פרק א'

שם פרוייקט: Clash Of Vikings

שם מתכנת: **אריאל כהן**

ת.ז**: 204902712**

מורה מלווה**: יוסי זהבי**

כיתה: **י'4**

בית ספר: **הרצוג**



פרק א'

שם קובץ:project.asm

קבצים נלווים:

amcard.bmp bluet1.bmp credits.bmp gamebg.bmp gbcard.bmp

giantb1.bmp giantb2.bmp giantr1.bmp giantr2.bmp gintr3.bmp

gintr4.bmp hamr0.bmp hamr1.bmp hamrb0.bmp hamrb1.bmp

hamrb2.bmp hamrb3.bmp hamrb4.bmp loseSc.bmp notcon.bmp

redt1.bmp rhmr0.bmp rhmr1.bmp rwhmr0.bmp rwhmr1.bmp

start.bmp taken.bmp tower.bmp warb0.bmp warb1.bmp

warr0.bmp warr1.bmp WINSCN.BMP wr.bmp wrab0.bmp

wrab1.bmp wrar0.bmp wrar1.bmp xmenrW0.bmp xmenrW1.bmp

xmenwB0.bmp xmenwB1.bmp xmhitB0.bmp xmhitB1.bmp xmhitR0.bmp

xmhitR1.bmp

סביבת העבודה

Notepad++

סביבת הרצה

DosBox

אופן ההפעלה

Mount %drive%

%drive%

Cd project

project

פרק ב'

**נושא העובדה**

העבודה היא משחק של שחקן 1 מול בוט מחשב. מטרת המשחק היא להרוס את שני המגדלים העוינים בעזרת החיילים שנמצאים לרשות השחקן ובאמצעות כמות חומר ה- blop שלו המאפשר לו לקנות חיילים. המשחק פותח בהשראת המשחק Clash Royale

**אופן ההפעלה**

מתחילים את המשחק על ידי לחיצת כפתור הSTART

גרסאות מערכת

**הגרסה ההתחלתית היא גרסה שמכילה באגים רבים ולא כוללת את תוכן המשחק נגד השחקן. הגרסה המצורפת היא הגרסה הסופית.**

תיעוד והסבר הפתרון

לולאה ראשית:

בלולאה הראשית קיימים שלושה פעולות:

1. DrawPlayersArray- שמכתיבה את השחקנים למפה דרך מערך המחזיק את כל המידע הנחוץ על השחקנים
2. checkIfNeedToGetTowersDowen- בודקת כל הזמן כחלק מהלולאה אם אחד מהבניינים נפל אם כן מעלימה את השחקן שעליו.
3. chcekWinner- בודקת כל הזמן אם אחד מהשחקנים מנצח אם כן אז הוא יוצא מהמשחק ומדפיס את זה על המסך.
4. AddBlop- הפעולה מוסיפה ‘blop’ כל איטרציה של הלולאה
5. MACRO- BLOP- מראה את כמות ה’blop’ של השחקן

בנוסף, קיימת פעולה א-סינכרונית הבודקת כל הזמן אם בשחקן מוסיף קלף חדש למפה אם כן אז יש לשחקן את האפשרות לבטל באמצעות לחצן ימני או לשים את הקלף באמצעות לחצן שמאלי.

**מבנה שמירת נתונים של שחקנים**

כל הנתונים הבאים שמורים בDATASEG- main

כל הנתונים על השחקנים שעל המפה נשמרים במערך דו מימדי כאשר כל תא קטן בו מכיל 12 תאים קטנים יותר של WORD להלן התאים:

1. מיקום הX של השחקן במפה
2. מיקום הY של השחקן במפה
3. סוג השחקן- כחול או אדום + איזה דמות הוא(ענק, גרזנאי או לוחם)
4. הכיוון שלו במפה
5. כמות החיים שלו
6. האינדקס של השחקן במערך
7. תא סתמי אחד נוסף למקרה שאצטרך עוד משהו ופספסתי

גרפיקה כללית

על מנת למנוע ריצודים במשחק, בתחילת המשחק הופעלה פעולה אשר מעבירה למערך את כל ערכי הפיקסלים של המפה

וכאשר נרצה למחוק דמות מהמפה ניקח את אותם פיקסלים התואמים את אותה הדמות ונשים אותם על מקום הדמות.

