



**תיק פרויקט**

**שם הפרויקט:** Super Mario Bros.

**שם המתכנת:**

**תעודת זהות:**

**מורה מלווה:**

**כיתה:**

**בית ספר:**

**תוכנן עניינים**

מבוא...........................................................................................................עמוד 2

נושא העבודה............................................................................................עמוד 3

אופן ההפעלה............................................................................................עמוד 4

גרסאות המערכת......................................................................................עמוד 5

תיעוד והסבר הפתרון...............................................................................עמוד 6

תרשימי זרימה...........................................................................................עמוד 8

רשימת הפעולות.......................................................................................עמוד 9

פרטים המאפשרים כניסה והפעלה....................................................עמוד 15

קוד התוכנית...........................................................................................עמוד 15

דוגמאות הרצה.......................................................................................עמוד 16

סיכום אישי..............................................................................................עמוד 17

**מבוא**

**שם העבודה:** סופר מריו ברוס.

**שם הקובץ:** mario.exe

**סביבת העבודה:** במקום בו הקובץ mario.asm קיים יש לבצע דרך סביבת ההרצה את הפקודות הבאות על פי הסדר:

1. tasm /zi mario

2. tlink /v mario

**סביבת הרצה:** DosBox

**סביבת הפיתוח:** Visual Studio Code, Paint.net, Microsoft Paint

**קבצים נלווים:**

|  |  |
| --- | --- |
| castle.bmp | black.bmp |
| flag.bmp | coin.bmp |
| gmovwin.bmp | gmovlose.bmp |
| goomba.bmp | goodbye.bmp |
| loadscr.bmp | lives.bmp |
| pole.bmp | opening.bmp |
| skynum.bmp | poletop.bmp |
| smlft.bmp | skystats.bmp |
| wrld1.bmp | smrgt.bmp |

**נושא העבודה**

במהלך השנה נלמדה השפה אסמבלי ויכולותיה.

באמצעות הידע מהלמידה, שילבתי בפרויקט גרפיקה מתקדמת, אלגוריתמיקה, וקלט-פלט.

העבודה מתבססת על המשחק הקלאסי משנות ה-80 של המאה ה-20, סופר מריו ברוס, הזכה להצלחה רבה ברחבי העולם.

בעבודה ניתן לשחק בעולם הראשון בלבד, אך הוא כולל מספר אפשרויות מהמשחק המקורי, כגון האנימציות העשירות, מטבעות, נקודות, עם ספירה לאחור ואף עולם שלם לחקור.

ניתן להסתובב בעולם במטרה להגיע לדגל ולהורידו כלפי מטה, לאסוף מטבעות, אך צריך להיזהר כי ניתן ליפול אל תוך תהום.

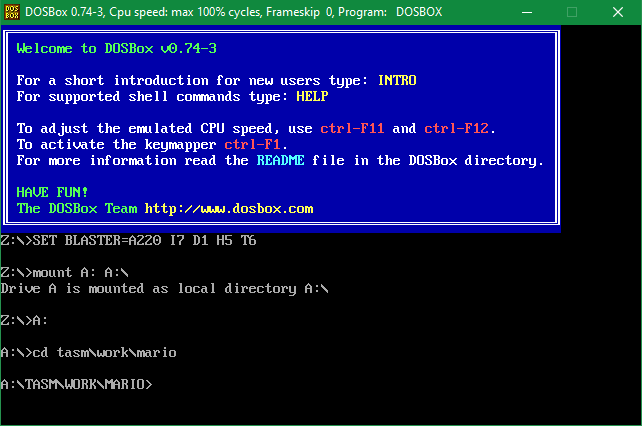
**אופן ההפעלה**

לצורך הפעלת המשחק יש לעקוב אחר השלבים הבאים:

0. מומלץ להגדיר cpu cycles=max בDosBox Options

1. הרצת התוכנה DosBox

2. הגדרת סביבת העבודה

3. הרצת התוכנית mario.exe

לאחר מכן יופיע מסך פתיחה, בו ניתן לצאת מהמשחק באמצעות לחיצה על מקש המקלדת “Q”, או להתחיל את המשחק באמצעות הקשה על “Enter”. ניתן לזוז באמצעות חצי המקלדת לאחר הפעלת המשחק.



