

פרויקט גמר

מה"ט

לתואר הנדסי במחלקה: תוכנה.

שם הפרויקט: XO-Helicopter

שם הסטודנט/ים: פואד חורי, זיני מר암.

ת.ז: 315954834, 207280710

מקום ביצוע העבודה : מכללת אורט ברואודה.

תאריך הגשה:

שם המנחה: עסקלה שדי.

חתימת ראש המחלקה

חתימת המנחה

תאריך

תאריך

הצהרת סטודנט

שם הסטודנט: לירן אליאס ת.ז. 507280702

שם המכללה בה למד הסטודנט: אליאס לירן כלאוזן

אני הכח"ם, מצהיר בזאת כי פרויקט הגמר וספר הפרויקט המצח"ב נעשו על ידי בלבד.
פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.
פרויקט הגמר וספר הפרויקט נעשו על בסיס הנקיטתו של המנחה האישי.
אני מודע לאחריות שהנני מקבל על עצמי על חתימתה זו שכל הנאמר בה אמת
ורוק אמת.

תאריך: 30/09/2019

חתימת הסטודנט: 

הצהרת סטודנט

שם הסטודנט: ניר כוכב ת.ז. 315954834
שם המכללה בה לומד הסטודנט: טולן קלעוקה

אני הח"מ, מצהיר בזאת כי פרויקט הגמר וsofar הפרויקט המצח"ב נעשו על ידי בלבד.
פרויקט הגמר נעשה על סמך הנושאים שלמדתי במכללה ובאופן עצמאי.
פרויקט הגמר וsofar הפרויקט נעשו על בסיס הנחייתו של המנחה האישי.
אני מודע לאחריות שהנני מקבל על עצמי על ידי חתימתה על הצהרה זו שכל הנאמר בה אמת
ורק אמת.

תאריך: 30/9/2019

חתימת הסטודנט: 



תאריך: _____

לכבוד
יחידת הפרויקט
מה"ט

א. פרטי הסטודנטים

שם הסטודנט	ת.ז. 9 ספרות	כתובת	טלפון נייד	תאריך סיום הלימודים
מרעם זוני	207280710	מזורעה	0543192566	2019
פואד חורי	315954834	ערabeeha	0502685335	2019

שם המכללה: אורט ברואודה.

סמל המכללה:

מסלול ההכשרה: הנדסאים.

מגמת לימוד: הנדסת תוכנה

מקום ביצוע הפרויקט:

ב. פרטי המנהה האישי

שם המנהה *	כתובת	טלפון נייד	תואר	מקום העבודה/תפקיד
שדי עסאלה	ס. פ. 1/ 4567724	050-5555555	M.A.	מטען מה"ט

* עברו מנהה איש חדש יש לצרף קורות חיים, ניסיון מקצועי ותעודות השכלה לאישור מה"ט.

חותמת הגורם המקצועית
חתימת המנהה הראשי
חותמת הסטודנט

דרך מנחם בגין 86 תל אביב ת.ד. 36049 מיקוד 67138

טלפון: 03-7347644 | פקס: 03-7347521

1. שםuproject

XO Helicopter

2. רקע

2.1 תיאור ורקע כללי

משחק חשיבה הוא משחק שבו יכולתו האינטלקטואלית של השחקן היא אשר מכיריה את הcaf. בנוסף לחשיבה מעמיקה דורשים משחקי חשיבה גם ריכוז וסבלנות". המשחק הוא שעשוע אך אינו מהו היפוכה של הרצינות".

משחקי חשיבה יכולים לתרום להתחזקות קוגניטיבית של הילד כאשר הם מותאמים לגילו. כאשר המשחק מהציב אתגרים מוחשיים עבור הילד, התמודדות מול האתגרים הללו מעודדת אותו לשפר את יכולתו הקוגניטיבית ולהפעיל יצירתיות בפתרון הבעיה שם מעלים.

לפי כך אנו רוצים לפתח משחק חשיבה שדורש הרבה ריכוז ותכוון ב OpenGL לראייה תלת ממדית.

המשחק מכיל שתי מופת לשחק אשר כל מפה מוכלת מרחב ארוך שמכיל האובייקטים:

שני מטוסים שאנו יצרנו.

קופסאות (כל קופסה מסומנת בסימן איקס או עיגול).

טנק שמכילה כמה טילים.

בניינים.

המטרה הינו השחקן שלנו אשר נעה בשמיים שמכיל מערכת היגוי כוללת הגה גבוהה (שולט על התנועה במישור העילרוד), הגה כיוון (סבסוב) ומאותות (גלאול), תנועת המטוס על ידי הלחצנים במקלדת). המשחק ישנים שני שחנים אשר מכה אחד לשני כדי לשחק (נקבע זמן מסוים).

דרישת השחקן לאסוף קופסאות למרחב המתאים לסימן שנקבע לו (קביעת הסימן לשחקן תהיה מהפעם הראשונה שימוש שמשחק ושם הסימן במשבצת משחק ה X ו O), יכול להיות בזמן איסוף הקופסאות הטנק מכוען המטוס, אם הטיל פגע במטוס הקופסה נפלת.

המטרה גם כך מכיל טילים אשר יכול לזרוק טילים על הטנקים שנמצאים למרוחק, ואם הטיל פגע בטנק כמה פעמים יתפוץן.

המשחק יסתתיים בשני מצבים:

1. איסוף הקופסאות המתאימות למלא שורה או עמודה או אחד מהאלכסונים בסימן שנקבע לכל שחזור, השחקן שמשים ראשון הוא המנצח.

2. זריקת טילים על המטוסים כמה פעמים המשחק יסתום בכישלון.

2.2 מטרות המערכת

- אנו רוצים לפתח משחק שהוא שילוב של שני משחקים, אחד מהמשחקים קוראים לו אייס מיקס דרייקס, המשחק משלב בין שני שחנים שימושיים ביחיד על לוח 3×3 של משבצות, אשר השחקנים הם השולטים במטוסים הנעים שצרכיהם לאסוף קופסאות מהמרחוב, כל קופסה תהיה בצורה של סימן אייס או עיגול.
- המשחק דורש ריכוז ברמה גבוהה ובכך נזoor לשחקנים שיש להם בעיות ריכוז להתרכו כמה שיורו.
- המשחק עוזר לשחקנים להתמודד על מכשולים ולהתגבר עליהם, ובמשחק שלנו השחקן (המטוס) לא ישר מט אם פגע בו טיל לרוב שאר המשחקים שהפגיעה הראשונה המשחק מסתים בכישלון בכך אנו מעבירים מסר לשחקנים שתמיד יש אפשרות שנייה.
- המשחק דורש מהשחקן חישוב כמה צעדים קדימה ולחשב מה עדיף לעשות ובכך יכול להיות שהשחקן ישתמש בדרישה הזאת בחיים ותמיד לחשב לפני לעשות משהו.
- במקרים הדו-מדדיים יכול השחקן הייתה מוגבלת ולבן התנועה במהלך התאפשרה בשמונה כיוונים עיקריים: למעלה, למטה, ימינה, שמאלה והאלכסונים שביניהם. לאחר שהחנעה הייתה במשיר, קשה מאוד היה ליצור במהלך תחווה מציאותית של עמוק,
- המשחק שלנו משחק תלת ממדי שמציאותי יותר ולבן התנועה הפכה אפשרית ביותר מ- 25 כיוונים שונים
- שימוש במתמטיקה ולפתח חשיבה מתחממתית ולדעת חוקי המתמטיקה.
- משחבי התלת ממד מספקים חוות ברמה אחרת, באיכות גבוהה ועם הרבה אתרי וענין.

- אסטרטגיות חשיבה פיתוח
- ניתוח והבנה של סביבה משתנה
- מהלכים תכנון
- שיפור הזיכרון
- קבלת החלטות יכולת

3. סקירת מצב קיימם בשוק אילו בעיות קיימות

קיימות גרסאות שונות של המשחק בשוק, בגרפיקה דו-ממדית ובגרפיקה תלת-ממדית אך המשחקים אלו כוללים את החוקים המקוריים.

במשחק שางן מפתחים הוסיפו חוקים חדשים שהשחקן נדרש להתמודד עם שני מכשולים לנוכח שם הזמן ופיגיות הטיל, גם כן המשחק שางן מפתחים יכול בתוכו משחק X ו O, בכך עשוינו שילוב בין שני משחקי המשחק אחד.

4. מה הפרויקט-Amor לחדר או לשפר

- משחק קל ונוח לשימוש.
- חוויות משחק למשתמש בעולם המדמה את המציאות.
- חוויות משחק תחרותי להבאה ופיתוח חשיבה וריכוז.

5. דרישות מערכת ופונקציונאליות

5.1 דרישת מערכת סביבת הטמעה ושימוש. שרידות, ביצועים\התמודדות עם עומסים.

סביבת הטמעה: מחשב עם מערכת הפעלה Windows .
שרידות: שמירה נתונים בקבצים, שמירה גיבוי לקבצי נתוני המשחק.
ביצועים: תצוגת המסך תתרען אחרי כל פעולה.

5.2 דרישות פונקציונאליות רשיימת דרישות מהמערכת, מיהן הפעולות בהן נדרשה המערכת לתמוך.

- * המערכת תאפשר להתחילה משחק חדש.
- * המערכת תשמור את נתוני המשחק לשחקן.
- * המערכת תאפשר לשחקן להזין את המטים.

6. בעיות צפויות במהלך הפיתוח ופתרונות (תפעוליות, טכנולוגיות, עומס ועוד):

טעויות בפונקציות החישוב והמתמטית כי המשחק דורש הרבה חשיבה.
לשימור על משחק בלי בגים ותקלות.

6.1 תיאור הבעיות הללו כפועל יוצא של דרישות המשמש מהתוכנה.

- * המשמש לא יודע איך לשחק.
- * חוסר תשומת לב במשחק.
- * חוסר ריכוז.

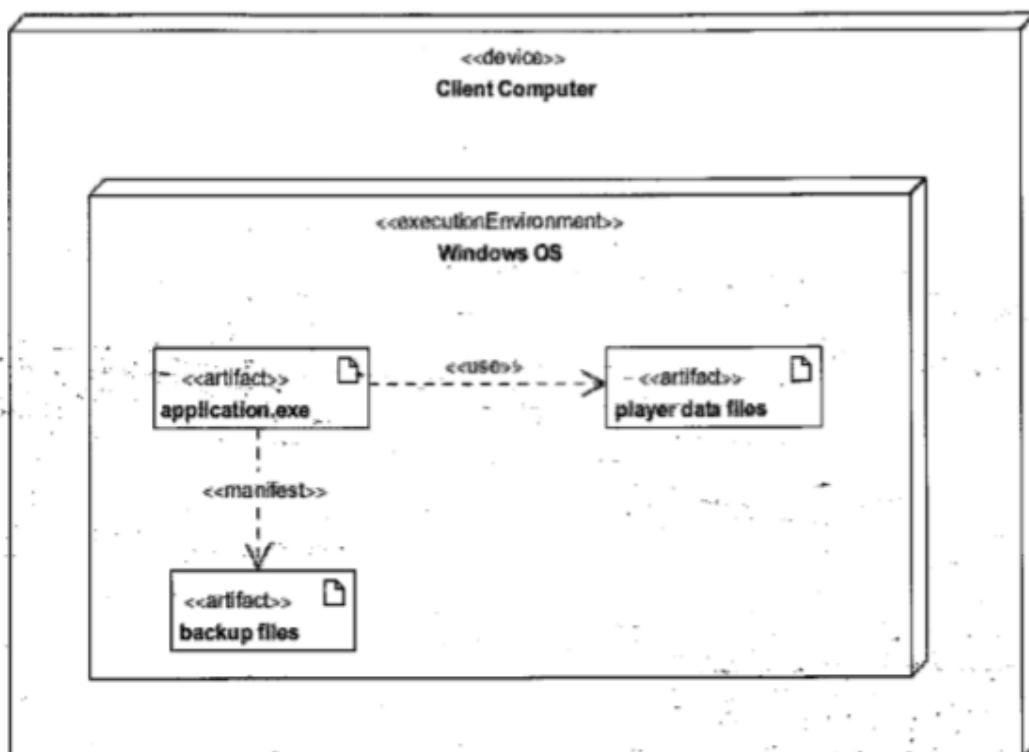
6.2 פתרונות אפשריים.(נא ציין פתרונות אפשריים וחלופות ארכיטקטוניות)

- * הדריכה נכונה אויר לשחק.
- * השחקן חייב להתרכו ולהזהר.

7. פתרון טכנולוגי נבחר :

7.1 טופולוגיה הפתרון

שמירת נתונים, שירותי גיבוי, המערכת תישם בסביבת Visual Studio, משחק - המערכת מורכב מ DB בעט קריסת המשחק, אם נדרש לשמור את הנתונים לצורך גיבוי אחריו הקriseה.



7.2 טכנולוגיות בשימוש

. תוכנה התומכת בתכנות שפת C למשל: Visual Studio .

7.3 שפות פיתוח (איוזה שפות ומדווע בכמה מילימ?)

C - כי נוח לתוכנת בה את החישובים של הפונקציות.

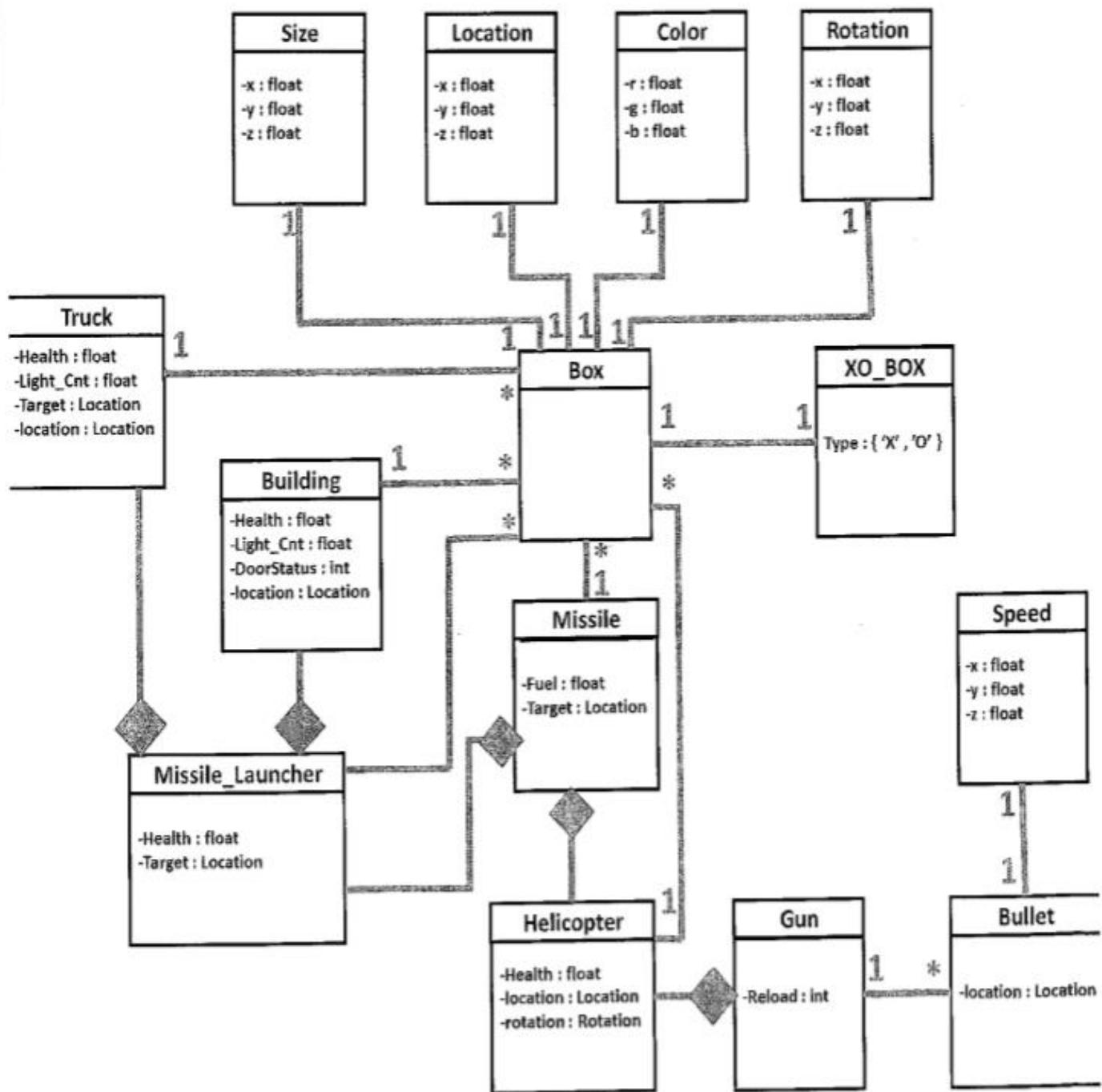
OPENGL - כי השפה הזאת נוחנת לנו לייצר גרפיקה תלת ממדית.

7.4 תיאור הארכיטקטורה הנבחרת - הסבר בכמה מילימ מדווע

המשחק הוא מערכת stand-alone שmoteknt על מחשב הלקוח.
השתמשנו בארכיטקטורה זה מכיוון שהמשחק הוא לשני שחקנים באותו מחשב.