יש לציין כי רוב הגרפיקה נכתבה למסך באמצעות תמונות BMP

**תרשימי זרימה**

**תרשים זרימה – DrawPlayersArray**

פעולה המתארת את כניסת השחקנים למפה

תחילת לולאה בכמות השחקנים שיש במערך של השחקנים

בדיקה אם יש אויב ליד השחקן הקרוב

לא

בדיקה אם השחקן נמצא ליד בניין מת

כן

לא

כן

להרביץ לו

לצייר שחקן אחד מהמערך קריאה לפעולה boot

לשנות כיוןן לבניין החי

**תרשים זרימה - boot**

לולאה באורך כמות השחקנים במפה

הוספת בלופ אם צריך

כן

כן

בדיקה אם כל האוייבים של הבוא מתים

לא

כן

בדיקה אם השחקן העכשווי בלולאה מת

חזרה להתחלת הלולאה

לא

בדיקה אם השחקן הנבדק הוא שחקן של הבוט

לא

בדיקה באיזה צד צריך לשים שחקן מגן cxוהעברת אותו ערך לרגיסטר

לא

לשים שחקן רנדומלי על הצד שנקבע קודם לכן

כן

בדיקה אם יש מספיק bloop בשביל לשים את השחקן

**תרשים זרימה - CheckCard**

פעולה המכניסה שחקנים חדשים למפה באמצעות העכבר בפעולה א-סינכרונית

את הערך 0selectOrPut העברה למשתנה

הכנסת השחקן למסך

את הערך 0selectOrPut העברה למשתנה

לא

כן

כן

כן

בדיקה אם התבטלה הקריאה באמצעות לחיצת לחצן ימני

בדיקה של איזה קלף נלחץ והעברה של playertypeputסוגו למשתנה גלובלי

ל1selectorputשינוי הערך של המשנה

משהו אחר

**0**

בדיקה של משתנה גלובלי - selectORput

**תיאור הפעולות**

|  |  |  |  |
| --- | --- | --- | --- |
| שם פעולה | Input | OutPut | הסבר |
| startGamei | - | משתנה- startOrExit הקובע מה היא הפעולה הבאה שהמשתמש רוצה לעשות | הפעולה מתחילה את התוכנית עם מסך כניסה |
| showCredits | - | - | הפעולה מראה בתמונה את כל האנשים אשר תרמו להכנסת הפרוייקט |
| winOrLoseImage | משתנה גלובלי – won המשתנה מחזיק את המידע של השחקן שניצח במשחק- (הבוט או המשתמש) | הצגת תמונה של ניצחון או הפסד בהתאם לinput | הפעולה מציגה מסך ניצחון או הפסד בהתאם לכך אם השחקן ניצח או הפסיד |
| addBloop | משתנה גלובלי – timeChanged | מעלה את הbloop אם צריך | מוסיפה bloop כל שתי שניות למשתמש ולבוט כל עוד הוא לא הגיע ל10 |
| boot | positionIndexeder-משתנה גלובלי  playersPositions-  מערך גלובלי | מוסיפה שחקנים אויבים למסך במקרה הצורך | הפעולה מתפעלת את הבוט כפי שתואר בתרשים הזרימה |
| putEnemyInRandomPosition | - | מוסיפה שחקן אויב למסך | הפעולה שמה שחקן אויב במסך |
| generateRandomRedEnemy | BloopBoot-משתנה גלובלי המחזיק את כמות הbloop של הבוט | מחזירה בBX סוג שחקן רנדומלי רק אם יש מספיק bloop בשביל לשים אותו במסך | למצוא שחקן רנדומלי אשר יש מספיק bloop לאויב בשביל לשגר אותו |
| checkIfAllPlayersDead | positionIndexeder-משתנה גלובלי  playersPositions-  מערך גלובלי | מחזירה במשתנה גלובלי AllDead אם כל השחקנים של המשתמש מתים | בודקת אם כל השחקנים של המשתמש מתים |
| DrawPlayersArray | positionIndexeder-משתנה גלובלי  playersPositions-  מערך גלובלי | מציירת את השחקנים אשר צריך לצייר במסך | הפעולה אחראית על כל הציור של השחקנים במסך |
| checkIfplayerEnemynearForAll | positionIndexeder-משתנה גלובלי  playersPositions-  מערך גלובלי | מציירת שחקנים מרביצים במסך במקרה הצורך | בודקת אם יש שני שחקנים אויבים שינם משתייכים לאותו קבוצה ואם כן גרומת להם להראות זאת במסך ולהוריד חיים |