**גרסאות המערכת**

בגרסה נכלל העולם הראשון של המשחק סופר מריו, עם חלק קטן אמנם מהאפשרויות, מטבעות, נקודות, שעון משחק, אנימציות מיוחדות ועולם שלם לחקור.

הגרסה עובדת עם הגרפיקה של 8086, ויש בה את האלמנטים ההכרחיים לעבוד כמשחק איכותי. ניתן לזוז, לקפוץ, ליפול, להתנגש בבלוקים ולהוציא מטבעות ואף לקפוץ על דגל הסיום, להורידו מטה ולרוץ לטירה.

מה שיוכל להיכלל בגרסאות הבאות הוא הוספת מפלצות בעלות יכולות אלגוריתמיות מיוחדות לשפר את המשחק, ואף עולם נוסף. יכולות התנועה של השחקן יכלו להיות טובות יותר, עם זיכרון תנועה ומהירות משתנה. נוסף על כך, המשחק הוא חסר צליל, מה שמהווה חלק משמעותי ביותר במריו, לכן בגרסאות הבאות צליל יכול להיות דומיננטי.

**תיעוד והסבר הפתרון**

מבנה התוכנית

התוכנית מתחילה בהצגת מסך פתיחה, ומשם נכנסת ללולאת משחק אינסופית בעלת תנאי עצירה. ברגע שאחד מתנאי העצירה מתקיים זה בסופו של דבר חוזר על עצמו בחזרה למסך הפתיחה. יציאה לחלוטין מהתוכנית מתקיימת באמצעות לחיצה על ‘Q’ במסך הפתיחה.

תיעוד והסבר הפתרון

**אלגוריתם ראשי:**

1. הצגת מסך הפתיחה וקריאת קלט מקלדת.

2. יציאה מהמשחק, הפעלתו או התעלמות והמשך הקלט.

3. לאחר הפעלה כניסה ללולאת המשחק הקוראת לפעולה Game כל עוד אין תנאי עצירה.

4. הפעולה מעדכנת את זמן המשחק שנותר, קולטת קלט מקלדת, מעדכנת את חוקי הפיזיקה של המשחק ואף מקדמת את השחקן במידת הצורך. לאחר מכן המסך מתעדכן ובסוף בודקים אם המשחק נגמר.

5. אם המשחק לא נגמר, שלבים 3-4 חוזרים על עצמם, אך אם המשחק נגמר יהיה לו קוד יציאה, אשר הוא 0 או 1.

6. אם קוד היציאה שווה ל-0, קריאה לפעולת מוות המורידה לשחקן חיים ומזהה אם הם נגמרו, במידה וכן היא קוראת לפעולת סיום משחק החוזרת לשלב 1 לאחר הצגת ההפסד. אם קוד היציאה שווה ל-1, קריאה לפעולת ניצחון המבצעת אנימציות על השחקן, הדגל והטירה. לאחר מכן מוצגת תמונת ניצחון והפעולה חוזרת לשלב 1 באלגוריתם.

הנתונים

הנתונים נמצאים במקצה הנתונים, יש מעט Constants לצורך קוד אלגנטי וקריא יותר, ורק חלק קטן מהפעולות מקבלות קלט באמצעות המחסנית. מקצה הנתונים מחולק לקטגוריות, כגון BMP, אנימציות, פיזיקה, שחקן ומערכת.