7.5 חלוקה לתוכניות ומודולים

מבנה המערכת (structs)



7.6 סביבת השירות(מקומי, וירטואלי, ענן, שירות אינטרנט)

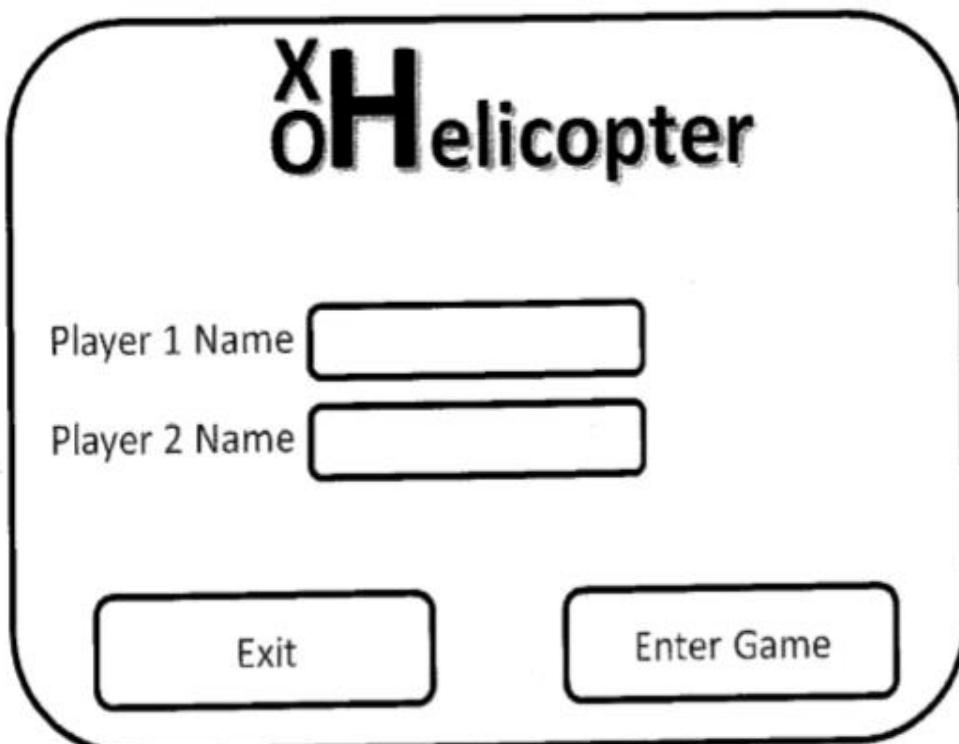
אין שרת.

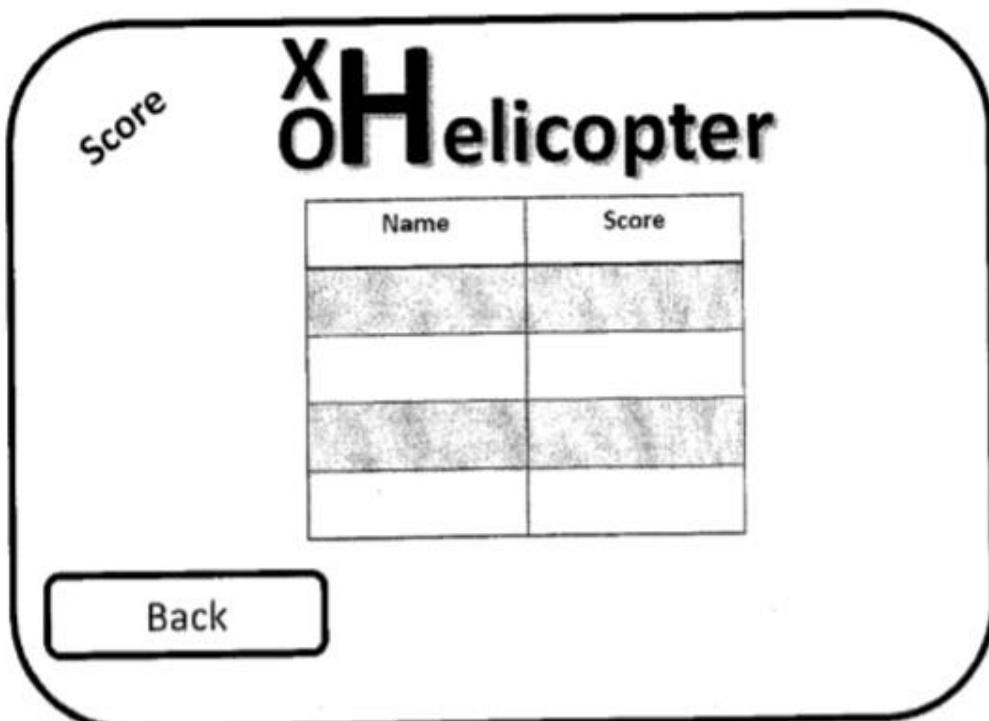
7.7 ממשק המשתמש/Lקוח-GUI

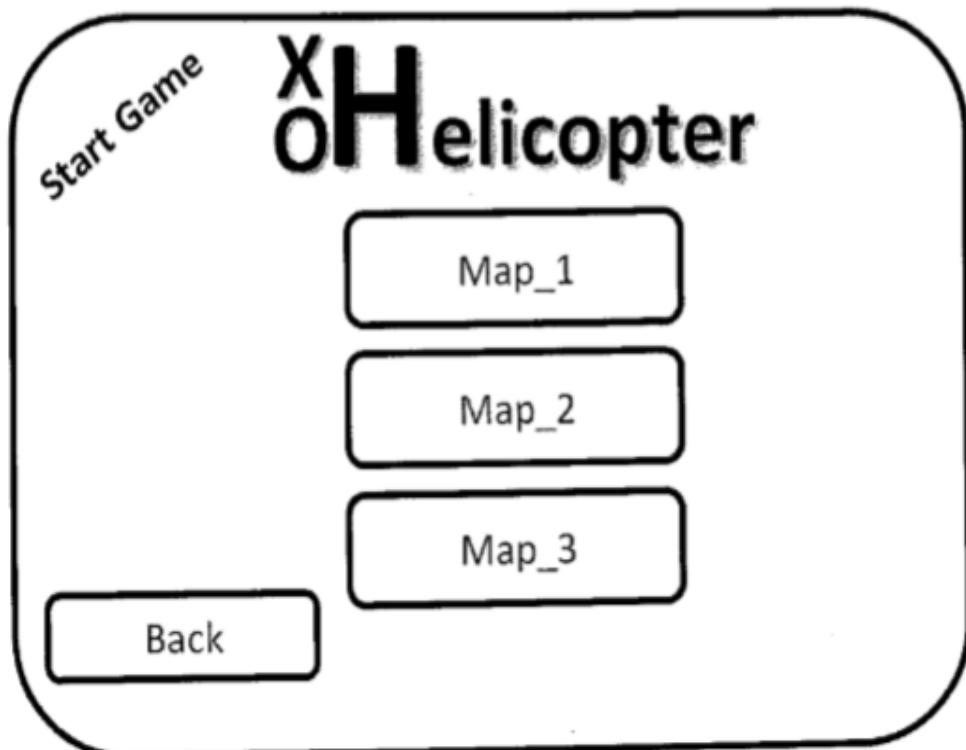
בהתחלת מסך של הכניסה למשחק שודרש לרשום את השם.

אחריו רישום יופיע על המסך שלושה לחיצנים:

1. לחץ העברה למסך שמוביל להנחות המשחק.
2. לחץ העברה למסך שמוביל טבלה של ציונים.
3. לחץ לתחילת המשחק.







7.8 משחקים למערכות אחרות / API:

אין משחקים למערכות אחרות

7.9 שימוש בחבילות תוכנה:

- ממשק גרפי GLUT
- ספריית מתמטיקה Math.h
- Exception.h
- חבילות קלט/פלט
- ספריית מחרוזות String.h
- ספריות סטנדרטיות stdlib.h, stdio.h

8. שימוש במבנה נתונים וארגון קבצים

8.1 נא פרט את מבני הנתונים

- Struct Size : מכיל גודל קופסה.
- Struct Color : מכיל צבע.
- Struct Rotation : מכיל רוטציה לשולשיה ציריהם.
- Struct Location : מכיל מיקום במרחב.
- Struct Box : מכיל ארבע המבנים Size,Color,Location,Rotation .
המבנה הזה הוא לשימוש בניה צורות שונים של אובייקטים.
 - : Struct Missile
 - מוגנה מכמה Box , מכיל מיקום שהיב להגעה אליו.
 - : Struct Missile_Launcher
 - מוגנה מכמה Box שתופס טיל.
 - : Struct Helicopter
 - מוגנה מכמה Struck Box מכילה מבנה של רוטציה ומבנה של מיקום משלו שהוחקן משתלת עלה.
 - : Struct Building
 - מוגנה מכמה Struck Box שהבניין זהה תופס Missile_Launcher
 - : Struck Truck
 - מוגנה מכמה Struck Box שהמסאית תופסת Missile_Launcher ועוקבת אחרי השחקן.

8.2 נא פרט את שיטת האחסון (מאגר, קבצים ובאיוזה טכנולוגיה)

קבצים:

- קובץ שמכיל את שמות השחקנים והציון של כל שחקן.
- קובץ לגביו נתונים משחק

8.3 נא ציין מנגנוני התאוששות מנפילה/קריסה/תמייה בטראנזיציות.

מנגנון גיבוי ושמיררת מידע.

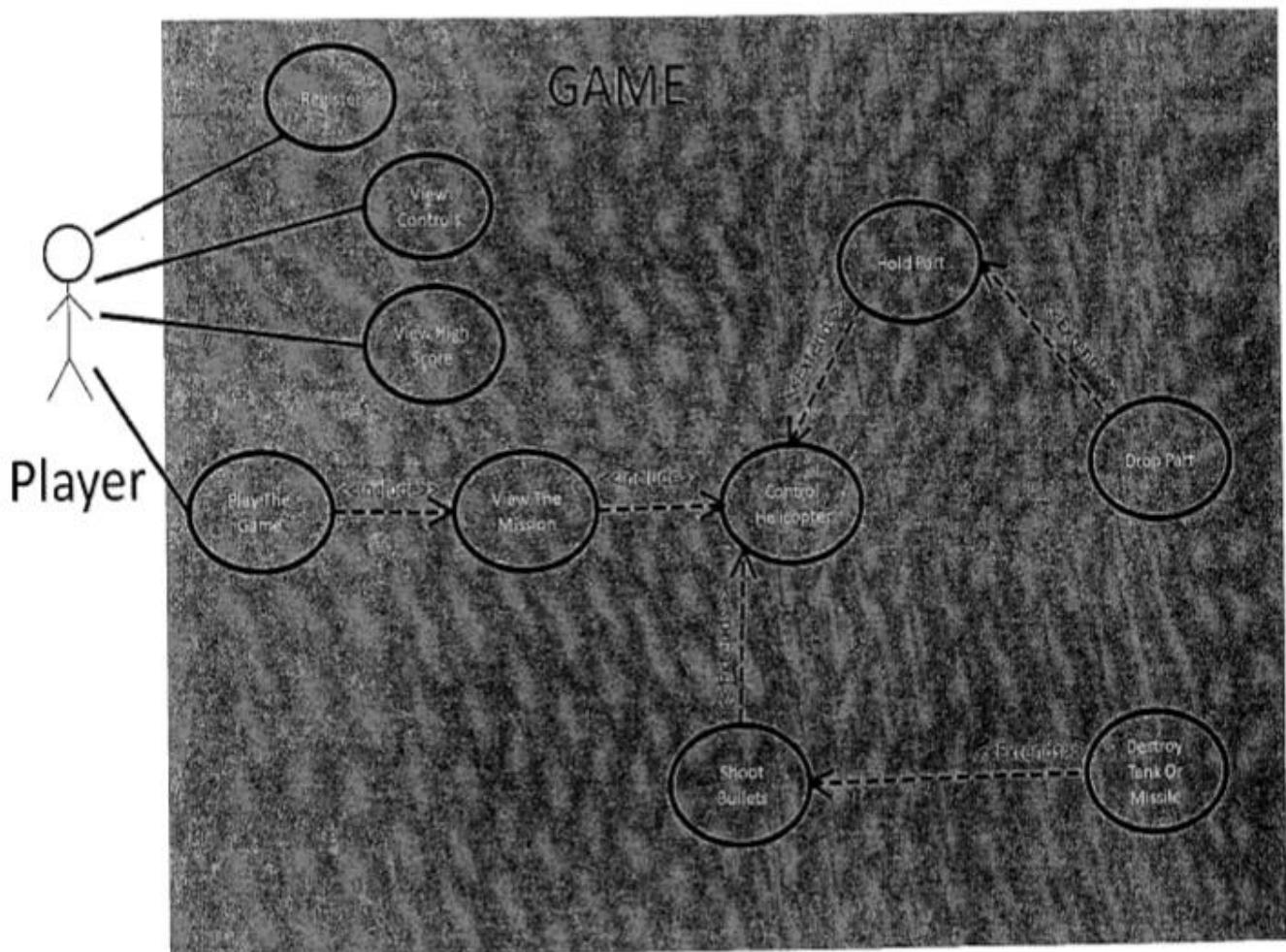
עבור כל פעולה שהשחקן מבצע הפעולה תיכתב לקובץ גיבוי טקסטואלי.

המערכת תעשה גיבוי באופן תדייר ותשמר אותו על מחשב השחקן. כך שבמידה של קריישה מכלול סיבה שהיא, הנתונים נשמרו בענן יכולו להיות גיבוי. ככל עת רק "משתמש" אחד יוכל לגשת לכל קובץ וכך למנוע מצב של קרייאת נתונים לא עדכניים..



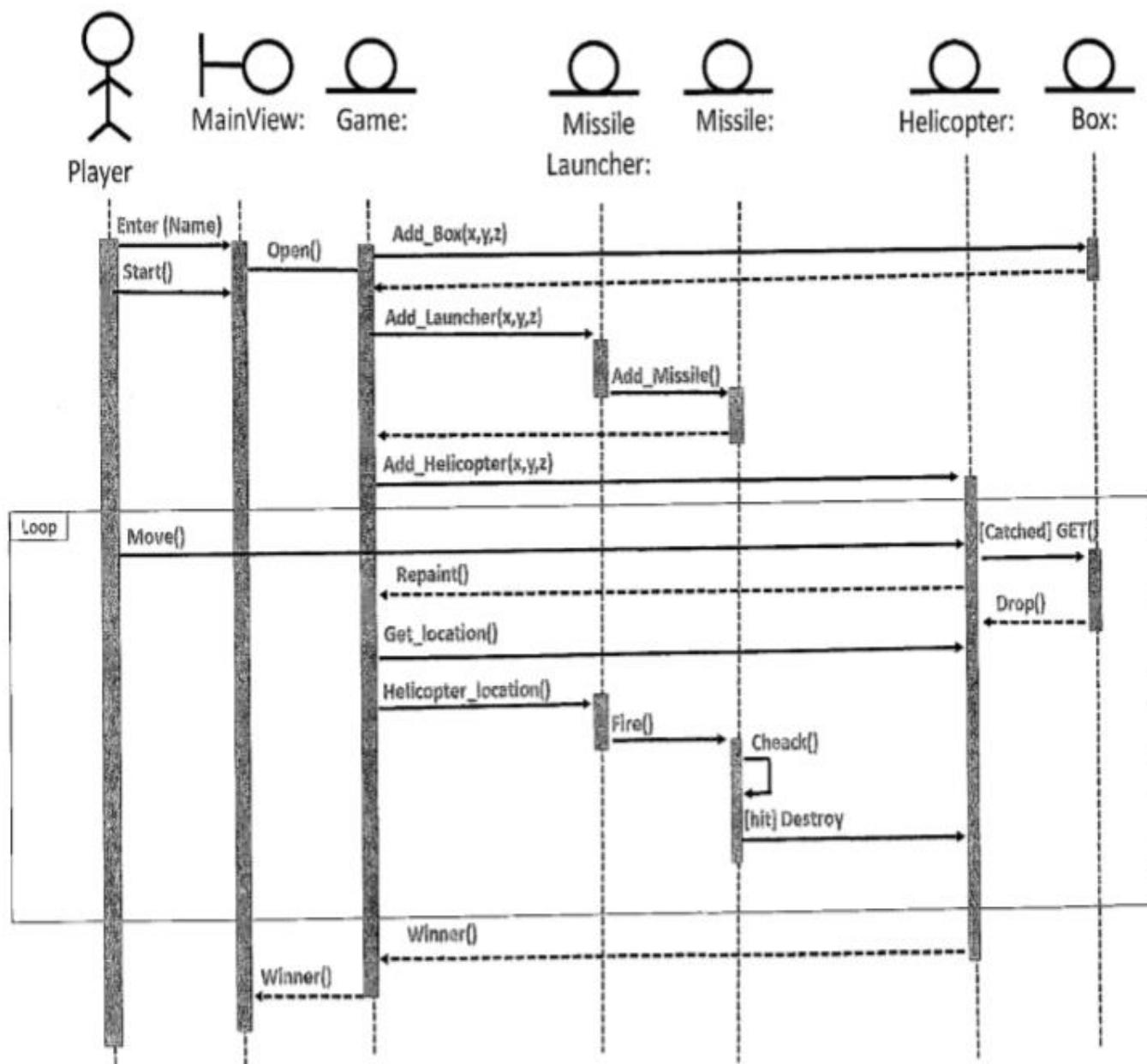
9. תרשימי מערכת מרכזיות

USE CASE 9.1

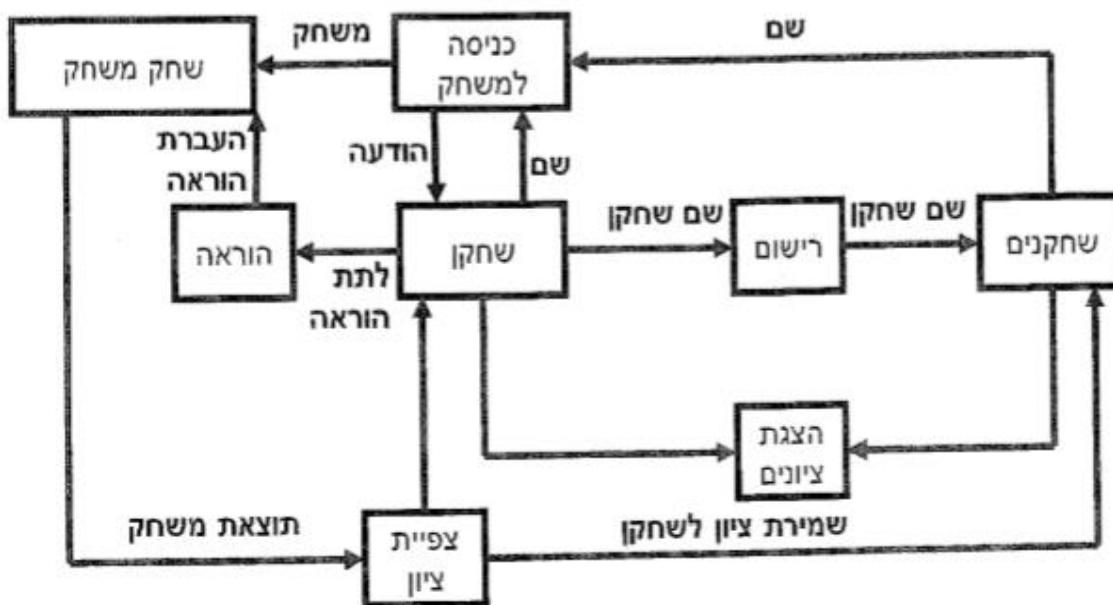


SEQUENCE DIAGRAM 9.2

- רצף קריאות פונקציות מרכזיות בלוגיקה העסקית המרכזית של הפרויקט



DATA FLOW 9.3



חישובי – 10. תיאור המרכיב האלגוריתמי

10.1 איזה בעיה בא לפטור, איך יפתר?