| decOrAddByType | מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | משנה את המיקום על ציר הY של השחקן שהתקבל במקרה הצורך | מטרת הפעולה היא לשנות את המיקום על ציר הy של השחקן שהתקבל. |
| checkIfTowerNear | מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | מרביצה ומורידה חיים אם יש בניין קרוב לשחקן שהתקבל | מרביצה ומורידה חיים אם יש בניין קרוב לשחקן שהתקבל בהתאם לצבע של השחקן ולסוגו |
| redTowerNear | מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | מרביצה עם השחקן שהתקבל אם נמצא בניין אדום | מרביצה עם השחקן שהתקבל אם נמצא בניין אדום |
| blueTowerNear | מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | מרביצה עם השחקן שהתקבל אם נמצא בניין כחול | מרביצה עם השחקן שהתקבל אם נמצא בניין כחול |
| checkIfLeftRedTowerDown | redTowersLife- משתנה המחזיק את החיים של הביניינים | מחזירה במשתנה dead אם הבניין הוחרב | בודקת אם הבניין האדום השמאלי הוחרב |
| checkIfLeftBlueTowerDown | redTowersLife- משתנה המחזיק את החיים של הביניינים | מחזירה במשתנה dead אם הבניין הוחרב | בודקת אם הבניין האדום הימני הוחרב |
| checkIfLeftBlueTowerDown | blueTowersLife- משתנה המחזיק את החיים של הביניינים | מחזירה במשתנה dead אם הבניין הוחרב | בודקת אם הבניין הכחול השמאלי הוחרב |
| checkIfRightBlueTowerDown | blueTowersLife- משתנה המחזיק את החיים של הביניינים | מחזירה במשתנה dead אם הבניין הוחרב | בודקת אם הבניין הכחול הימני הוחרב |
| hitBasedOnPlayerTypeToTowerBlue | Bx = סוג השחקן וקבוצתו  Dx = top  Ax = left | מראה על המסך שחקן מהקבוצה הכחולה מרביץ לביניין כחול | מרביצה לבניין מהקבוצה האדומה בהתאם לסוג השחקן |
| hitBasedOnPlayerTypeToTowerRed | Bx = סוג השחקן וקבוצתו  Dx = top  Ax = left | מראה על המסך שחקן מהקבוצה האדומה מרביץ לביניין אדום | מרביצה לבניין מהקבוצה הכחולה בהתאם לסוג השחקן |
| hitBasedOnPlayerTypeToenemy | input:  1. bl = player type  2. dx = top  3. ax =left | במשתנה -amountToLower את הכמות של החיים שצריך להוריד בהתאם לסוג שהתקבל בנוסף מראה את סוג השחקן שהתקבל מרביץ לאויב במסך | להחזיר את כמות החיים שצריך להוריד לאויב בהתאם לסוג השחקן שהתקבל והראות את אותו סוג שחקן שהתקבל מרביץ לאויב אליו או קרוב |
| getPlayerSide | Ax = ערך ציר x | מחזירה בBH ‘A’ אם הX שהתקבל משתייך לצד השמאלי מחזירה בBH ‘B’ אם הX שהתקבל משתייך לצד הימני | להחזיר לאיזה צד משתייך הX שהתקבל כפרמטר |
| lowerRedLifeTower | input = bx  bl = player type  bh = 'A'-for left tower or 'B'- for right tower | - | מוריד חיים לבניין אדום בהתאם לצד ולסוג השחקן שהתקבלו כפרמטרים |
| lowerBlueLifeTower | input = bx  bl = player type  bh = 'A'-for left tower or 'B'- for right tower | - | מוריד חיים לבניין כחול בהתאם לצד ולסוג השחקן שהתקבלו כפרמטרים |
| cheangeDirectionBasedOnLiveTowersForBlue | x = ax, y = dx  redTowersLife- משתנה גלובלי  מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | - | משנה את הכיוון של השחקן הכחול שהתקבל בSI אם הביניין האדום שנמצא מולו הוחרב |