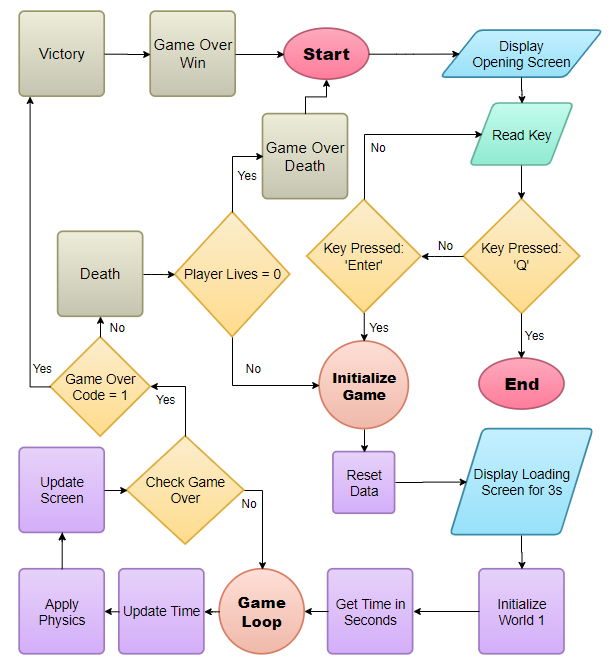
הגרפיקה

כאשר מריו מתקדם על המסך, הוא יזוז, אך כאשר הוא מגיע למרכז המסך הוא צריך להפסיק לזוז בעצמו, ובעצם כל העולם וכל העצמים זזים ביחס אליו. זה נעשה באמצעות שימוש ב-2 משתנים, אחד להתקדמות ואחד למיקום ההצגה על המסך. מיקום ההצגה על המסך משתנה בהתאם לתזוזה בפעולות התזוזה, ואם הן מזהות שהשחקן מגיע ל-160 הן לא מקדמות אותו על המסך, אלא רק את תהליך ההתקדמות שלו.

העולם הוא קובץ BMP בגודל 3584 על 169, והוא מוצג צמוד לחלק התחתון של המסך. הדרך לדעת היכן לצייר את המיקום שבקובץ היא באמצעות הגדרה של התחלת קריאה החל מנקודה מסוימת, שהיא ההתקדמות של השחקן פחות 160.

במידה והשחקן לא בגבול ה-160, זה פשוט יצייר את העולם ברקע כסטטי, והשחקן יזוז במקום.