המשחק מתמקד בשיפור היכולת הקוגניטיבית של השחקן, השחקן צריך להיות מרוכז כל הזמן במהלך המשחק. נדרש לשמור על קואורדינציה עין-יד וגם לח奸ן צעדים קדימה לעבר המכשולים בהצלחה. המשחק משפר את היכישורים של השחקן בזמן המשחק ועזר לו להיות ממוקד כל הזמן. ההתמקדות במהלך משחק מונעת הטהות וועורת לשחקן לשמור על ערכנות כל הזמן. המשחק מסוג זה עוזרים לשחקנים לשפר את היכישורים

10.2 איסוף מידע וביתוחים סטטיסטיים (אבליטיקות)

חישובי טבלת הציונים ושאר הפרטים של המשחק לשחקן וערכונה בעת גמירתו
המשחק ושמירתם
בחור קובץ.

11. תיאור/התיחסות לנושאי אבטחת מידע

נא לציין אזורים הדורשים אבטחה, כגון: שרת, בקרת גישה לאתר, חשבונות משתמשים, מאגרי מידע וכיוצא ניתן מענה.

נא ציין מס' מקרים ותగובות להם ניתן מענה אבטחתי.

לצורך שמירת הנתונים:

הקבצים מוגבלים לתוכים פשוטים בלבד לצורכי אבטחה.

הקבצים יהיו ניתנים לפתיחת אופן ישיר ומועדים רק לкриאה, שברגע שהשחקן סיים את המשחק הוא יכול לדעת את הציון שקיבל וכמה זמן לקח לו לסיים המשחק.

12. משאבים הנדרשים לפROYIKT:

12.1 מס' שעות המוקדש לפROYIKT, חלוקת עבודה בין חברי הצוות

עד 200 שעות עבודה

12.2 ציוד נדרש

הציוד הנדרש לבנית הפROYIKT הוא:

-שני מחשבים אשר עובדים תחת מערכת הפעלה Windows (מחשב לכל שותף).

12.3 תוכנות נדרשות

Visual Studio

12.4 ידע חדש שנדרש ללמידה לצורך ביצוע הפROYIKT

פיתוח יותר בשפת OpenGL, נסחאות מתמטית לפונקציות שנדרשות במערכת.

12.5 ספרות ומקורות מידע

* ספרים לצורך בניה ועיצוב ב OpenGL

* אינטרנט ,Youtube ,Google)

13. תכנית עבודה ושלבים לימוש הפרויקט

שלב ראשון : בחירת בעיה אשר מעוררת צורך להקמת פרויקט טכנולוגי שמאפשר דרך ויטות טכנולוגיות חדשות להתמודד בצורה ייעילה ומסודרת.

שלב שני : חקירת הרעיון וניתוח השיטות והדרכים הייעלים ביותר לבניית הפרויקט שיכולים להוביל את הרעיון למציאות בצורה הכוי טובה והכוי מסודרת.

שלב שלישי : בניית תרשימי זרימה , תרשימים מחלקות (Class Diagram) וטבלאות הנדרשות לפרויקט , תרשימים הרצף (Sequence).

שלב רביעי : הגשת הצעת הפרויקט לאישור.

שלבים לאחר אישור ההצעה

שלב חמישי : התחלנו בחיפוש והקמת סביבות העבודה, מערכות, תוכנות והציגו הנדרש להתחולל בבניית הפרויקט הנ"ל.

שלב שישי : תחילת כתיבה ומימוש הרעיון הרצוי, וכל המבנים בתרשימים בשלב שלישי.

שלב שביעי : בדיקות תקינות והשלמת באגים .

שלב שמני : ייצור גרסה סופית של הפרויקט.

שלב תשיי: הגשת פרויקט.

14. תכנון הבדיקות שיבוצעו

14.1 נא פרט בטבלה, בדיקות תהליכיות בראשימת משתמש בהן נדרשה המערכת לעמוד (Full Flow).

מספר סידורי	פעולה	בדיקה	תוצאה רצiosa
1	כניסה למשחק	בדיקה אם השם קיים	כניסה או חזרה להרשמה
2	תזוזת המטוס לכיוון הנכון עם רוטציה מתאימה	בדיקה לאיזה כיוון המטוס משקיפה	תזוזת המטוס
3	גבולות המטוס במרוחך	בדיקה לאיזה כיוון המטוס משקיפה והמקום שלו	המטוס לא יוצא מגבול המרחב

14.2 נא פרט בטבלה, מס מיצג של בדיקות יחידה למודולים המרכזים בהן (unit test).

מספר סידורי	פשליה	בדיקה	תוצאה רצiosa
1	תזריז הטיל במעקב המטוס	בדיקות מיקום המטוס ביחס למיקום הטיל	תזריז הטיל לכיוון המטוס
2	חטzieה של הטיל כיוון המטוס	בדיקות מיקום המטוס ביחס למיקום הטיל ולחשב את הזווית המתאימה	ocabuteה הטיל כל הזמן כיוון המטוס
3.1	התנגשות הטיל המטוס	בדיקה אם מיקום הטיל באותו מקום של המטוס.	פיזוץ הטיל ולהוריד מהחיים של המטוס
3.2	התנגשות הטיל ברצפה או בבניין	בדיקה אם מיקום הטיל באותו מקום של הבניין או הרצפה.	פיזוץ הטיל
3.3	התנגשות התבלייטים ש המטוס הושקה אותם על הטנק	בדיקה אם מיקום התבלייטים באותו מקום של הטנק.	פיזוץ קטן ולהוריד מהחיים של הטנק
4	לאיזה כיבון הטנק ייר	בדיקות מיקום המטוס ביחס למיקום הטנק	תזריז הטנק לכיוון המטוס
5	התנגשות הטנק בבניין	בדיקה מאיזה כיוון התנגש ולתת לטנק ללכת לכיוון אחר	טנק יתרחק מהבנייהים

15. גרסאות בקרה (version control)

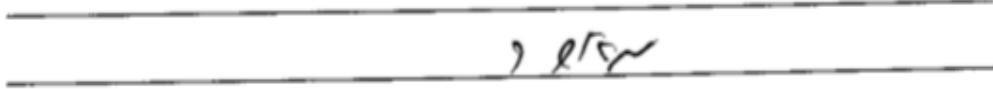
מספר גרסה	תיאור
1	בנייה האובייקטים.
2	בנייה פונקציית שהתקיף שלהם לעשות כל מנה פעולה שצריכים בין כל האובייקטים למשל : <ul style="list-style-type: none">• התנגשות בין אובייקטים.• חטzieiyut אובייקט ביחס לאובייקט אחר.• סרטוט לכל אובייקט.• חישוב דמיות לאובייקט מסוים.
3	בנייה המפה ולפזר עליה אובייקטים.
4	הגדרת מסימה.

אורט ברואודה
המכללה הטכנולוגית להנדסאים


חתימת המנהה האישי


חתימת הסטודנט

ג. הערות ראש המגמה במכלה



ד. אישור ראש המגמה

תאריך: 03.6.19 חתימה:  שם: שאדי נסאקייב
ראש המחלקה להנדסה
המגמה הבוגריה לתהום

שם: אורט ברואודה כרמייאל

ה. הערות הגורם המקצועני מטעם מה"ט

ו. אישור הגורם המקצועי מטעם מה"ט

תאריך: 26/6/19 חתימה:  שם: (ס/ן)



תוכן העניין:

פתחנו מערכת ב **OPENGL** הינה משחק תלת ממד שהוא שילוב בין שני משחקים. המשחק הראשון מסוק אמולטור שבנינו אותו לפי נקודת הרניה שלו שמשתלטים בה בכיוונים ובמהירות מתאימה. הוא נוע במרחב מסוים בכך להשיג את המשימות שדרושה ממנו. ושם אנו יכולים להציג את המשחק השני שרובהו מקרים אותו שהוא משחק איקס מיקס דרייקס בצורה שונה יותר כיפית ואטגרית.

משחק איקס מיקס דרייקס מסורתי השחקן שם את האקס או העיגול במשבצת המשחק עד שאחד מהשחקנים ינצח או המשבצת תהיה מלאה זהה. במשחק שפתחנו חידשנו את זה שיש תחרות משמעותית בין השחקנים שהם מוגבלים בכמה דברים כמו צרכים לנקחת האקס או עיגול מהמרחב לשים אותו לפני שהדלק יסתום או יפגע על ידי טילים.

במשחק יש אפשרות שהשחקן י משחק לבד או י משחק עם החבר שלו.

התכוון שהיה בתחלת פיתוח המשחק שיכיל שני שחקנים. על התקדמות בפיתוח שנינו את זה ש אם השחקן אין לא שותף לשחק הוא יכול גם כן לשחק עם המחשב. וזה לעת עוד אפשרות לשחקן.

המשחק מכיל 4 מפות שונות. כל מפה מכילה דרגת קושי שונה.

כל מפה מורכבת ממכלול ארכור שמכיל האובייקטים הבאים:
שני מסוקים/מסוק אחד.

משבצת המשחק 3X .

טנקים. כל טנק מכיל טיל.

רכבת מורכבת מ 4 קרטרים שרובה מכילות טיל.

בנינים ב 25 צורות שונות ויש-Calha מהם שמכילים טיל.

גשר.

בונוסים.

עכנים.

עצים.

គוכבים.

כבישים.



המסוק הינו המשחק שלנו אשר נע למרחב שמכיל מערכת היגוי כוללת הגה גובה (שלוט על התנועה במישור העילרוד). הגה כיוון (סבסוב) ומאזנות (גלאגול). תנועות המטים על ידי העכבר). במשחק ישנו שני משחקים אשר מחליף אחד לשני כדי למשחק (נקבע זמן מסוים).

דרישת המשחק לאוסף קופסאות למרחב המתאימות לסימן שנקבע לו מרגע הכניסה למשחק. יכול להיות בזמן אוסף הקופסאות הטנק מכון המסוק. אם הטיל פגע במסוק הקופה נופלת.

טכנולוגיות בשימוש :

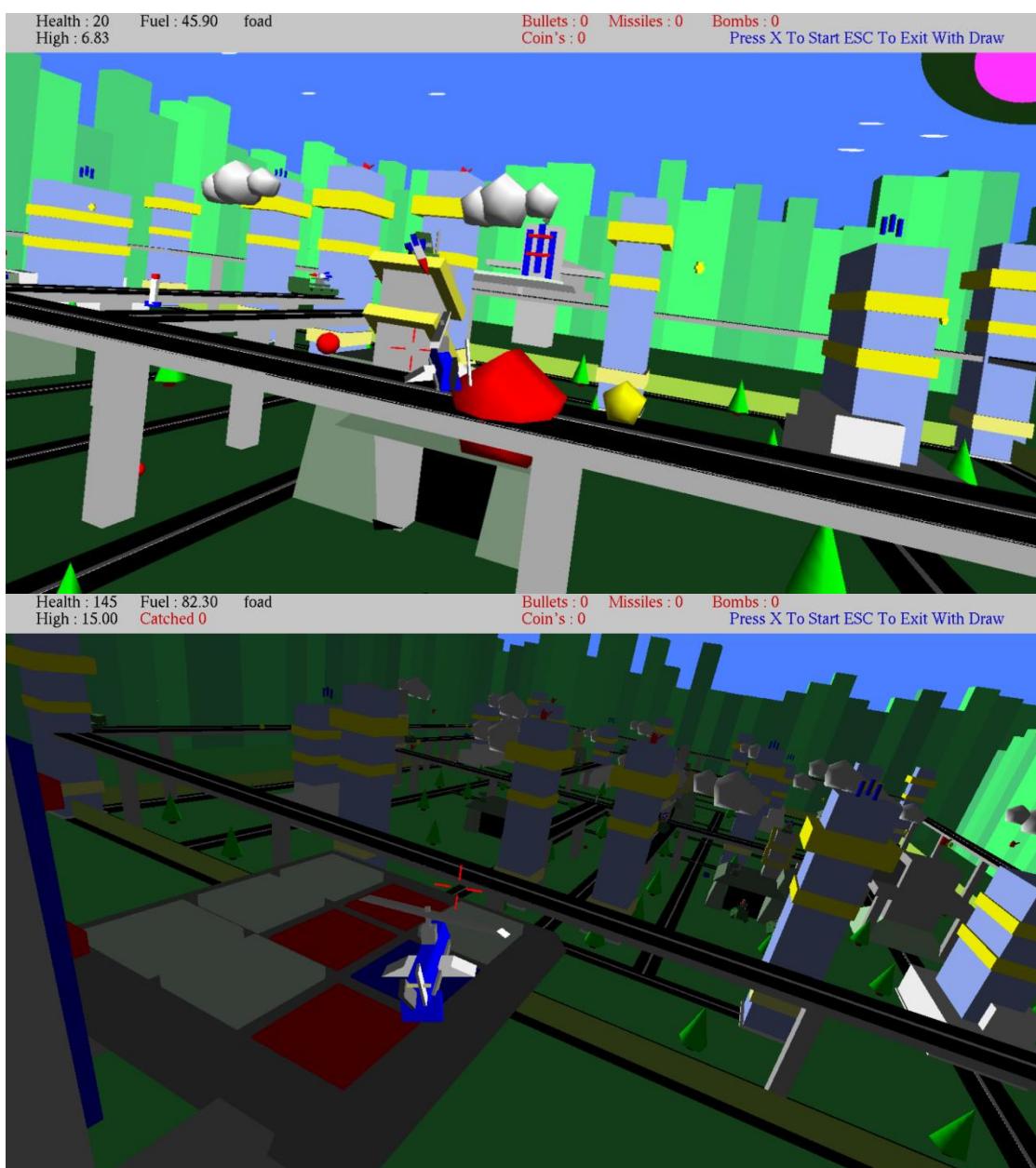
השתמשנו בטכנולוגיית **Visual studio 2012**

כי הינה תוכנה התומכת בתוכנות בשפת **C**. בחרנו לתוכנת בשפת **C** כי נוח לתוכנת בה את החישובים של הפונקציות.

גם כן בחרנו להשתמש בשפת **OPENGL** כי השפה הזאת נותנת לנו ליצור בגרפיקה תלת ממדית.

