**Adventures of tron**

**תהליכים שמומשו/שונו -**   
**Displayer** – התהליך הזה לא שונה מהמקור, תפקיד התהליך הוא לבצע את ההדפסה למסך דרך המערך של ה DISPLAY לכתובת B800H.

**Updater** – לאחר ביצוע שינויים עליו, תפקידו לבצע שינויים בתנועת השחקן, כמו כן בשורותיו האחרונות הוא מבצע עידכון של כל המיקומים החדשים במערכת. בנוסף, תנועת המעליות בעת מעבר נקודת המרכז, או קלט מהשחקן בעת קיום התנאים, נמצא כאן.

**Receiver** – תפקידו לנהל מערך תווים , מעין באפר, המתקבל מפסיקת המקלדת ונשמר. עשינו שימוש בבאפר הנ"ל כדי לנהל חלק מתנועת השחקן – למנוע דילוג על תנועה ( החזקת מקש כיוון למשל), על ידי בדיקת תו קודם, אם איננו זהה ( לא שיחררנו) אז תדלג את מימושו.

**Checker** – זהו תהליך שאנו יצרנו , בכדי לנהל הצלבות של השחקן עם המוט בכל שלב במשחק, בקפיצה או בהליכה על הקומה התחתונה.

**Game\_Over** – התהליך הינו תהליך שאנו יצרנו , תפקידו לסיים את התהליכים הקודמים שרצו – כל מה שנ"ל. בנוסף הוא יציג את מסך הסיום של המשחק, בו יוצג לשחקן התוצאה שהשיג, וכמו כן הזמן שלו IN – GAME, כמה זמן שיחק.

**מבני נתונים ושימושיהם-**

מימשנו בעיקר Structs , עבור דברים רבים, למשל מבנה של Position שמחזיק עבור כל משבצת את הערך הקורדינטה שלה.  
מבנים נוספים :

\***Block** – for each block hold the character and attribute for the B800H definition.  
\***Enemy** – For each enemy hold it’s height, width, blocks ( it’s color and character), his current floor, is he active or waiting for spawn, and his direction. Also we held an array of these structs to represent more enemies in our system.  
\***Floor** – a struct to represent the makings of the floors/platforms, also hold it’s height ( actual floor location).  
\***Ladder** – a struct holding the ladder’s position and it’s attributes  
\***Elevator** – also a struct that is held in an array of structs similar to the enemies, hold it’s attributes, its status – activated or not, in order to engage movement, and it’s width.  
\***Tron** – a struct holding the attributes of the tron, also his floor location.  
\***Fbits** – also an array of structs, holding each bit and it’s score according to the spawn floor for the summation part.  
\***Sailers** – also held in an array of structs, according to their position change the char to represent it to be as such, also spawn catch and active information is held.  
**\*Tank** – represents all the info needed for the tank to exist

**\*Fire** – a struct to hold the character and location of the shell fired from our tank on the third level.

|  |  |
| --- | --- |
| **הדרישה בחלק א'** | **המימוש בחלק ב'** |
| חציית קו האמצע = סימון מעליות(Extra)+ איפשור תנועה | בפונקציה Checker אנו משווים את המיקום של השחקן עם פיקסלי המוט, כמו כן בעזרת דגלים וקטע קוד ב Updater שמזיז את המעליות. |
| מימוש חיים וניקוד (UI+) | הצגת חיים על ידי כוכביות, מחיקת כוכבית (הדפסה מחדש) בעת פגיעה באוייב, וסכימת תוצאה לפי תנאי המשחק כולל הצגתה. |
| תנועות אוייבים ותזוזות, כולל Spawn delay ויציאות רנדומאליות בשלבים מתקדמים יותר | תנועת האוייבים נעשת בפסיקה 70, שם אנו גם מעכבים יציאת אוייב אחר בכדי ליצור רווח בין אוייב לאוייב, על ידי שינויים בשדה Active במבנה האוייב, וכמו כן קיצוב התנועה והיצירה על ידי מניפולציות זמנים על קאונטר המערכת  if (count0x70 % (RTC\_RATE / ENEMY\_RATE) == 0) |
| שינוי קצב אוייבים לאחר דקה פר שלב | קיום תנאי אשר כעבור דקה (RTC\_RATE\*60) משנה את ה- ENEMY\_RATE להיות גדול פי 2 – כי בנ"ל הוא מחלק את הקצב הכללי בעזרת דגל– ובכך נאיץ את האוייבים. |
| מימוש מפרשית עזר לשחקן | לאחר איתחול המפרשית בפונקציית init\_sailers() יש לנו פונקציה שמשנה דגלים בשביל תנאים בתזוזת המפרשית - caught\_sailer() ותנאי התזוזה וההצגה נמצאים בפסיקה 70. |
| ביטים מעופפים+צבירת ניקוד+מעבר שלב | מימשנו מערך של ביטים עם מיקום מוגרל, בפונקצית init\_fbits() אשר מביא ניקוד שונה פר קומה. כמו כן איסופן מביא לטריגר של צביעה בכחול, שמאפשר שש שניות של עליית שלב. |
| מימוש פסילות | לאחר כל הצלבת מיקום, שמביאה לפגישה עם אוייב, נחסר חיים, נעלה דגל init\_tron להיות TRUE , בעזרת תנאי בפסיקה 70 נפיל את השחקן חזרה לקומה הראשונה ממרכז המסך, ונדפיס את החיים ב UI. |
| הצגת שלב המשחק (Extra) | הצגנו את שלב המשחק על ידי הפיכת גוף השחקן (בין רגליו לראשו) להיות מספר השלב בו הוא נמצא. |
| סאונד באוונטים מסויימים | הדלקה וכיבוי בפסילה, בעליה במעליות, בירידה רצונית של השחקן ובקפיצה. ההדלקה נעשת בעזרת עידכון LATCH ומשתנה של מהירות, לאחר מכן קריאה לפונקציות שינוי הלאטצ', הדלקה וכיבוי הרמקול. |
| שינוי קצת האוייבים ב20% פעם בחמש שניות , ביחס זה לזה בין הקבוצות | מומש שינוי מקצבים בין האוייבים השונים בשלב שלוש, נעשה בפונקציית init\_enemy3() ו – change\_enemy\_speed () |
| קפיצה | הקפיצה נעשת על פי טריגר מהמקלדת שנקלט, לאחר מכן לפי סטטוס הדגל jump , הוא עולה ומתוזמן בסיקה 70, לאחר מכן כמו מערכת הודעות, בסוף העליה הוא מסמן לתחילת הירידה שנעשת מספר שורות לאחר מכן. |
| מימוש מסך סיום למשתמש (Extra) | בתהליך נפרד, Game\_Over , נעשה מימוש של הריגת התהליכים והסבר מפורט יותר נמצא בהסברי התהליכים לעיל. |
| טנק ויריה | מומש טנק כחול ונחמד שיורה פגז פעם בקומה, כאשר הוא יוצא אך ורק בקומה השלישית. הפגז הינו בצורת תו נגיגנ. מומש בתוך פסיקה 70 בכדי לשלוט על מהירות התנועה הן שלו והן של הפגז. |