| cheangeDirectionBasedOnLiveTowersForRed | x = ax, y = dx  blueTowersLife- משתנה גלובלי  מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | - | משנה את הכיוון של השחקן האדום שהתקבל בSI אם הביניין הכחול שנמצא מולו הוחרב |
| checkPlayerPath | מקבלת בCX ערכי X על המפה  מקבלת בBL את סוג השחקן (אדום או כחול) | מחזירה בCX את ערכי הX של הצד שאמור להיות על פי הפרמטר  מחזירה בDX את ערכי הY ההתחלתים שאמורים להיות לשחקן ברגע שמשוגר למשחק בהתאם לBL | להחזיר את ההערכים ההתחלתיים שאמורים להיות לשחקן |
| checkRedLeftTowerDown | redTowersLife- משתנה גלובלי | אם הבניין השמאלי האדום מת הפעולה מוחקת אותו מהמסך | למחוק את הבניין האדום השמאלי מהמסך אם הוא הוחרב |
| checkRedRightTowerDown | redTowersLife- משתנה גלובלי | אם הבניין הימני  האדום מת הפעולה מוחקת אותו מהמסך | למחוק את הבניין האדום הימני מהמסך אם הוא הוחרב |
| checkBlueLeftTowerDown | blueTowersLife משתנה גלובלי | אם הבניין השמאלי  הכחול מת הפעולה מוחקת אותו מהמסך | למחוק את הבניין הכחול השמאלי מהמסך אם הוא הוחרב |
| checkBlueRightTowerDown | blueTowersLife משתנה גלובלי | אם הבניין הימני  הכחול מת הפעולה מוחקת אותו מהמסך | למחוק את הבניין הכחול הימני מהמסך אם הוא הוחרב |
| checkIfNeedToGetTowersDowen | - | מוחקת בניין מהמסך אם הוא צריך להימחק | מוחקת בניין מהמסך אם הוא צריך להימחק |
| DrawPlayer | input [BmpTop] = y, [BmpLeft] = x,  bx = index of card [direction ] = direction of movment | מציירת תמונה של השחקן שניתן ומראה מעליו את החלק שהיה על המפה קודם לכן על מנת למנוע ריצוד | מציירת תמונה של השחקן שניתן ומראה מעליו את החלק שהיה על המפה קודם לכן על מנת למנוע ריצוד |
| checkDirectionAndAct | Direction- משתנה גלובלי  מקבלת בSI אופסט לתא במערך הדו מימדי של כל השחקנים | - | לשנות את המיקום של השחקן בהתאם לכיוון שקיבלו |
| checkPlayerType | מקבלת בBL- הקבוצה של השחקן  מקבלת בBH את סוג השחקן  input [BmpTop] = y, [BmpLeft] = x, | מציירת כל המסך את סוג השחקן שהתקבל | לצייר את השחקן שהתקבל בהתאם לפרמטרים |
| MoveGiant | מקבלת בBL- הקבוצה של השחקן  input [BmpTop] = y, [BmpLeft] = x, | מציירת את הענק מקבוצה הנתונה בפרמטר על המסך | לצייר את הענק מהקבוצה הנתונה בפרמטר על המסך |
| moveWrrior | מקבלת בBL- הקבוצה של השחקן  input [BmpTop] = y, [BmpLeft] = x, | מציירת את הלוחם מקבוצה הנתונה בפרמטר על המסך | לצייר את הלוחם מהקבוצה הנתונה בפרמטר על המסך |
| moveHmmer | מקבלת בBL- הקבוצה של השחקן  input [BmpTop] = y, [BmpLeft] = x, | מציירת את הגרזנאי מקבוצה הנתונה בפרמטר על המסך | לצייר את הגרזנאי  מהקבוצה הנתונה בפרמטר על המסך |
| checkWinner | - | מחזירה במשתנה Won אם שחקן ניצח ואם כן מחזירה גם את סוגו | בודקת אם אחד מהקבוצות ניצחו ואם כן אומרת מי |
| drawGiantCard | - | מחזירה על המסך ציור של הקלף של הענק | לצייר ציור של הקלף של הענק על המסך |
| hammerCard | - | מחזירה על המסך ציור של הקלף של הגרזנאי | לצייר ציור של הקלף של הגרזנאי על המסך |
| wrriorCard | - | מחזירה על המסך ציור של הקלף של הלוחם | לצייר ציור של הקלף של הלוחם על המסך |