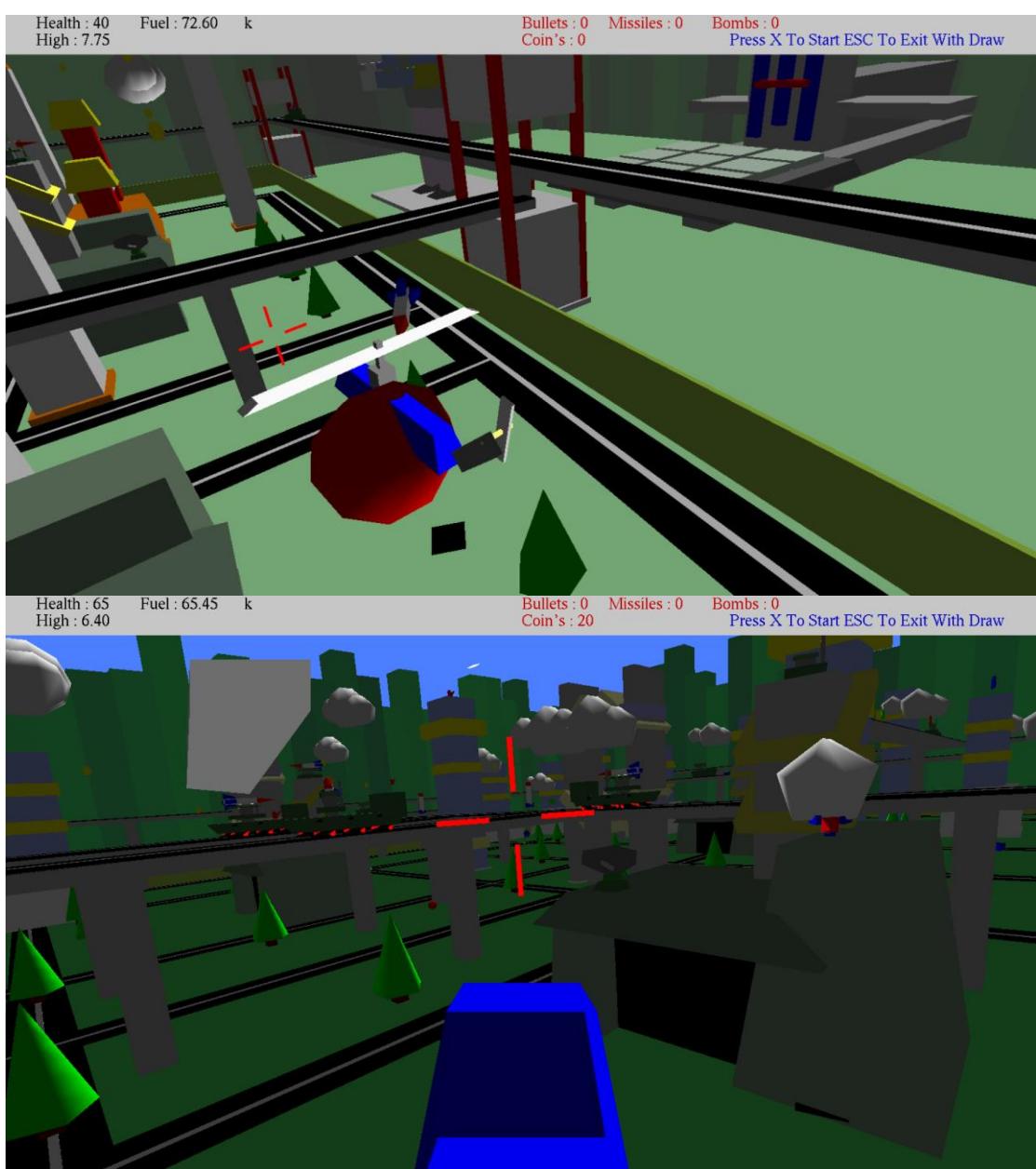


תמונות מהפרויקט:



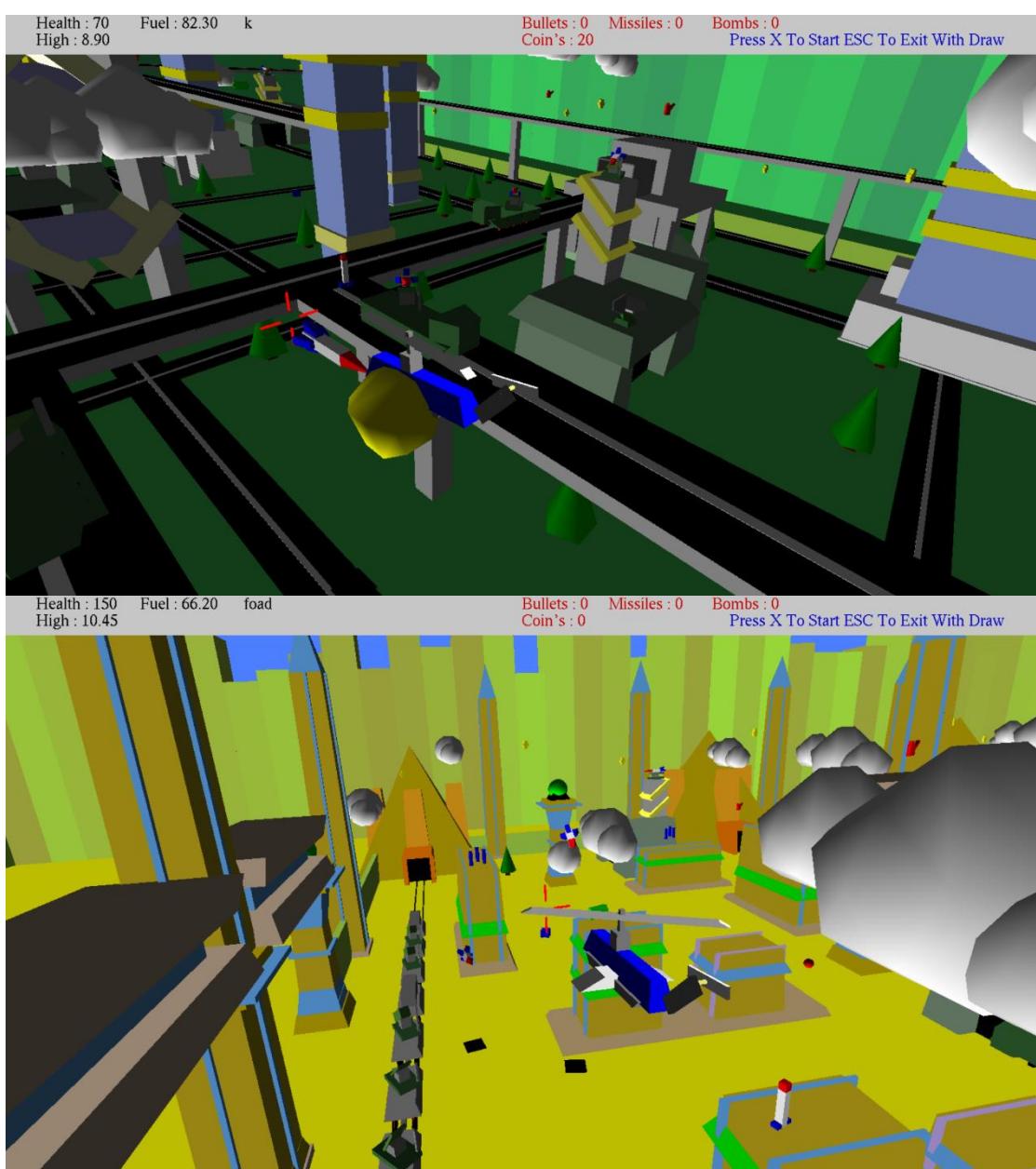
אורט ברואודה

המכללה הטכנולוגית להנדסאים



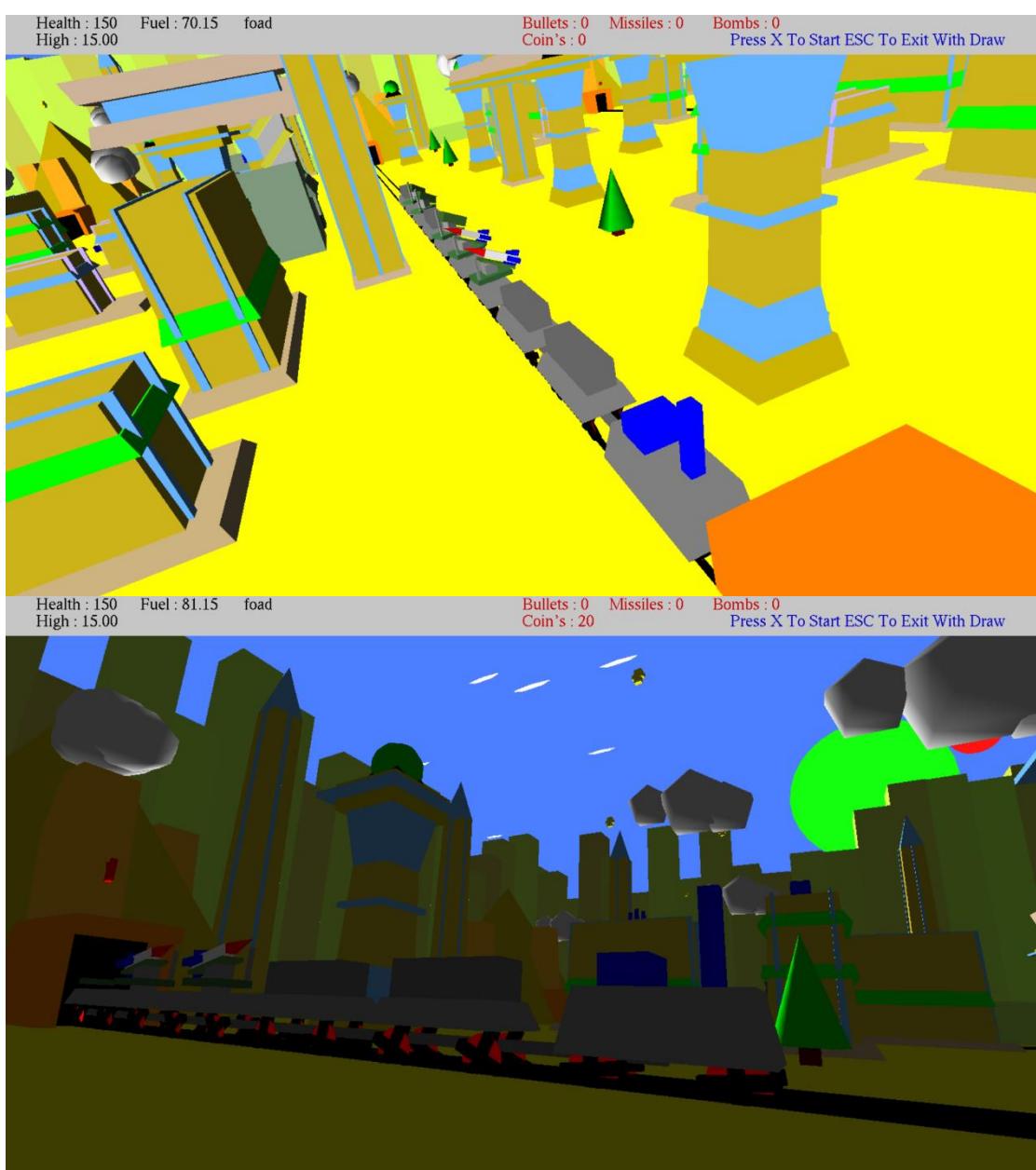
אורט ברואודה

המכללה הטכנולוגית להנדסאים



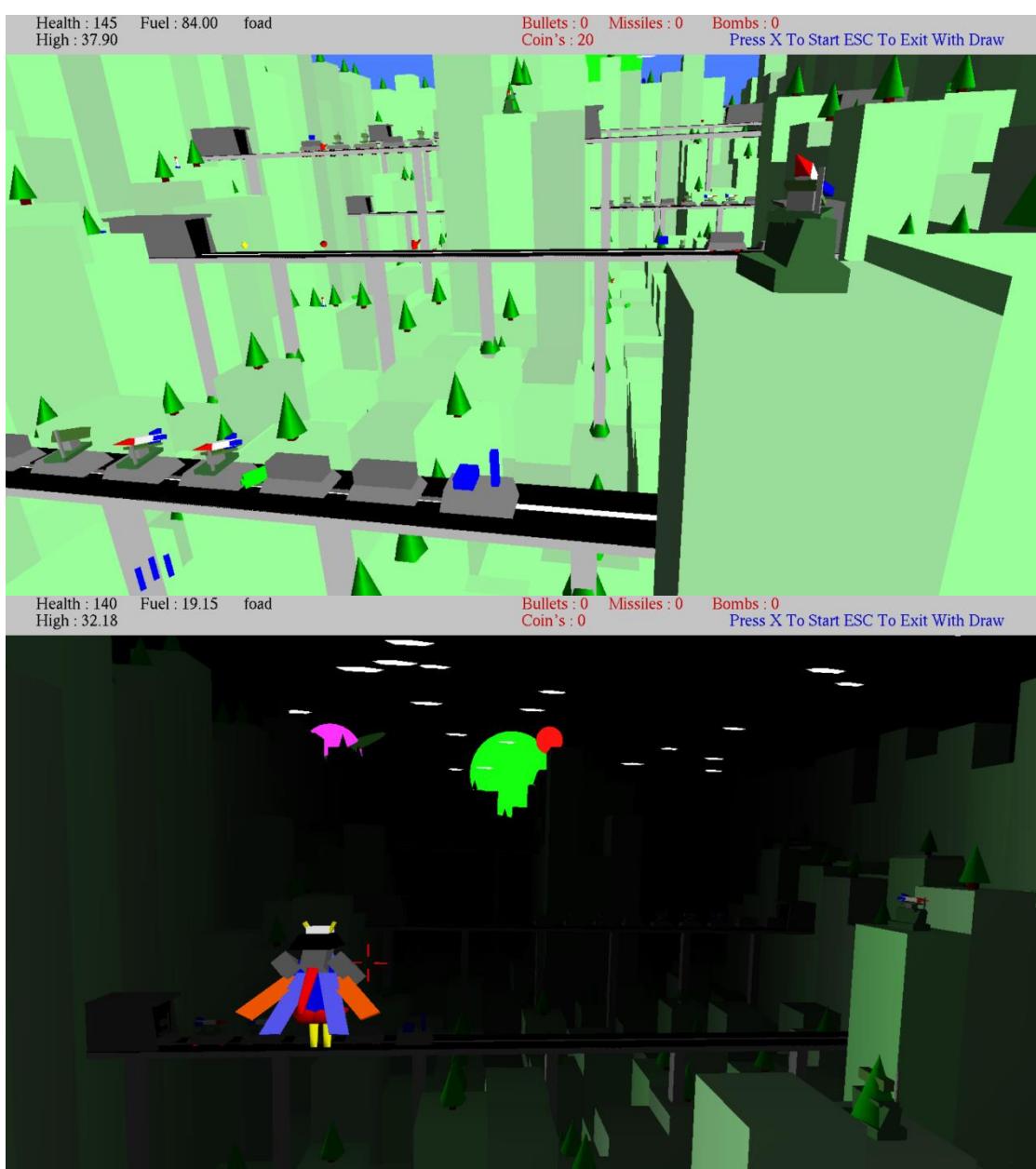
אורט ברואודה

המכללה הטכנולוגית להנדסאים



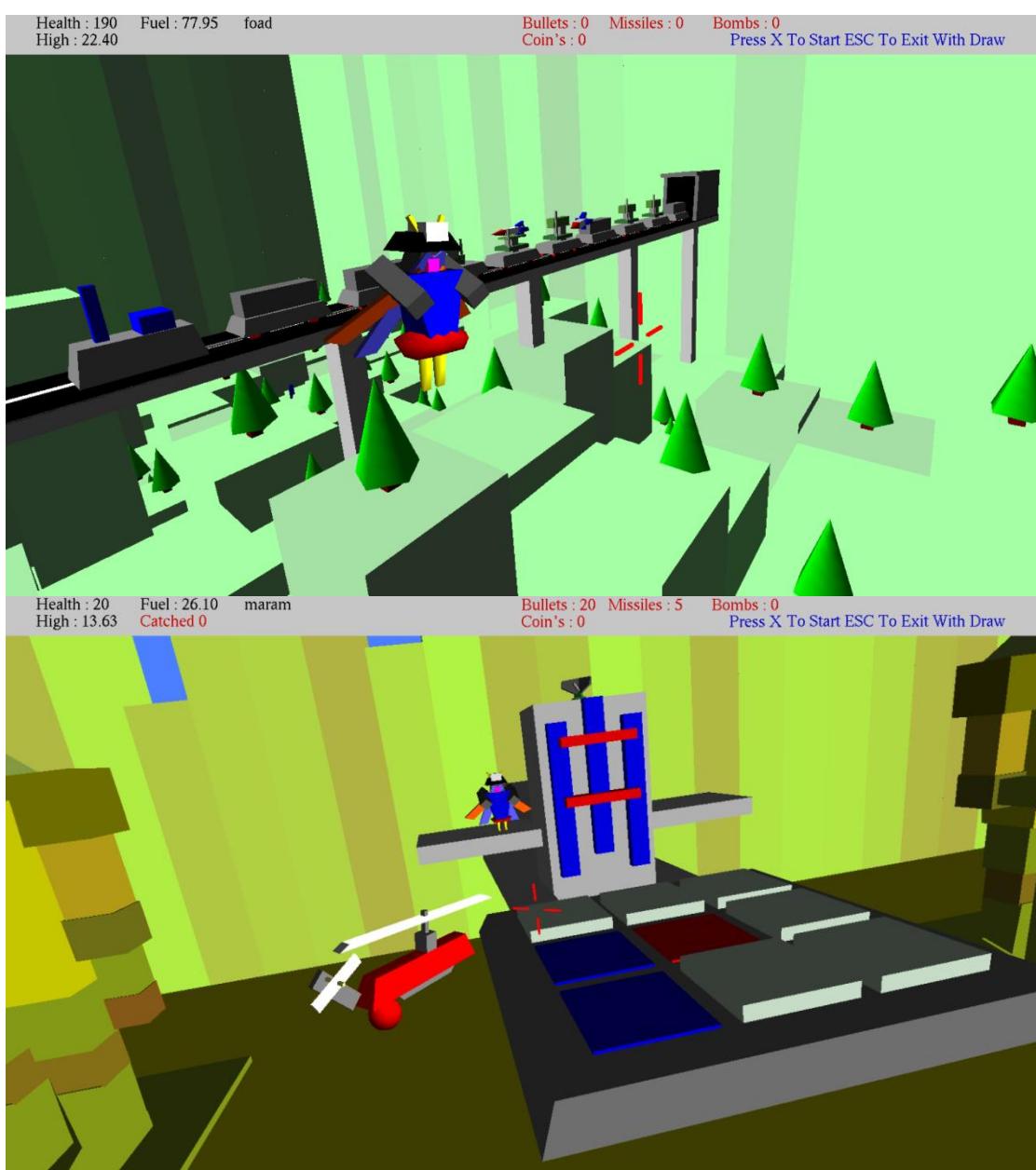
אורט ברואודה

המכללה הטכנולוגית להנדסאים



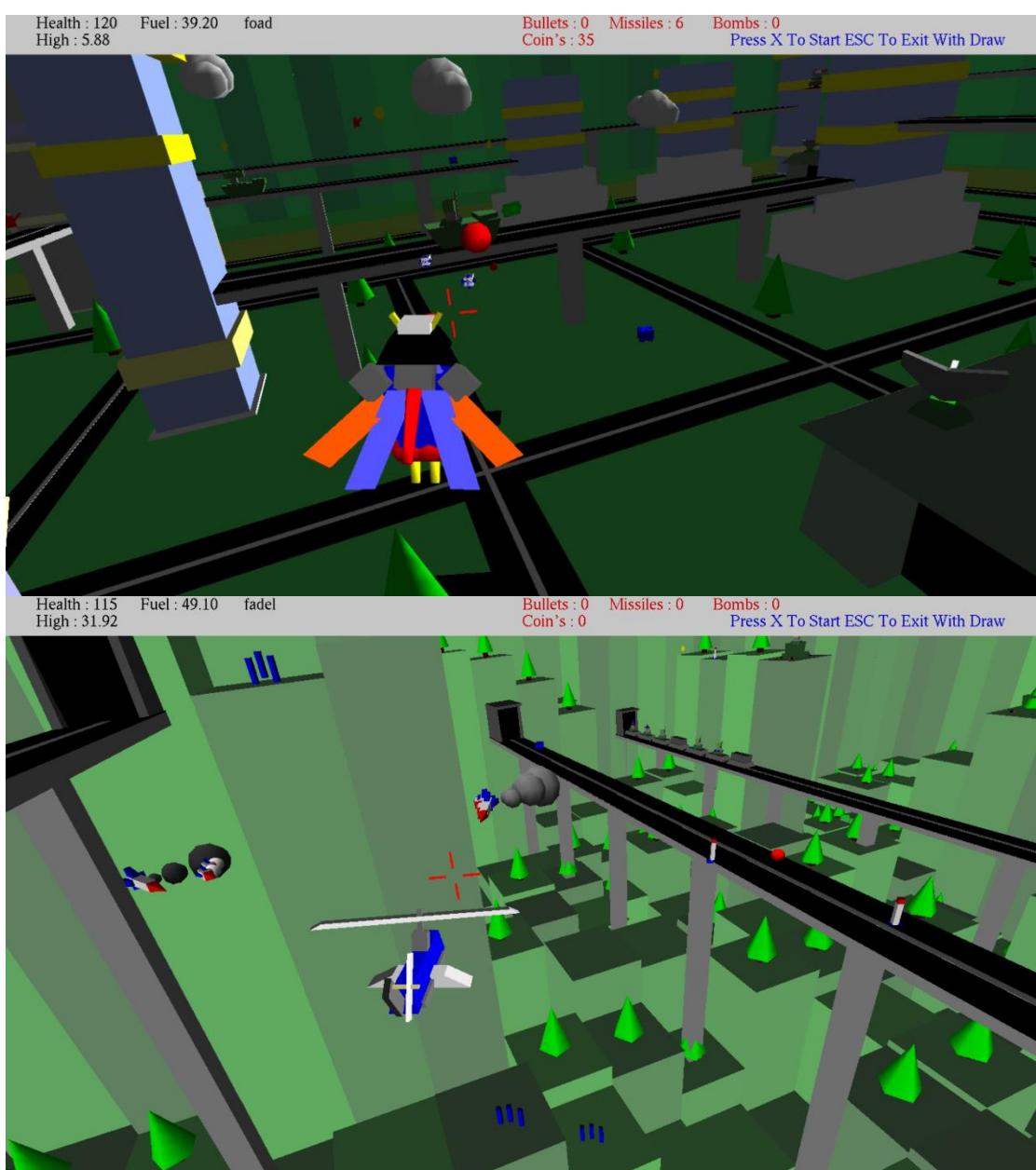
אורט ברואודה

המכללה הטכנולוגית להנדסאים



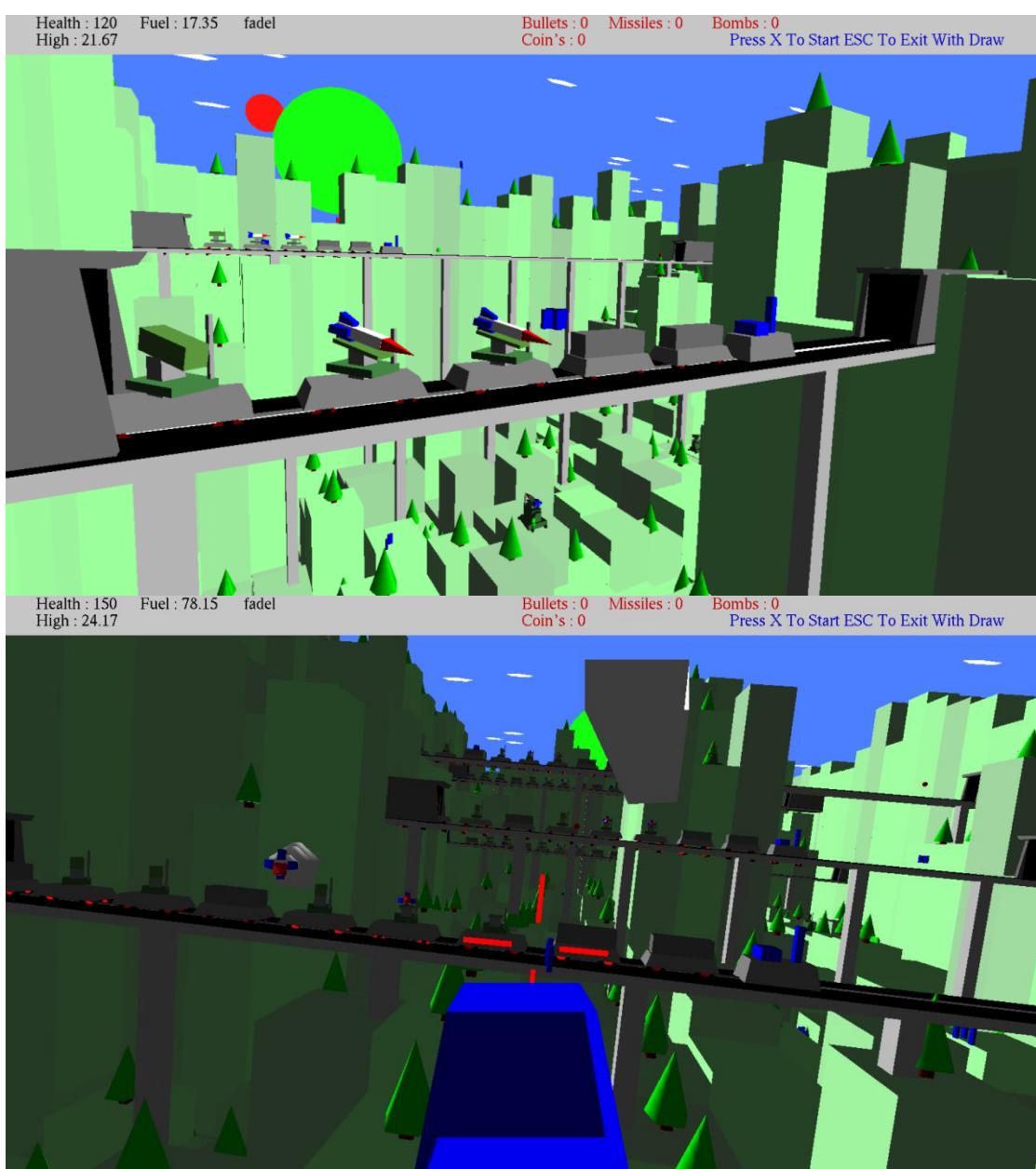
אורט ברואודה

המכללה הטכנולוגית להנדסאים



אורט ברואודה

המכללה הטכנולוגית להנדסאים



אורט ברואודה

המכללה הטכנולוגית להנדסאים



New User

1 Players
---VS Computer---

2 Players
---VS Player---

Exit

Player 1 :

Player 2:

foad

Password :

Password :

ChooseMap

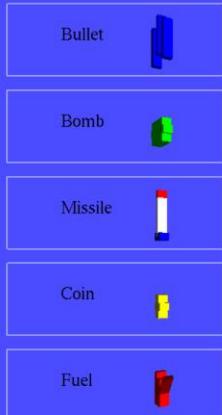
Details

ShowGames



אורט ברואודה

המכללה הטכנולוגית להנדסאים



Bullet	
Bomb	
Missile	
Coin	
Fuel	

Game Controls

W : Aim To Up	Space : Go Up	Shift + Q : Camera Up
S : Aim To Down	C : Go Down	Shift + E : Camera Down
D : Rotate Right	G : Throw XO	'T' : Camera Right
A : Rotate Left	Shift + W : Look Up	'T' : Camera Left
E : Look Right	Shift + S : Look Down	'+' : Camera In
Q : Look Left	Shift + D : Look Right	'-' : Camera Out
R : Shoot Bullets	Shift + A : Look Left	
F : Shoot Missiles		
V : Shoot Bombs		

Level 1 : Mini Helicopter
Health : 100
Shooting Bullets

Level 2 : Air Ship
Health : 150
Shooting Bullets
Shooting Missile

Level 3 : Air Monster
Health : 200
Shooting Bullets
Shooting Missile
Shooting Bombs

[Show_Objects](#)
[Back](#)

Game Rules :

You Have To Take The X Or O And To Throw It At The XO_Table Before The Fuel Ends

Important , If Some Missile Hit You, The X Or O Will Fall And You Will Have To Get It Again

And Maybe It Can Destroy Your Helicopter And You Lose Your Turn

You Can't Destroy : Bridge's , Pyramid , Train

You Can Destroy : Tanks , Building

While Playing You Can Collect Coins

Winner Takes 100% Of The Coins He Collected

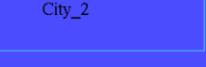
Loser Takes 50% Of The Coins He Collected



Mountains HARD



Villa



City_2



Egypt HARD



Day



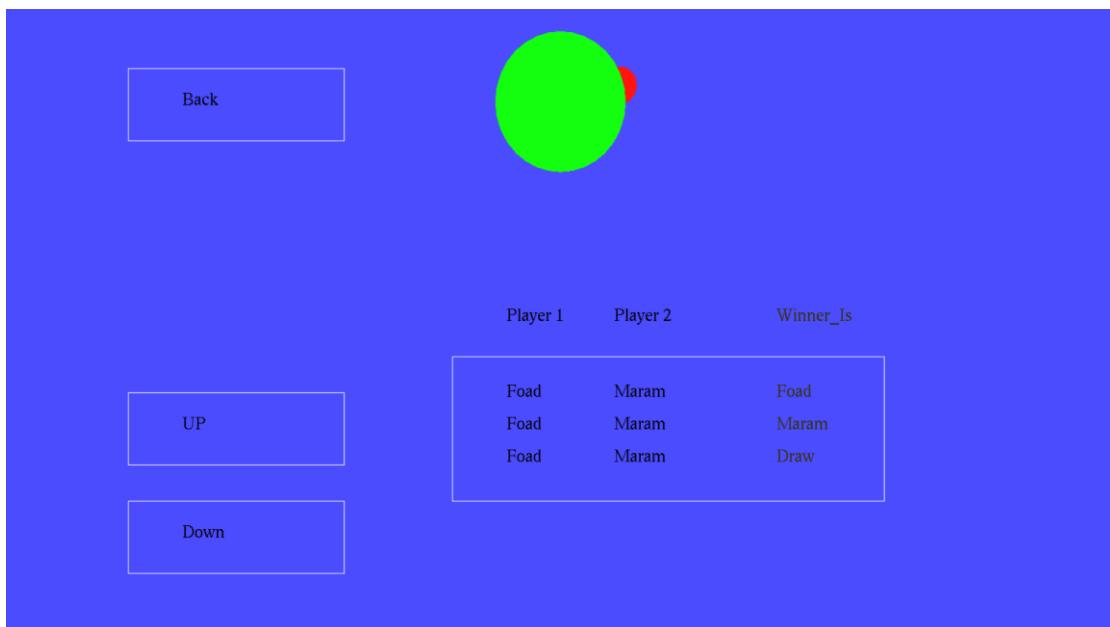
Store

[Back](#)