**תרשימי זרימה**

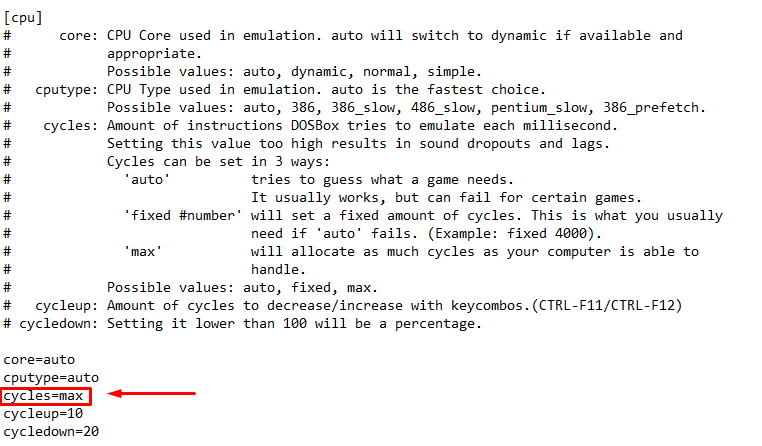


**רשימת הפעולות**

|  |  |  |  |
| --- | --- | --- | --- |
| **טענת יציאה** | **טענת כניסה** | **תיאור** | **שם הפעולה** |
| **GAME** | | | |
| אין | אין | מנהלת את כל הפעולות המרכזיות במשחק | Game |
| אין | אין | מפעילה את המשחק באמצעות איפוס של בסיס הנתונים, הצגת מסך הטעינה והפעלה של העולם הראשון (והיחיד), ולאחר מכן לוקחת את זמן התחלת המשחק בשניות לצורך האלגוריתם בפעולה UpdateTime | game\_init |
| אין | אין | מציגה את תחילת עולם 1 ושמה את השחקן בנקודת ההתחלה | init\_world1 |
| [DS:GameOver] = 0 OR 1, if 1:  [DS:GameOverCode] = 0 OR 1 | אין | הפעולה בודקת אם המשחק נגמר מהסיבות: 1. זמן נגמר 2. שחקן נפל ומת ובכך יוצאת עם GameOver=1, וקוד 0.  אם השחקן ניצח באמצעות הגעה לדגל, צא עם GameOver=1 וקוד 1  אחרת אל תשנה כלום | CheckGameOver |
| אין | אין | עוצר את לולאת המשחק, מאפס את החיים | Victory |
| [DS:PlayerLives] -= 1  OR  [DS:PlayerLives] = 3 | אין | עוצר את לולאת המשחק, מחסיר והחיים ואם הם נגמרו מגדיר סוף משחק, מציג את אנימציית המוות של מריו | Death |
| הנתונים על השחקן והמשחק מתאפסים | אין | מאפס את נתוני המשחק, אם PlayerLives שווה ל-0, משנה ל-3, אחרת משאיר את מה שהיה | ResetData |
| **SYSTEM** | | | |
| [DS:IntString] = String of AX | AX: Number to convert into a string | ממיר מספר אינטגרלי למחרוזת | int\_to\_string |
| [DS:IntString] = Digits of AX as Decimal numbers  [DS:TotalD] = Amount of Digits | AX: Number in range of 10 and 0FFFFh | ממיר את AX למספר בבסיס 10 המתבטא כספרותיו חלוקות בתוך מערך במקצה הנתונים | SplitAXDecimal |
| AX = Pixel Position | Parameters: (X, Y) | ממיר נקודה על המסך למספר הפיקסלים שנדרש להתקדם בכדי להגיע אליה | coords\_to\_pix |
| **MOVEMENT PHYSICS** | | | |
| אין | אין | מממש את חוקי הפיזיקה של המשחק ומפעיל אותם על השחקן במידת הצורך | ApplyPhysics |
| אין | אין | קורא לפעולה שמושכת את השחקן כלפי מעלה אך עם מהירות משתנה בהתאם לכוח הכבידה | UpMovementMomentum |
| אין | אין | קורא לפעולה שמושכת את השחקן כלפי מטה אך עם מהירות משתנה בהתאם לכוח הכבידה | DownMovementMomentum |
| [DS:PMovesRight] increases or resets, [DS:PlayerX] increases or remains the same, [DS:PlayerXProgress] increases, [DS:PRightForm] increases | אין | מושך את השחקן ימינה בהתאם לתנועה | ForcePlayerRight |
| [DS:PMovesLeft] increases or resets, [DS:PlayerX] decreases or remains the same, [DS:PlayerXProgress] decreases, [DS:PLeftForm] increases | אין | מושך את השחקן שמאלה בהתאם לתנועה | ForcePlayerLeft |
| [DS:PMovesUp] increases or resets, [DS:PlayerY] increases, [DS:UpVelocity] increases or resets | אין | מושך את השחקן למעלה בהתאם לתנועה | ForcePlayerUp |
| [DS:PMovesDown] increases or resets, [DS:PlayerY] decreases, [DS:DownVelocity] increases or resets | אין | מושך את השחקן למטה בהתאם לכוח הכבידה | ForcePlayerDown |
| **HITBOX DETECTION** | | | |
| AX = 0 if no impact detected, AX = 1 if impact detected | אין | מזהה התנגשות של השחקן מצד ימין | ImpactDetectionRight |
| AX = 0 if no impact detected, AX = 1 if impact detected | אין | מזהה התנגשות של השחקן מצד שמאל | ImpactDetectionLeft |
| AX = 0 if no impact detected, AX = 1 if impact detected | אין | מזהה התנגשות של השחקן מלמעלה | ImpactDetectionUp |
| AX = 0 if no impact detected, AX = 1 if impact detected | אין | מזהה התנגשות של השחקן מלמטה | ImpactDetectionDown |