| loadMap | - | - | מעלה לזכרון תמונה עכשווית של המפה |
| loadTowerToMat | si = offset to the array that needed to the tower  mapTop = top of the tower neded  mapLeft = Left of the tower neded | - | מעלה למערך בזיכרון פיקסלים של המגדל הנתון |
| LoadMatToScreen | si = offset to array with the data  mapTop = top to where to put the mat  mapLeft = Left To where To put the mat | מעלה את הפיקסלים שנמצאים במערך הנתון למיקום שניתן- מערך | לשים במסך את הפיקסלים של המערך שניתן על פי המיקום הנתון- המערך שניתן הוא מערך שמצביע על פיקסלים של בניין |
| loadMapToScreen |  | מעלה למסך סטטוס של מפה | הפעולה מעלה למסך את הפיקסילים של המפה אשר נשמרו במערך בפעולה loadMap |
| checkCard | תואר בתרשים זרימה | תואר בתרשים זרימה | תואר בתרשים זרימה |
| canceldaction | - | - | מבטלת את הגבולות של העכבר |
| putPlayerInMap | playerTypePut | - | מכניסה למערך של השחקנים את השחקן הניתן במשתנה |
| putWrroir | bloopUser | - | מכניס למערך של השחקנים את לוחם כחול רק אם יש מספיק bloop |
| putHammer | bloopUser | - | מכניס למערך של השחקנים את לוחם כחול רק אם יש מספיק bloop |
| putGiant | bloopUser | - | מכניס למערך של השחקנים את לוחם כחול רק אם יש מספיק bloop |
| MovePosition | cx = x of player, dx = y of player bx= player index | מעלה את המשתנה positionIndexer | להכניס שחקן כחול למערך של השחקנים בהתאם לפרמטרים |
| boundries | - | משנה מיקום עכבר | להעביר את העכבר למרכז המסך |
| asyncMouseCard | - | - | מעלה את הסאב רוטינה – checkCard לפעולה האסינכרונית של מערכת ההפעלה |
| showBlueTowers | - | מראה את המבנה של הבניינים הכחולים | לשים על המסך את המבנה של הביניינים הכחולים |
| showTowersFacesBlue | - | תמונות על המסך | להציג מעל הביניינים הכחולים לוחמים של הביניין |
| showTowersFacesRed | - | תמונות על המסך | להציג מעל הביניינים האדומים  לוחמים של הביניין |
| showRedTowers | - | מראה את המבנה של הבניינים האדומים | לשים על המסך את המבנה של הביניינים האדומים |
| RandomByCs | 1. Bl = min (from 0) , bh, Max (till 64k -1)  2. RndCurrentPos a word variable, help to get good rnd number  Declre it at DATASEG : RndCurrentPos dw ,0  3. EndOfCsLbl: is label at the end of the program one line above END start | מחזיר בAL מספר רנדומאלי בין bl bh | להחזיר מספר רנדומאלי בין הפרמטרים הנתונים |
| OpenShowBmp | BmpLeft – המיקום X של התמונה הרצויה  BmpTop- המיקום Y של התמונה הרצויה  BmpHeight – גובה התמונה  BmpWidth – רוחב תמונה  dx- אופסט בזיכרון של שם התמונה הרצויה | הצגה של התמונה על המסך | להציג את התמונה על המסך במיקום הרצוי |
| OpenTransBmp | BmpLeft – המיקום X של התמונה הרצויה  BmpTop- המיקום Y של התמונה הרצויה  BmpHeight – גובה התמונה  BmpWidth – רוחב תמונה  dx- אופסט בזיכרון של שם התמונה הרצויה | הצגה של התמונה על המסך ומחיקה של כל צבא לבן ושחור | להציג את התמונה על המסך במיקום הרצוי ומחיקה של צבא לבן ושחור |
| ReadBmpHeader | FileHandle - משתנה שמחזיק את האנדל של התמונה | - | העברה למערך Headr את האדר של התמונה |