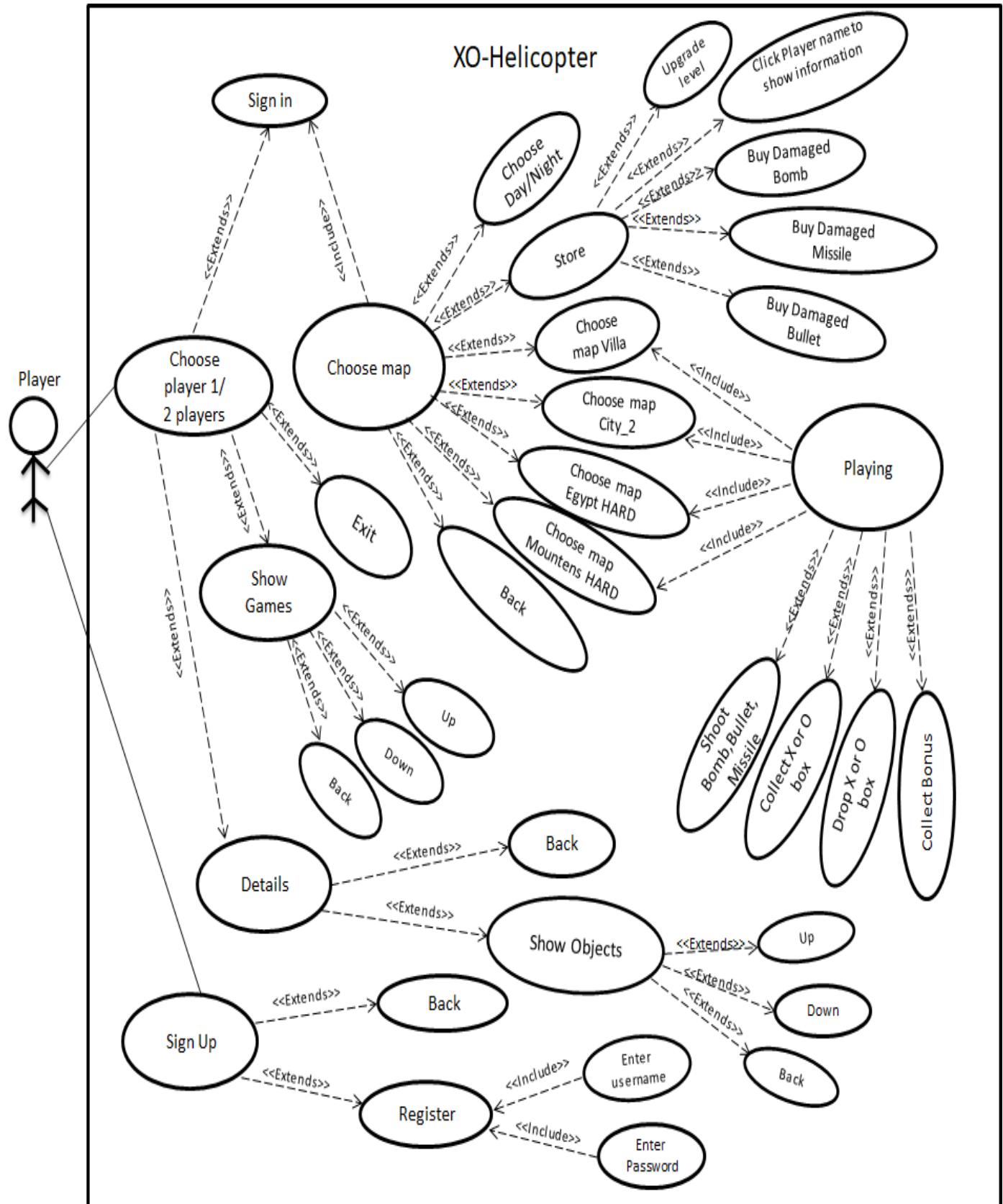


אורט ברואודה

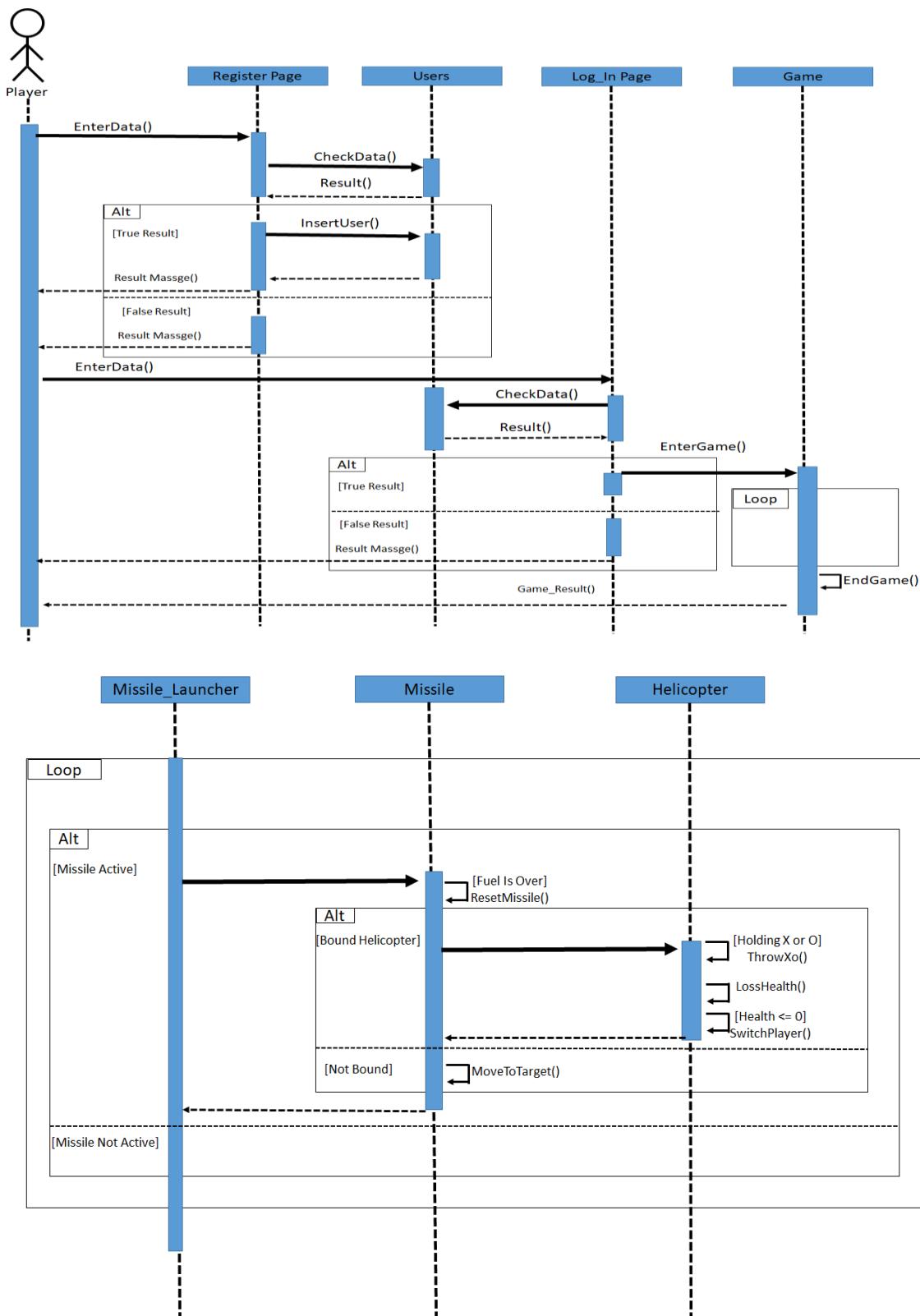
המכללה הטכנולוגית להנדסאים

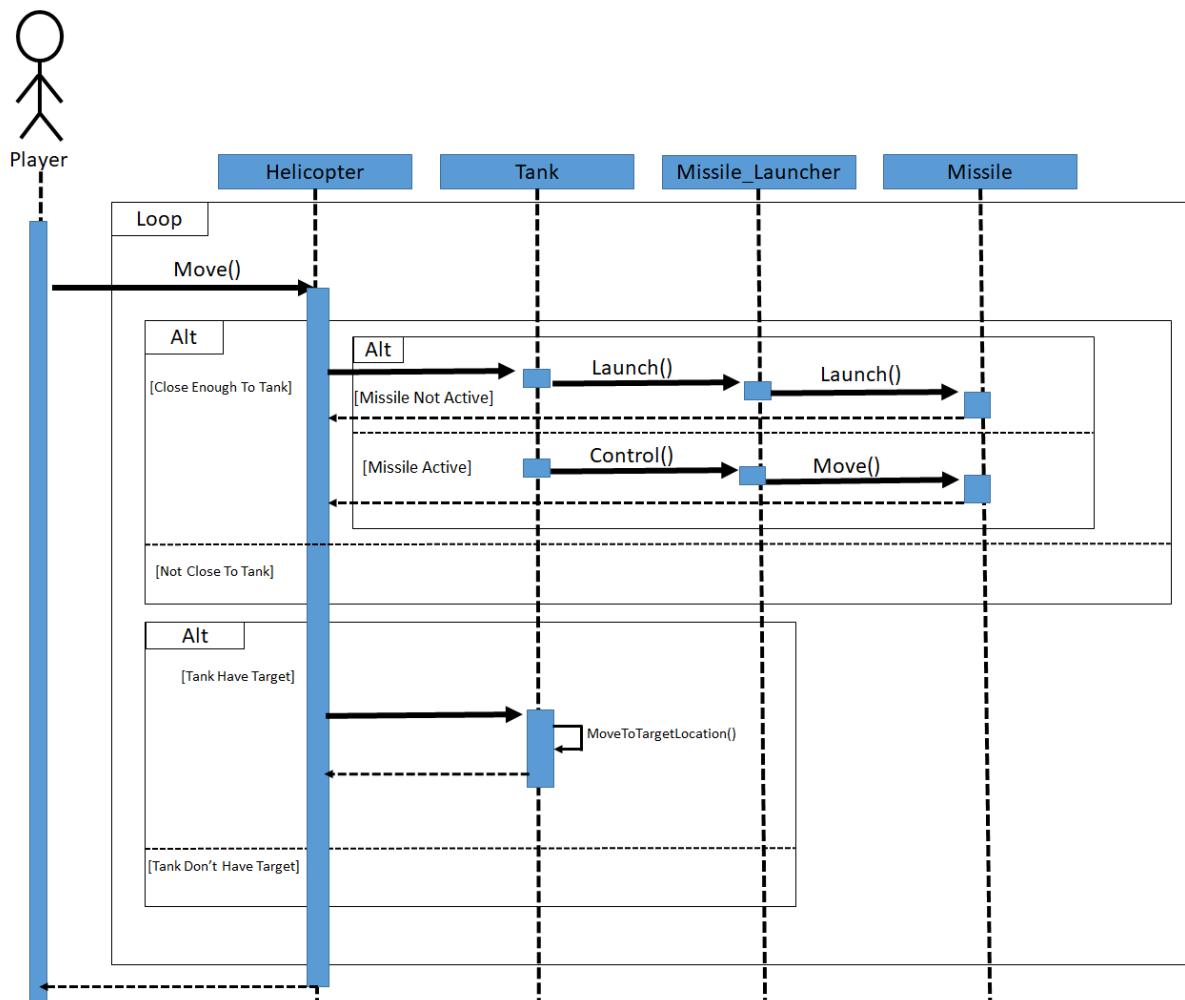


Use Case



Sequence Diagram:





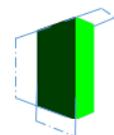
Structs:

```
typedef struct Color {           struct Color : מאפיין צבע
float r,g,b ;                 ;
}Color ;
```

```
typedef struct Rotation {        struct Rotation : מאפיין רוטציה לאובייקט מסוים :
int x,y,z ;                   ;
}Rotation ;
```

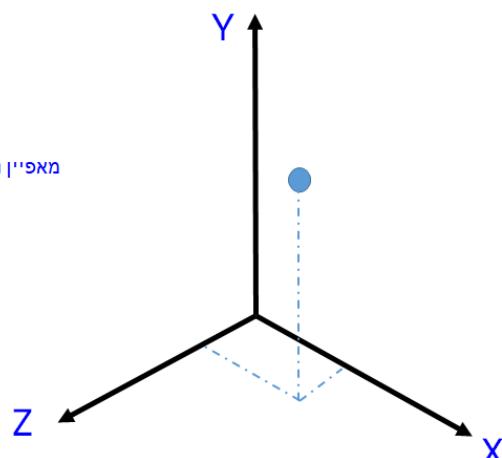


```
typedef struct Size {            struct Size : מאפיין גודל מסוים לבוקס מסוים
{                                ;
float x,y,z ;                 ;
}Size ;
```

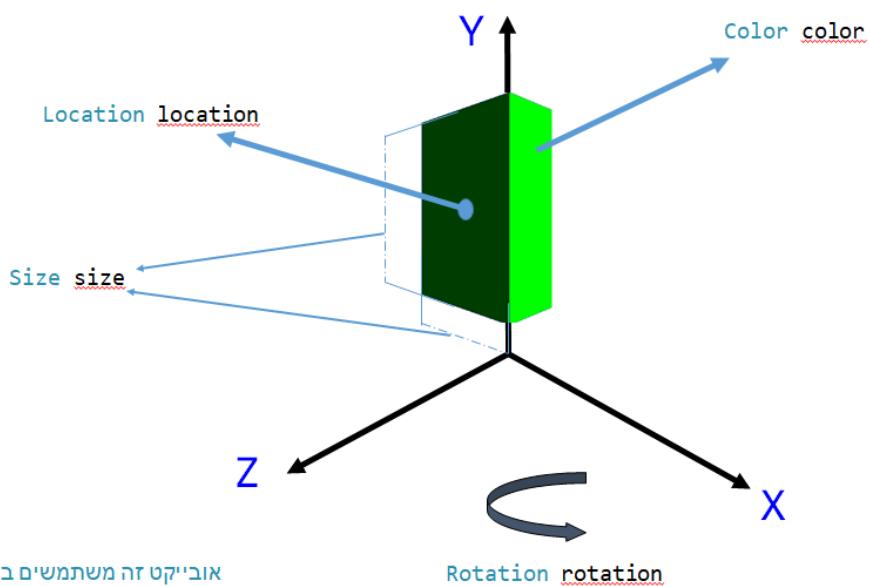


```
typedef struct Speed {          struct Size : מאפיין מהירות מסוימת לאובייקט מסוים
{                                ;
float x,y,z ;                 ;
}Speed ;
```

```
typedef struct Location {       struct Location : מאפיין נקודה מסוימת במרחב :
{                                ;
float x,y,z ;                 ;
}Location ;
```



```
typedef struct Box
{
Color color ;
Location location ;
Rotation rotation ;
Size size ;
}Box ;
```



`struct Box :`
אובייקט זה משתמש בו לבניית צורות שונות של אובייקטים.

```
typedef struct Tree
{
Location location ;
}Tree ;
```



`struct Tree :`
אובייקט המכיל רק מיקום במקומות אשר הוצאה:

```
typedef struct Star
{
Location location ;
}Star ;
```



`struct Star :`
אובייקט המכיל רק מיקום במקומות אשר הוצאה:

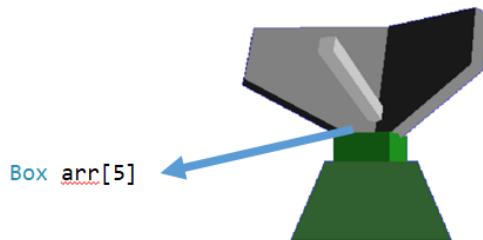
```
typedef struct Cloud
{
Location location ;
}Cloud ;
```



`struct Cloud :`
אובייקט המכיל רק מיקום במקומות אשר הוצאה:

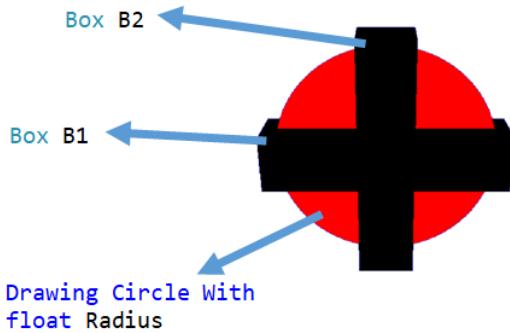
```
typedef struct Sata{
    Location location ;
    Rotation rotation ;
    Box arr[5] ;
}Sata ;
```

struct Sata :
אובייקט לעיצוב המכיל מיקום,רטציה וחמשה קוביות שמאתחלים בצורה זו:



Box arr[5]

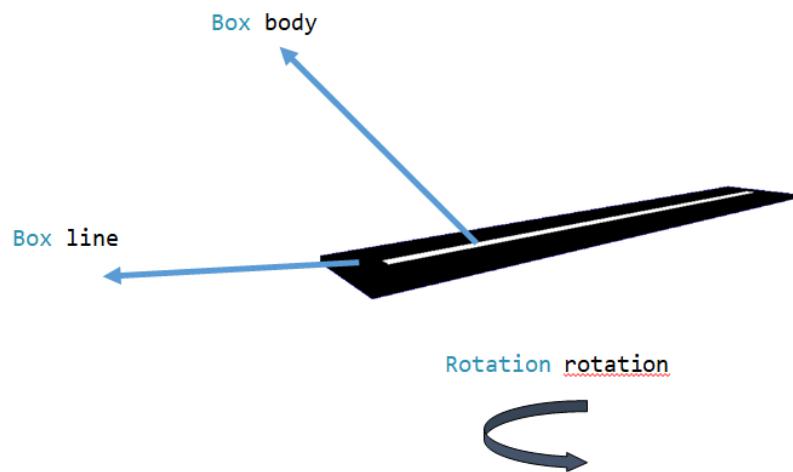
```
typedef struct Wheel
{
    Location location ;
    Rotation rotation ;
    Box B1,B2 ;
    float Radius ;
}Wheel ;
```



Drawing Circle With
float Radius

struct Wheel :
אובייקט לעיצוב המכיל מיקום,רטציה ושני קוביות שמאתחלים בצורה זו
משתמשים באובייקט זה ב TRAILER TANK .

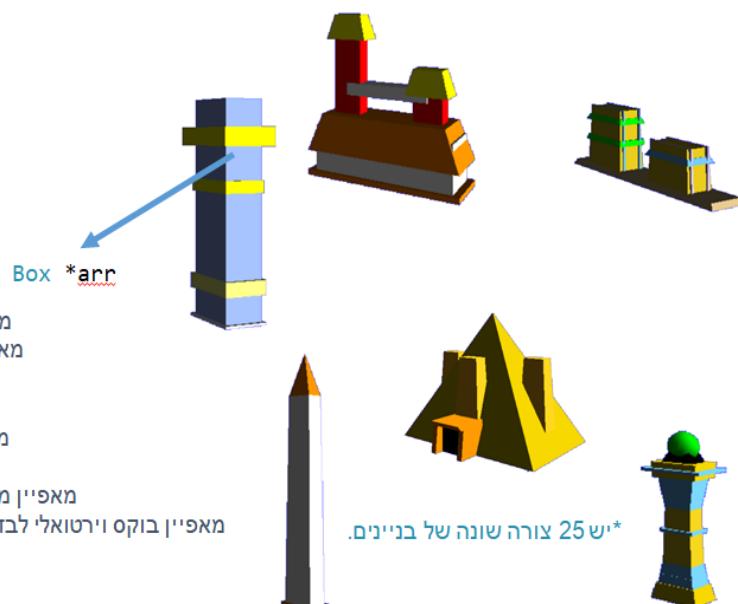
```
typedef struct Street{
    Location location ;
    Rotation rotation ;
    Box body ;
    Box line ;
}Street ;
```



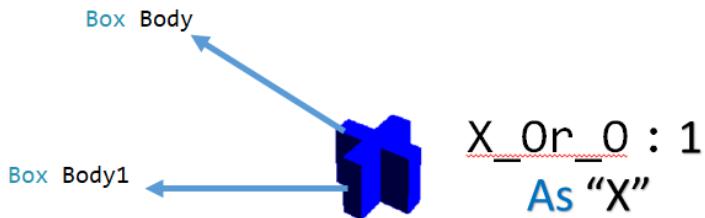
אובייקט לעיצוב המכיל מיקום,רטיחה ומספר קוביות שמאתחלים בצורה זו:

```
typedef struct Building
{
    Location location;
    Rotation rotation;
    int Health;
    int Type,Size,Shake;
    Box *arr;
    Box Tank_Bound;
}Building;
```

מאפיין מיקום הבניין :
 מאפיין רוטציה הבניין :
 מאפיין חיים הבניין :
 מאפיין סוג או צורת הבניין :
 מאפיין גודל מערכת הקוביות בבניין :
 מאפיין טולט הבניין אם נפל :
 מאפיין בוקס וירטואלי לבדיקת התנגשות עם טנק :
 מאפיין בוקס וירטואלי לבדיקת התנגשות עם טנק :

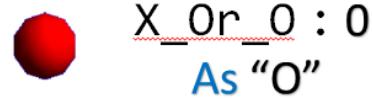


```
typedef struct X0
{
float MoveFlag ;
int Status ;
int X_Or_0 ;
Location location ;
Box Body ;
Box Body1 ;
}X0 ;
```



```
struct X0 :
```

אובייקט שמאפיין א 0 לפני צבע.
שחנקן עם צבע אדום יכול לחתך רק בוקס של צבע אדום.
שחנקן עם צבע כחול יכול לחתך רק בוקס של צבע כחול.