| **KEYBOARD** | | | |
| מעדכן את הנתונים בהתאם:  [DS:RightArrowPress], [DS:LeftArrowPress], [DS:UpArrowPress] | אין | בודק את סוג הקלט שהשחקן הזין ובהתאם מעדכן נתונים | GetUserInput |
| ah = scan code, al = ASCII character or 0 | אין | בודק וקולט פעולת מקלדת של המשתמש | CheckAndReadKey |
| **GRAPHICS** | | | |
| אין | אין | מגדיר את המוד הגרפי | SetGraphic |
| זיכרון המסך | אין | מעדכן שינויים על המסך כגון אנימציות, סטטיסטיקות ותנועה | UpdateScreen |
| [DS:PlayerCoins] += 1  if PlayerCoins = 99:  [DS:PlayerCoins] = 0, [DS:PlayerLives] += 1 | אין | מוסיף לשחקן מטבע, אם הושגו 99 מעלה את החיים ומאפס את המטבעות, מציג אנימציית מטבע במקום בו הושג | SurpriseCoin |
| אין | אין | מציג את הסטטיסטיקות של המשחק על המסך | ShowStats |
| אין | [DS:Digits]: Array of Digits  [DS:TotalD]: Amount of Digits in Array  [DS:DigitXStart]: Starting Position on Screen | מציג מערך ספרות על המסך לפי נקודה נתונה | show\_numbers |
| אין | אין | מציג את מסך הפתיחה של המשחק | opening\_screen |
| אין | אין | מציג את מסך הטעינה של המסך | show\_loading\_screen |
| אין | אין | מציג את מסך הניצחון של המשחק | game\_over\_win |
| אין | אין | מציג את מסך ההפסד של המשחק | game\_over\_lose |
| אין | אין | מציג את מסך היציאה של המשחק | ExitScreen |
| אין | Parameters: (X, Y), Width, Height, Color | מצייר מלבן על המסך בהתאם לנתונים | draw\_rect |
| אין | אין | מציג אנימציית ניצחון של מריו הולך לטירה | MarioCastleAni |
| אין | אין | מציג את הטירה בכדי לתקן רקע | DrawCastle |
| אין | אין | מציג אנימציית דגל יורד על ידי מריו | FlagAniDraw |
| אין | אין | מתקן את העמוד שנפגע מהאנימציה | FlagPoleDraw |
| אין | אין | מציג את מריו גולש על העמוד ומורידו | MarioFlagAni |
| אין | אין | מציג את אנימציית מריו מת | DeathAniDraw |
| אין | אין | מציג אנימציית הליכה ימינה של מריו | PMovesRightAnimation |
| אין | אין | מציג אנימציית הליכה שמאלה של מריו | PMovesLeftAnimation |
| **BITMAP** | | | |
| אין | אין | מנהל את כל פעולות הBMP | OpenShowBMP |
| AX = [DS:FileHandle] | DX = File Name | קורא קובץ BMP | OpenBMPFile |
| אין | [DS:FileHandle] | קורא את הHeader של קובץ הBMP | ReadBMPHeader |
| אין | אין | קורא את לוח הצבעים של הBMP | ReadBMPPalette |
| זיכרון המסך | אין | מעתיק את לוח הצבעים של הBMP | CopyBMPPalette |
| אין | אין | מציג את קובץ הBMP על המסך | ShowBMP |
| אין | אין | קורא ממקום מסוים בBMP | PositionBMP |
| אין | [DS:FileHandle] | סוגר את קריאת קובץ הBMP | CloseBMPFile |
| **TIME** | | | |
| אין | Parameters: SleepTime = Amount of times to Sleep 100ms | הפעולה מקבלת כמה פעמים לקרוא ללישון 100 מילישניות, ובכך עוצרת את התוכנית לזמן נתון | Sleep |
| אין | אין | הפעולה ישנה 4 פעמים 25 מיליות, סה"כ 100 מילישניות | Sleep100ms |
| אין | אין | הפעולה נכנסת ללולאה ארוכה המתבססת על 50000/Max cycles שגורמת לעצירה של 25 מילישניות | Sleep25ms |
| אין | אין | הפעולה מאטה את הפרויקט למשך זמן קצר לצורך ביצוע אנימציות | MomentumSleep1 |
| אין | אין | הפעולה מאטה את הפרויקט למשך זמן קצר לצורך ביצוע אנימציות | MomentumSleep2 |
| [DS:Time] -= 1  [DS:LastTimeSeconds] = dh or same | אין | לוקחת את הזמן הנוכחי בשניות לתוך dh ובודקת אם הזמן בשניות שונה מהזמן הקודם שנשמר, ובכך מחליטה אם עברה כשנייה או לא | UpdateTime |
|  | | | |