| ReadBmpPalette | - | - | העברה למערך palette את הפלטת צבעים של התמונה |
| CopyBmpPalette | - | - | העברה לזכרון של פלטת הצבעים של המסך את פלטת הצבעים של התמונה |
| hitWrrior | dx = top ax = Left  bx= סוג שחקן וקבוצה | העברה למסך לוחם מרביץ בהתאם לפרמטר שניתן | העברה למסך לוחם מרביץ בהתאם לפרמטר שניתן ובמיקום שניתן |
| hitHammer | dx = top ax = Left  bx= סוג שחקן וקבוצה | העברה למסך גרזנאי מרביץ בהתאם לפרמטר שניתן | העברה למסך גרזנאי מרביץ בהתאם לפרמטר שניתן ובמיקום שניתן |
| hitGiant | dx = top ax = Left  bx= סוג שחקן וקבוצה | העברה למסך ענק מרביץ בהתאם לפרמטר שניתן | העברה למסך ענק מרביץ בהתאם לפרמטר שניתן ובמיקום שניתן |
| \_100MiliSecDelay | - | - | מפעילה דיליי על התוכנית של 100 מילי שניות |
| copyBuffer | bufferTop- מיקום Y ממנו נרצה לקחת מידע מהמסך  bufferLeft-מיקום X ממנו נרצה לקחת מידע מהמסך | - | העברה למערך Buffer מידע מהמסך בגודל 20 על 20 מהמיקום הנתון |
| MakeMask | - | SI | מעבירה לSI את הטווח הרצוי |
| loadBuffer | bufferTop- מיקום Y ממנו נרצה להעלות מידע אל המסך  bufferLeft-מיקום X ממנו נרצה להעלות מידע אל המסך | העברת הפיקסלים הנמצאים בBuffer אל המיקום הרצוי | העברת המידע הפיקסלי הנמצא המערך buffer אל המיקום הרצוי |
| proc DrawHorizontalLine | Cx = מיקום X התחלתי לקו  Dx = מיקום Y התחלתי לקו  Si = אורך הקו  Al = צבע רצוי | ציור של קו מקביל לציר הX על המסך | ציור של קו מקביל לציר הX על המסך |
| DrawVerticalLine | Cx = מיקום X התחלתי לקו  Dx = מיקום Y התחלתי לקו  Si = אורך הקו  Al = צבע רצוי | ציור של קו מקביל לציר הY על המסך | ציור של קו מקביל לציר הY על המסך |
| SetGraphic | - | - | העברה למצב גרפי |
| ShowTransBMP | BmpLeft – המיקום X של התמונה הרצויה  BmpTop- המיקום Y של התמונה הרצויה  BmpHeight – גובה התמונה  BmpWidth – רוחב תמונה | ציור של התמונה הרצויה על המסך ללא הצבעים שחור ולבן | ציור של התמונה הרצויה על המסך ללא הצבעים שחור ולבן על פי פלטת הצבאים והמיקום של השחקן  ללא הצבעים שחור ולבן |
| ShowBMP | BmpLeft – המיקום X של התמונה הרצויה  BmpTop- המיקום Y של התמונה הרצויה  BmpHeight – גובה התמונה  BmpWidth – רוחב תמונה | ציור של התמונה הרצויה על המסך | ציור של התמונה הרצויה על המסך ללא הצבעים שחור ולבן על פי פלטת הצבאים והמיקום של השחקן |
| openFile | input dx filename to open | אנדל של הקובץ במשתנה- filehandle | לפתוח קובץ ולהחזיר אנדל |
| CloseFile | Filehandle משתנה | - | סוגר קובץ עם אנדל |
|  |  |  |  |

קוד התוכנית



הקוד מצורף למעלה וניתן להורידו.

תמונות הרצה

תמונה שמכילה מפה

התיאור נוצר באופן אוטומטי

סיכום אישי

אני מרגיש שהפרויקט היה חוויה אישית טובה וכיפית לעשות למרות הקושי הרב ביישום של הפתרון. בנוסף, אני חושב שהפרויקט תרם לי מאוד מבחינת הבנת המחשב והגרפיקה של DOS. למדתי המון דברים כמו: הצגה של תמונות במסך, העובדה שDOS עובד עם תוכנית אחת כל פעם, כיצד לעבוד עם מערכים דו ממדדים באסמבלי, עבודה עם שני סגמנטים של זיכרון ועוד הרבה מאוד.

לסיכום, החוויה שלי מהפרויקט מאוד חיובית ומהנה.