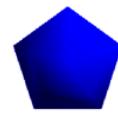
```
typedef struct Smoke
{
Location location ;
Color color ;
float Radius,Max_Radius,Speed ;
}Smoke ;
```



```
struct Smoke :
```

אובייקט שנמצא באובייקט סיל' שמאפיין עשן שייצא מהטיל.
וגם מאפיין פיצוץ מסוים.

Location location :
מאפיין מיקום העשן.
Color color :
מאפיין צבע העשן.
float Radius :
מאפיין הגודל הנוכחי של העשן.
float Max_Radius :
מאפיין הגודל המקסימלי של העשן.
float Speed :
מאפיין מהירות הגדלת העשן.



```
typedef struct Bullet
{
    float Damage ;
    int Status ;
    Location location ;
    Speed speed ;
    float Radius ;
    Smoke Blow ;
    Box Bound ;
}Bullet ;
```

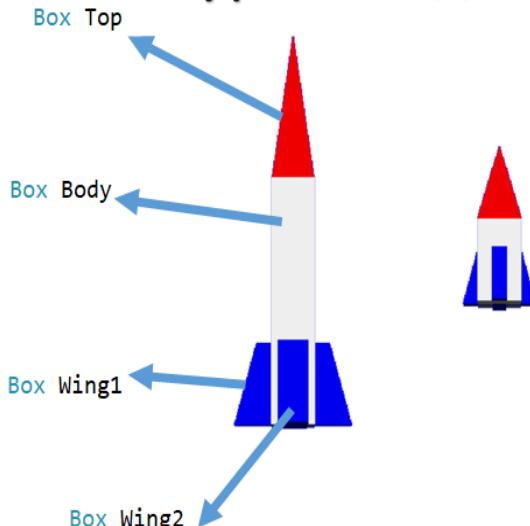
אובייקט שנמצא באובייקט המסוק שמאפיין את הקדרות.

float Damage :	מאפיין חזק הcadור.
int Status :	מאפיין מצב הcadור למשול = 0 לא פעיל = 1 הפעלה מיידית 0 > פעיל 0 = פצוץ וחזקת -100
Speed speed :	מאפיין מהירות הcadור ובאיזה זיהית.
Location location :	מאפיין מיקום הcadור.
float Radius :	מאפיין גודל הcadור.
Smoke Blow :	מאפיין פצוץ הcadור.
Box Bound :	מאפיין בוקס ורטואלי לצורך בדיקת התנגשות.

```
typedef struct Missile
{
```

```
int Type ;
int status ;
int fuel ;
float Damage ;
Location location ;
Location *Launch_Location ;
Location *target ;
Helicopter *Helicopter_target ;
Rotation rotation ;
Speed speed ;
Speed Maxspeed ;
Box Body ;
Box Top ;
Box Wing1,Wing2 ;
Smoke smoke[4] ;
Smoke Blow ;
}Missile ;
```

Type 0 Type 1



```
struct Missile :
```

הטיל שנמצא באובייקט **Missile_Launcher** שהשימוש שלו כמכשול שיעורק אחרי השחקן ומנסה להכשיל אותו.
וגם השחקן יכול לשימוש באובייקט זה להגן על עצמו.

```
struct Missile :
```

- אובייקט הטיל יש ממנו שני סוגי:
 1. משגר טילים מהירים
 2. משגר טילים איטיים

התפקיד שלו לעקוב אחר המ██וק (השחקן) ולכשיל אותו.

```
int Type ;  

int status ;  

int fuel ;  

float Damage ;  

Location location ;  

Location *Launch_Location ;  

Location *target ;  

Helicopter *Helicopter_target ;  

Rotation rotation ;  

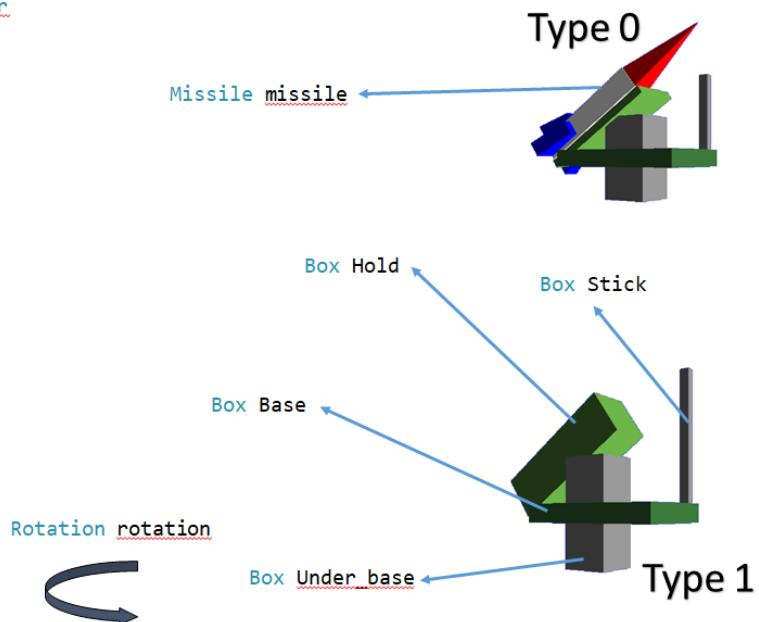
Speed speed ;  

Speed Maxspeed ;  

Smoke smoke[4] ;  

Smoke Blow ;
```

```
typedef struct Missile_Launcher  
{  
int Type ;  
int *TankRotation ;  
Location location ;  
Location *target ;  
Location *launch_Location ;  
Trailer *trailer ;  
Rotation rotation ;  
Missile missile ;  
Box Base ;  
Box Hold ;  
Box Stick ;  
Box Under_base ;  
}Missile_Launcher ;
```



struct Missile_Launcher :

אובייקט משגר הטילים יש ממנו שני סוגי :

1. משגר טילים מהירים
2. משגר טילים איטיים

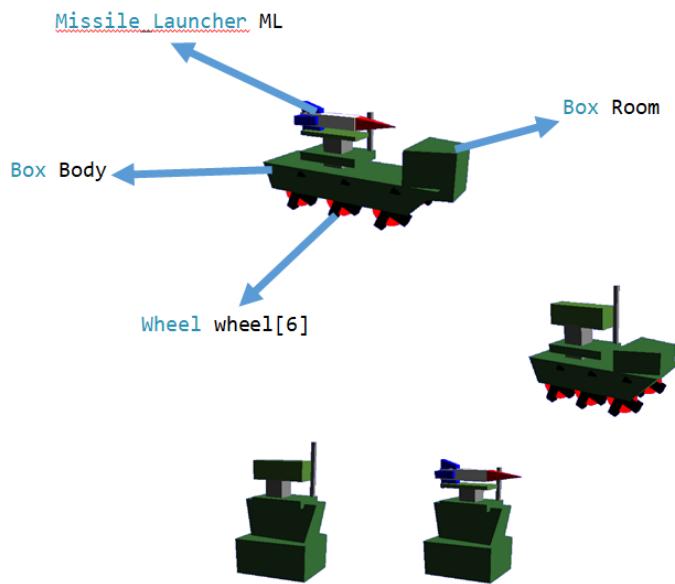
אובייקט זה מורכב על קמה אובייקטים אחרים :

1. Trailer (קרוואן)
2. Tank (טנק)
3. Building (בניין)

שהתפקיד שלו לשגר את הטיל ולנהל אותו.

```
int Type ;               מאפיין סוג משגר טילים
int *TankRotation ;      מאפיין רוטציית הטנק (אם מורכב על טנק)
Location location ;      מאפיין מקום משגר טילים
Location *target ;       מאפיין מיקום המטרה שעוקב אחריו
Location *Launch_Location ; מאפיין נקודת שיגור
Trailer *trailer ;      מאפיין קרוואן (אם מורכב על קרוואן)
Rotation rotation ;     מאפיין רוטציית משגר טילים
Missile missile ;       מאפיין הטיל במשגר טילים
```

```
typedef struct Tank
{
    int Health ;
    int Type ;
    int Status ;
    int Maxspeed ;
    int Start_Tank ;
    Smoke Blow ;
    Location location_1,location_2 ;
    Location location ;
    Location *target ;
    Rotation rotation ;
    Speed speed ;
    Wheel wheel[6] ;
    Box Body ;
    Box Room ;
    Missile_Launcher ML ;
    Box Missile_Area ;
    Box Tank_Bound_Area ;
}Tank ;
```



```
struct Tank :
```

אובייקט טנק יש קמה סוגים ממנו
 1. עוקוב אחר המטוס (משחקון)
 2. נע בין שתי נקודות
 3. סטטי (לש דז)
 המטרה שלו לשגר טילים על המטוס (משחקון) ולהכשיל אותו

```
int Health : מאפיין חיים הטנק :  

int Type : מאפיין סוג הטנק :  

int Status : מאפיין מצב הטנק :  

int Maxspeed : מאפיין מהירות המטוס מילית של טנק :  

int Start_Tank : מאפיין תנועה מסויימת (יציאה מהמבנה) :  

Smoke Blow : מאפיין פיצוץ הטנק :  

Location location_1,location_2 : מאפיין שני מקומות שהטנק נubbyינהם :  

Location location : מאפיין מיקום הטנק :  

Location *target : מאפיין מיקום המטרה שעקב אחריו :  

Rotation rotation : מאפיין רוטציית הטנק :  

Speed speed : מאפיין מהירות הטנק ובאייה זווית :  

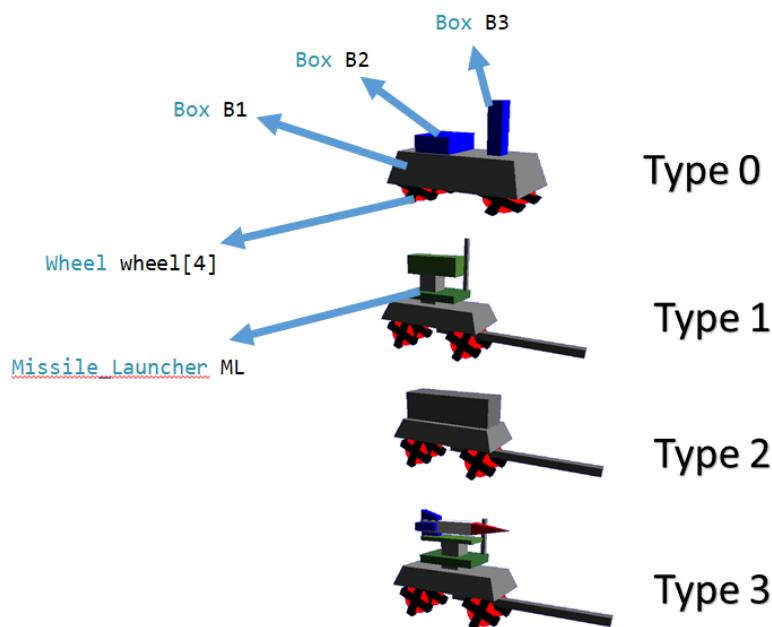
Wheel wheel[6] : מאפיין מהירות הטנק ובאייה זווית : [ ]  

Missile_Launcher ML : מאפיין משגר טילים :  

Box Missile_Area : מאפיין שטח התקרכות המטוס לטנק :  

Box Tank_Bound_Area : מאפיין שטח התגשות הטנק :
```

```
typedef struct Trailer
{
int Type ;
Location location ;
Rotation rotation ;
Box B1,B2,B3 ;
Box Missile_Area ;
Wheel wheel[4] ;
Missile_Launcher ML ;
}Trailer ;
```

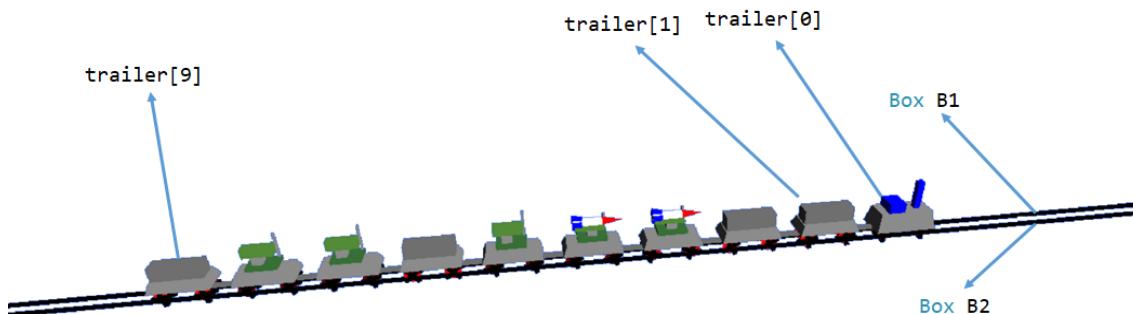


```
struct Trailer :
```

אובייקט קרוואן יש כמה צוריות ממה יכולם לשגר טילים על המסוק
(השחקן) ולהכשיל אותו.

```
int Type :               מאפיין סוג הקרוואן :  
Location location :     מאפיין מקום הקרוואן :  
Rotation rotation :    מאפיין רוטציה הקרוואן :  
Box Missile_Area :     מאפיין שטח התקרכבות המסוק לקרוואן :  
Wheel wheel[4] :         מאפיין ששה גלגלים הקרוואן :  
Missile_Launcher ML :  מאפיין משגרטיים
```

```
typedef struct Train
{
    Location location_1,location_2 ;
    Trailer trailer[10] ;
    Speed speed ;
    Box B1,B2 ;
}Train ;
```

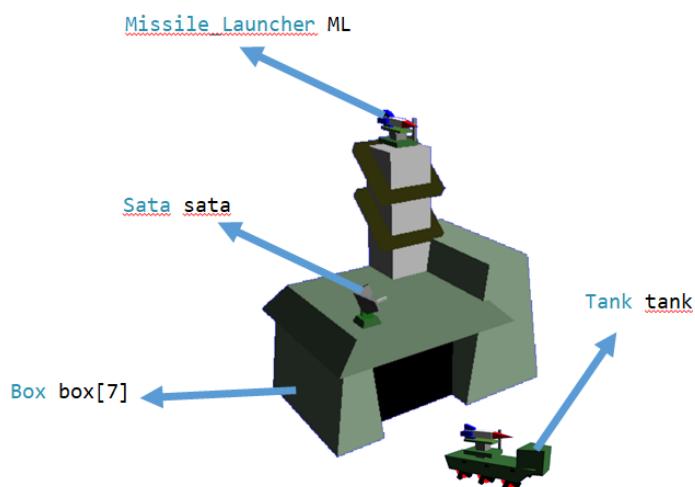


```
struct Train :
```

אובייקט רכבת מנוהל עשרה קרוואנים מנוקדה לנוקדה ב מהירות מסויימת.

מאפיינים שתי מקומות שהרכבת נעה ביניהם:
`Location location_1,location_2` מאפיין עשרה קרוואנים :
`Trailer trailer[10]` מאפיין מהירות הרכבת :
`Speed speed`

```
typedef struct Bullding_With_Tank
{
    Tank tank ;
    Sata sata ;
    Location location ;
    Missile_Launcher ML ;
    Box box[7] ;
    Box Virtual_Boxes ;
    Box Missile_Area ;
}Bullding_With_Tank ;
```



struct Bullding_With_Tank :

אובייקט בניין עם טנק, בניין זה מנהל טנק ו משגר טילים במקרה שהמסוק (השחקן) התקרב אליו.