**פרטים המאפשרים כניסה והפעלה**

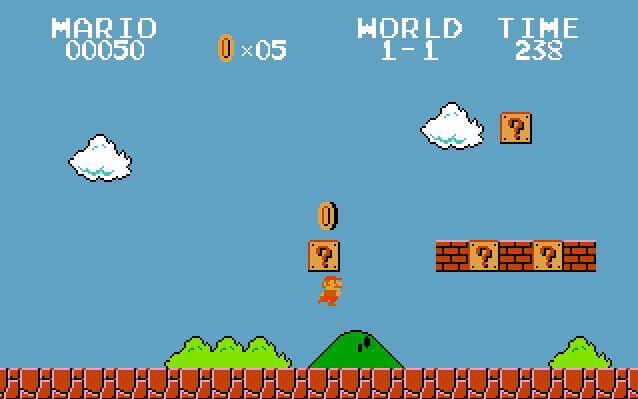
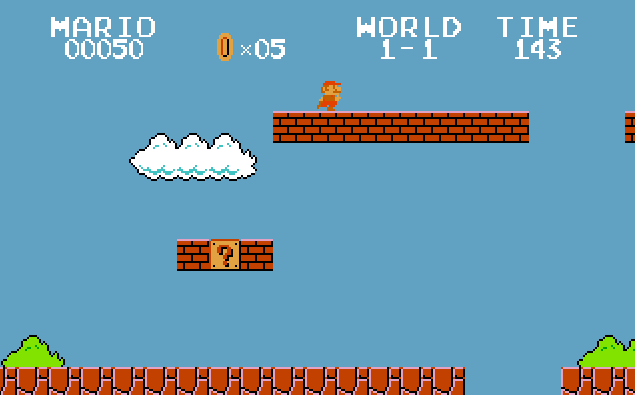
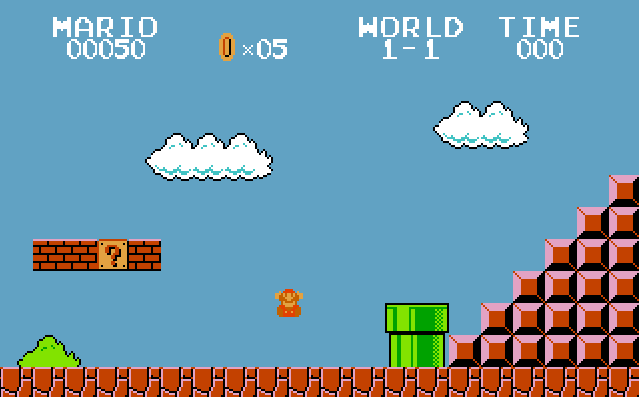
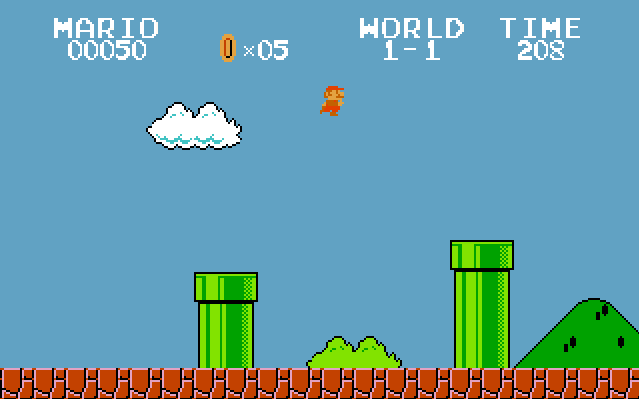
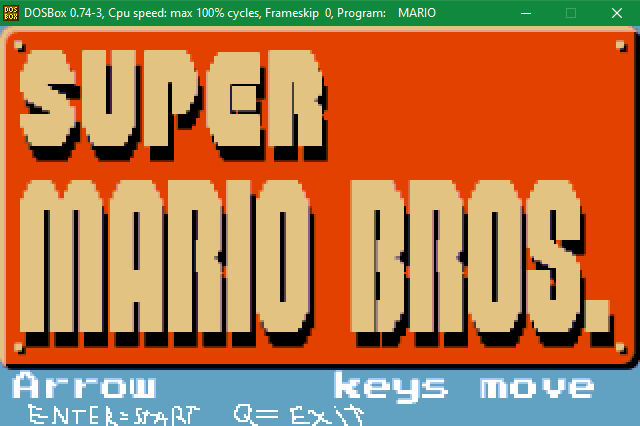
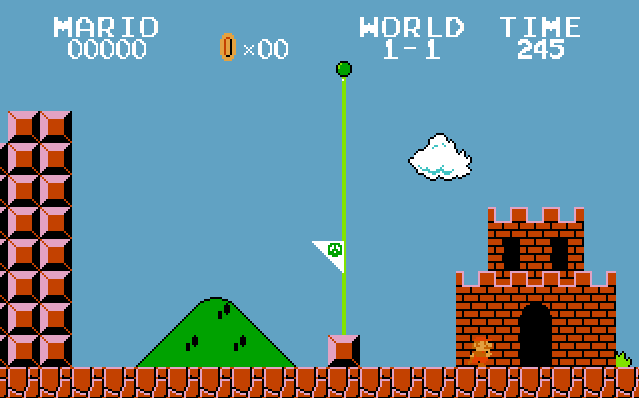
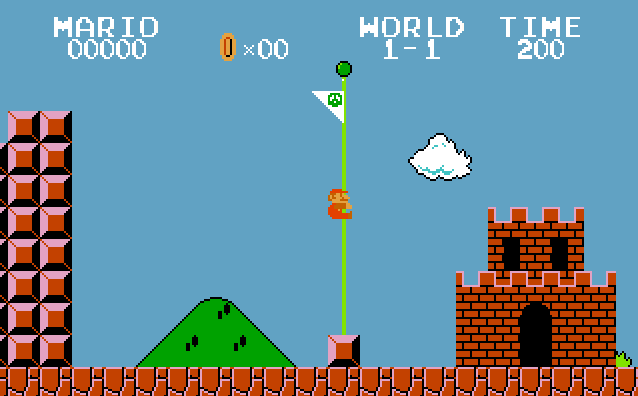
לצורך הפעלת הפרויקט דרושה הגדרה של ה-DosBox, בקובץ ה-Options, הנקראת cycles. ניתן למצוא אותה תחת קטגוריית ה-cpu, ומומלץ רבות ואף חובה לשנותה ל-max, אחרת המשחק יחדל מלתפקד כראוי.

