: (cycles = 20000. כדי לצאת ממסך הפתיחה יש ללחוץ אנטר וכדי לצאת למסך הסיום יש ללחוץ esc, כשהמשתמש נמצא במסך סיום עליו ללחוץ אנטר כדי להחזיר את השליטה לדוסבוקס.

# תמונה שמכילה טקסט, גופן, צילום מסך, עיצוב גרפי התיאור נוצר באופן אוטומטידוגמאות הרצה

תמונה שמכילה צילום מסך, קו, מקביל, ריבוע

התיאור נוצר באופן אוטומטי

 תמונה שמכילה טקסט, גופן, צילום מסך, גרפיקה

התיאור נוצר באופן אוטומטי תמונה שמכילה צילום מסך, קו, ריבוע, מקביל

התיאור נוצר באופן אוטומטי

# סיכום אישי

תכנון וכתיבת פרויקט זה גרמו לי לפתח את יכולת התכנות שלי וללמוד איך לכתוב תוכנית קריאה שאוכל להבין בהמשך. לאורך כל הכנת הפרויקט הייתי צריך לדבג את הפעולות שלי. כדי לעשות זאת למדתי איך להשתמש בדיבאגר בצורה אפקטיבית, כלי שתורם לי גם במשימות אחרות שאני מבצע.

כשהתחלתי לתכנן את הפרויקט שלי אני חשבתי שאני לא אצליח לעשות פרויקט שנראה ומתפקד טוב. זו הפעם הראשונה שאני כותב פרויקט בסדר גודל הזה, ותוכנית שרצה ועושה פעולות מסובכות ברקע נראתה לי מעבר ליכולת שלי. אחרי שהתחלתי לעבוד, ראיתי שאם אני עובד כל פעם על אלמנט אחר זה בסוף מתחבר, ומצטבר לתוכנית גדולה, שעוברת את הציפיות שהיו לי.

לסיכום, הפרויקט נתן לי מוטיבציה לעבוד קשה ושינה איך שאני מתכנן, מתעד, כותב ומתקן תוכניות.