מומלץ כמובן לשחק את המשחק על Full Screen באמצעות לחיצה על קיצור המקלדת Alt+Enter, לצורך חווית משחק מרבית. 

**קוד התוכנית**



**דוגמאות הרצה**



**סיכום אישי**

יחסית נהניתי מהתהליך, כי להצליח משהו שאתה עובד עליו הרבה זמן זה תחושה נהדרת, במיוחד אחרי הרבה debugging.

נתרמתי רבות מהפרויקט במספר דרכים:

1. למדתי לארגן תוכנית עבודה ולממש פרויקטים גדולים.

2. למדתי כיצד לגשת לפרויקט לפני תחילת כתיבת הקוד, ולעצב אותו.

3. למדתי לארגן את הזמנים שלי בצורה איכותית יותר לפעם הבאה.

4. שפת האסמבלי שקועה לי היטב בראש, ואני מבין אותה טוב מתמיד.

הפרויקט היה מהנה ליצור, מריו הוא משחק שתמיד אהבתי בתור ילד, והיצירה של האפשרות לחקור את העולם ואף להתנגש באובייקטים, לעשות אנימציות ותורגמן של מספר בבסיס 16 למספר דצימלי שמוצג על המסך מתוך תמונה של מספרים, היה דבר מהנה ומיוחד. נהניתי לשכלל את הפעולה שמציגה קובץ BMP על המסך ובכך להתאים אותה לצרכים שלי, על ידי הוספה של מגוון דברים כגון פרמטרים יותר קריאים וגמישים ואף היכולת לסנן צבעים מהרקע.

אם היה לי עוד זמן, או שהייתי מתכנן את הזמן שלי כראוי, הייתי מספיק אף להוסיף עולם, ומפלצות מיוחדות. זאת אני אקח לפעם הבאה, ואזכור את החשיבות של תכנון הזמן והפרויקט.

לבצע פרויקטים זה חוויה לכל דבר, כיוון שהיא מממשת רצון, ואני אשמח לעשות זאת רבות. בסופו של דבר למדתי רבות על עצמי ועל דרכי העבודה, ההתנהלות וכמובן, שפת האסמבלי.