מאפיין הטנק בתוך הבניין :

מאפיין מקום הבניין :

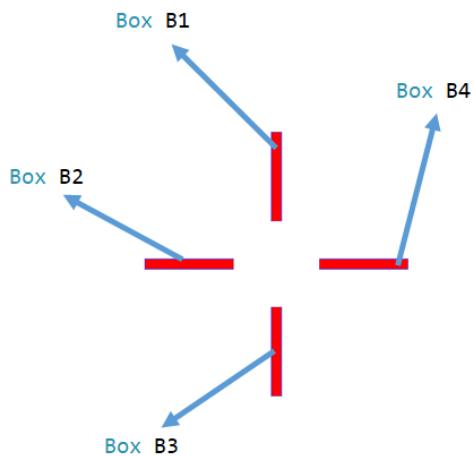
מאפיין משגר טילים בתוך הבניין :

מאפיין בוקס וירטואלי לבדיקת התנגשות :

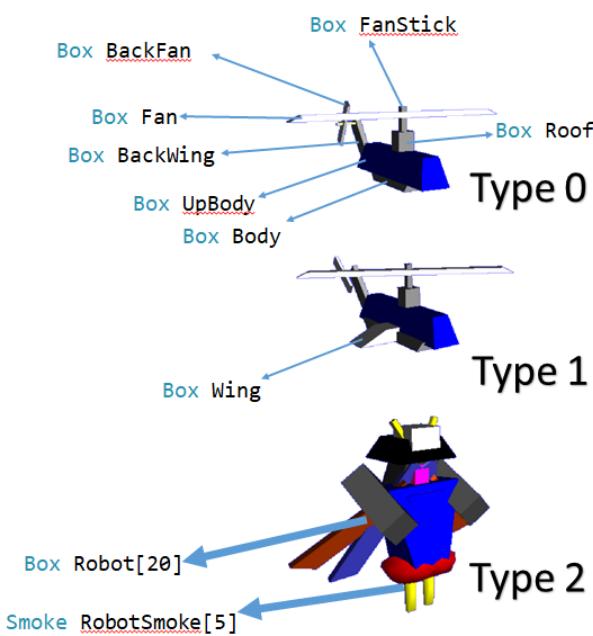
מאפיין שטח התקרובות המסוק לבניין :

```
typedef struct Aim
{
    Location location ;
    Rotation rotation ;
    Box B1,B2,B3,B4 ;
    float UPDown ;
}Aim ;
```

```
struct Aim :
    אובייקט זה נמצא בתוך המטוק לחישוב איזה דואית לזרק
    טילים, פצצים וכדומה.
{
    Location location : מאפיין מיקום המטרה
    Rotation rotation : מאפיין רוטציית המטרה
    float UPDown : מאפיין מיקום המטרה בהתאם למסקוק בציר X
```



```
typedef struct Helicopter
{
    int Health,Reload,Ready,Status,FanSpeed,X_Or_0 ;
    float fuel ;
    int Missile_Cnt,Bullet_Cnt,Bomb_Cnt,Hit,Coins ;
    int CameraX,CameraY,CameraZ,LookAround_X,LookAround_Y ;
    Speed speed ;
    User *user ;
    XO *MyXo ;
    Location location ;
    Rotation rotation ;
    Box Fan,BackFan,Roof ;
    Box FanStick,BackFanStick,BackWing ;
    Box Body,UpBody,Wing,BoundArea ;
    int bomb_index ;
    Bullet Bombs[4] ;
    Missile missile[4] ;
    int missile_index ;
    Aim aim ;
    int bullet_index ;
    Bullet bullet[5] ;
    Box Robot[20] ;
    Smoke RobotSmoke[5] ;
}Helicopter ;
```



```

struct Helicopter {
    מאפיין חיים המסוק : אובייקט המסוק זה אובייקט שהשחקן משתמש עליו.
    int Health ;
    מאפיין זמן בין כל יירה : מאפיין שען אחרי יירה.
    int Reload ;
    מאפיין אם השחקן מוכן לשחק : מאפיין מהירות המסוק.
    int Ready ;
    מאפיין מצב המסוק (מתגנש לא מתגנש) : מאפיין הקופסה שהמסוק טופס אותה.
    int Status ;
    מאפיין מהירות המסוק בציר Y : מאפיין מיקום המסוק.
    int FanSpeed ;
    מאפיין סוג הקופפה השicityת לשחקן : מאפיין רוטציית המסוק.
    int X_Or_O ;
    מאפיין הדלק למסוק : מאפיין ארבעה ביצים במסוק.
    float fuel ;
    מאפיין כמה טילים נשאר במסוק : מאפיין ארבע טילים לירוק.
    int Missile_Cnt ;
    מאפיין כמה כדורים נשאר במסוק : מאפיין ארבעה כורים לירוק.
    int Bullet_Cnt ;
    מאפיין כמה פצצים נשאר במסוק : מאפיין חמישה כורים לבולטים.
    int Bomb_Cnt ;
    מאפיין סטלל המסוק במצב שבגע בה טיל : מאפיין המטרת של המסוק.
    int Shake ;
    מאפיין כמה נקודות השחקן אסף : מאפיין איזה כדור מוכן לירוק.
    int Coins ;
    מאפיין מיקום המצלימה בaczירא : מאפיין איזה טיל מוכן לירוק.
    int CameraX ;
    מאפיין מיקום המצלימה בaczירא : מאפיין חמישה כורים לבולטים.
    int CameraY ;
    מאפיין מיקום המצלימה בaczirא : מאפיין חמישה כורים לבולטים.
    int CameraZ ;
    מאפיין זווית המצלימה בaczirא : מאפיין חמישה כורים לבולטים.
    int LookAround_X ;
    מאפיין זווית המצלימה בaczirא : מאפיין חמישה כורים לבולטים.
    int LookAround_Y ;
    מאפיין זווית המצלימה בaczirא : מאפיין חמישה כורים לבולטים.
}

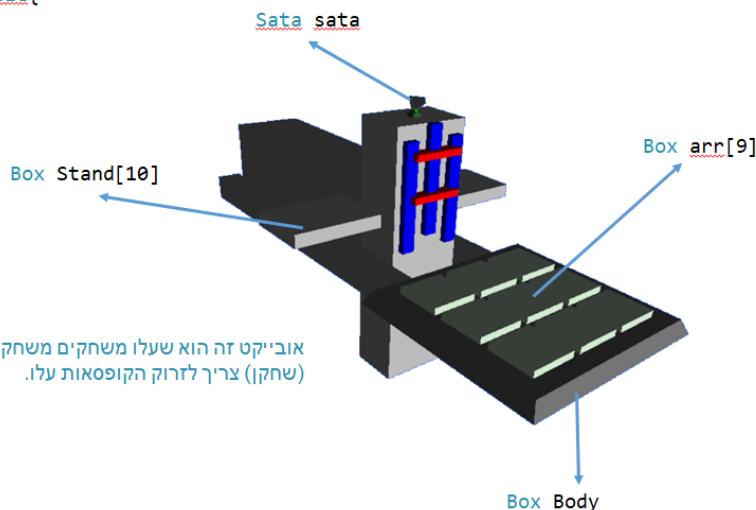
```

```

typedef struct XO_Table{
    Location location ;
    Box arr[9] ;
    Box Body ;
    Box Stand[10] ;
    Sata sata ;
}XO_Table ;

```

אובייקט זה הוא שולו של שחקן משחק האקס עיגול קר שמסוק
(שחקן) צריך לזרוק הקופסאות על.



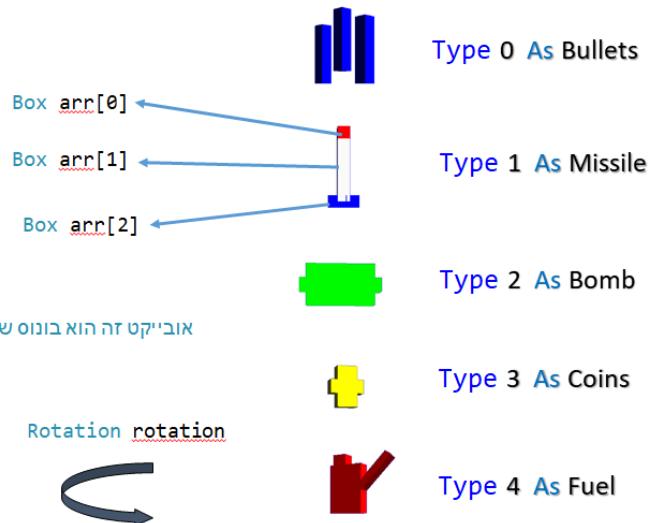
אורט ברואודה

המכללה הטכנולוגית להנדסאים

```
typedef struct Bonus{
    Location location ;
    Rotation rotation ;
    int Type ;
    int Status ;
    Box arr[3] ;
    Box Bound ;
}Bonus ;
```

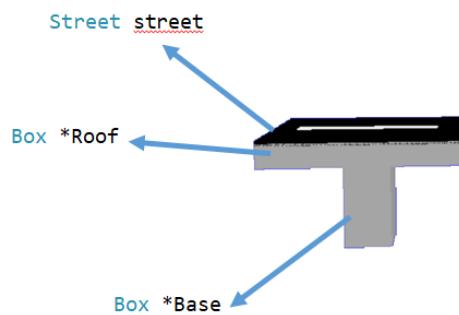
```
struct Bonus :
```

אובייקט זה הוא בונוס שהשחקן יכול לאסוף בזמן המשחק.

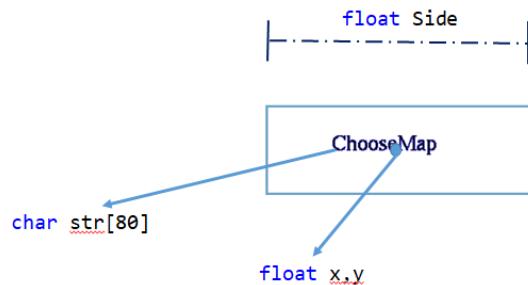


```
typedef struct Bridge
{
    Location location ;
    int Size ;
    Street street ;
    Box *Roof ;
    Box *Base ;
}Bridge ;
```

```
Box Base[3]
Box Roof[3]
```



```
typedef struct Button
{
    float x,y ;
    char *strPointer ;
    char str[80] ;
    float Side ;
    Color color ;
}Button ;
```



מאפיין מיקום הכפתור על המסך :
 מאפיין מצביע לצורכי הדפסה על המסך :
 מאפיין טקסט מסוים שמנוע בעבור הכפתור :
 מאפיין גודל הכפתור :
 מאפיין צבע הכפתור :

```
typedef struct User
{
    char User_Name[20] ;
    char Password[20] ;
    int Type ;
    int Coins ;
    int Wins ;
    int Loses ;
    float Bullet_Damage ;
    float Bomb_Damage ;
    float Missile_Damage ;
}User ;
```

struct User :

אובייקט זה מכיל נתונים של שחקן מסוים.

```
typedef struct GameList
{
float X,Y ;
char Player1[20] ;
char Player2[20] ;
char Winner[20] ;
char *str ;
struct GameList *next ;
}GameList ;
```

Player 1	Player 2	Winner_Is
Foad	Maram	Foad
Foad	Maram	Maram
Foad	Maram	Draw

struct GameList :
אובייקט זה תוצאות משחקים. בצורת רשימה מקוישת בכל חוליה יש
תוצאות משחק אחד.

מאפיין ההדפסה על המסך : **X,Y**
מאפיין השחקן הראשון : **Player1[20]**
מאפיין השחקן השני : **Player2[20]**
מאפיין השחקן המנצח : **Winner[20]**
מאפיין מצביע לצורכי הדפסה על המסך : **str**
מאפיין מצביע על המשחק הבא : **struct GameList *next**

ח堤ימת הפונקציה	הסבר
<code>float Distince_xz(Location *L1, Location *L2);</code>	מקבלת שני משתנים מסוג מיקום כפרמטרים ומחזירה ערך שהוא המרחק בין ציר X ו Z בין שני המיקומים (חישוב פיתגורוס).
<code>float Distince_x(Location *L1, Location *L2);</code>	מקבלת שני משתנים מסוג מיקום ומחזירה ערך שהוא הפרש בין נקודות ה X בפרקטר הראשון והשני (המרחק בין שתי נקודות בציר X).
<code>float Distince_z(Location *L1, Location *L2);</code>	מקבלת שני משתנים מסוג מיקום ומחזירה ערך שהוא הפרש בין נקודות ה Z בפרקטר הראשון והשני (המרחק בין שתי נקודות בציר Z).
<code>void Set_Speed(Speed *S, float x, float y, float z);</code>	מקבלת כפרקטר משתנה מסוג מהירות ומתחילה ערכיו לפי ה Z,Y,X שהועברו כפרקטרים.
<code>void Set_Size(Size *S, float x, float y, float z);</code>	מקבלת כפרקטר משתנה מסוג גודל ומתחילה ערכיו לפי ה Z,Y,X שהועברו כפרקטרים.
<code>void Set_Box_Size(Box *B, float size_x, float size_y, float size_z);</code>	מקבלת כפרקטר משתנה מסוג קוטסה ומתחילה את ערכי המאפיין גודל שבקובוסה לפי ה Z,Y,X שהועברו כפרקטרים.
<code>void Set_Color(Color *R, float r, float g, float b);</code>	מקבלת כפרקטר משתנה מסוג צבע ומתחילה ערכיו לפי ה b,g,r שהועברו כפרקטרים.
<code>void Set_Box_Color(Box *B, float r, float g, float b);</code>	מקבלת כפרקטר משתנה מסוג קוטסה ומתחילה את ערכי המאפיין צבע שבקובוסה לפי ה b,g,r שהועברו כפרקטרים.
<code>void Color_On(Color C);</code>	מקבלת כפרקטר משתנה מסוג צבע ומתחילה את הצבע שהועבר.
<code>void Set_Location(Location *L, float x, float y, float z);</code>	מקבלת כפרקטר משתנה מסוג מיקום ומתחילה אותו לפי ה Z,Y,X שהועברו כפרקטרים.
<code>void Set_Box_Location(Box *B, float x, float y, float z);</code>	מקבלת כפרקטר משתנה מסוג קוטסה ומתחילה את ערכי המאפיין מיקום שבקובוסה לפי ה Z,Y,X שהועברו כפרקטרים.
<code>void Go_Location(Location L);</code>	מקבלת כפרקטר משתנה מסוג מיקום וחולכת למיקום זהה.
<code>void Back_Location(Location L);</code>	מקבלת כפרקטר משתנה מסוג מיקום ו חוזרת למיקום זהה.
<code>void Copy_Location(Location *L, Location L1);</code>	מקבלת שני משתנים מסוג מיקום כפרקטרים ומתחילה ערכיו הפרקטר הראשון בערך הפרקטר השני.
<code>void Add_Location(Location *L, Location L1);</code>	מקבלת שני משתנים מסוג מיקום כפרקטרים ומוסיפה לערכו הפרקטר הראשון את ערכו הפרקטר השני.

<code>void Set_Rotation(Rotation *R,int x,int y,int z);</code>	מקבלת כפרמטר משתנה מסוג רוטציה ומאתחלת אותו לפיה Z,Y,X שהועברו כפרמטרים.
<code>void Set_Box_Rotation(Box *B,int Rotx,int Roty,int Rotz);</code>	מקבלת כפרמטר משתנה מסוג קובסה ומאתחלת את ערכי המאפיין הרוטציה שבקובסה לפיה Rotx,Roty,Rotz שהועברו כפרמטרים.
<code>void Rotation_Open(Rotation R,int f1,int f2,int f3);</code>	מקבלת משתנה מסוג רוטציה כפרמטר ועושה רוטציה בפלאס לצירים Z,y,x (לפי f1,f2,f3) בהתאם (כלומר אם הערך 1 אז מפעילה).
<code>void Rotation_Close(Rotation R,int f1,int f2,int f3);</code>	מקבלת משתנה מסוג רוטציה כפרמטר ועושה רוטציה במינוס לצירים Z,y,x (לפי f1,f2,f3) בהתאם (כלומר אם הערך 1 אז מפעילה).
<code>void RotateUntil_X(Rotation *R,Location FROM,Location TO,int add);</code>	מקבלת כתובות של ROTATION ומחשבת אלו את הרוטציה המתאימה בציר X בין שתי ניקודות.
<code>void RotateUntil_Y(Rotation *R,Location FROM,Location TO,int add);</code>	מקבלת כתובות של ROTATION ומחשבת אלו את הרוטציה המתאימה בציר Y בין שתי ניקודות.
<code>void RotateUntil_X_Half(Rotation *R,Location FROM,Location TO,int add);</code>	מקבלת כתובות של ROTATION ומחשבת אלו את הרוטציה המתאימה בציר X בין שתי ניקודות. (אבל מוגבל נתן רק בין 90 ל-90 מעלות)
<code>void RotateUntil_XY(Rotation *R,Location FROM,Location TO,int add);</code>	מקבלת כתובות של ROTATION ומחשבת אלו את הרוטציה המתאימה בציר X בין שתי ניקודות. (אבל מוגבל נתן רק בין 90 ל-90 מעלות) וגם מחשב לציר Y.
<code>void Set_Box(Box *B,float x,float y,float z,float size_x,float size_y,float size_z,float Rotx,float Roty,float Rotz,float r,float g,float b);</code>	מקבלת משתנה מסוג קובסה ומאתחלת את מאפיינו לפיה הפרמטרים שהועברו.
<code>void Draw_Box(Box B);</code>	מקבלת כפרמטר משתנה מסוג קובסה ומצירת אותו.
<code>void Draw_Box_R(Box B);</code>	מקבלת כפרמטר משתנה מסוג קובסה מצירת הקובסה עם רוטציה.
<code>void Draw_Box_T(Box B);</code>	מקבלת כפרמטר משתנה מסוג קובסה מצירת הקובסה עם במיקום מסוים.
<code>void Draw_Box_TR(Box B);</code>	מקבלת כפרמטר משתנה מסוג קובסה מצירת הקובסה עם רוטציה ובמיקום מסוים.
<code>void Draw_Shaped_Box_Up(Box B,float x,float y,float z);</code>	מקבלת כפרמטר משתנה מסוג קובסה מצירת הקובסה עם רוטציה ובמיקום מסוים. וגם מאפשרת לשנות גודל השטח העליון בקובסה.
<code>void Draw_Shaped_Box_Front(Box B,float</code>	מקבלת כפרמטר משתנה מסוג קובסה מצירת

<code>x, float y, float z);</code>	הוקופה עם רוטציה ובמיון מסוים. גם אפשרות לשנות גודל השטח הקדמי בהקופה.
<code>void Set_Smoke(Smoke *s, float MaxRadius, float speed, float R, float G, float B);</code>	מקבלת משתנה מסוג עשן כפרמטר ומאתחלת את ערכיו לפי הפרמטרים שהועברו.
<code>void Draw_Smoke(Smoke *s);</code>	מקבלת משתנה מסוג עשן כפרמטר ומצירת אותו.
<code>void Draw_Blow_Smoke(Smoke *s);</code>	מקבלת משתנה מסוג עשן כפרמטר ומצירת את העשן האחרון.
<code>void Set_Missile(Missile *missile, float x, float y, float z, int type)</code>	מקבלת כפרמטר משתנה מסוג טיל ומאתחלת את מאפיינו לפי הערכים שהועברו כפרמטרים.
<code>void Draw_Missile(Missile *missile);</code>	מקבלת כפרמטר משתנה מסוג טיל ומצירת אותו.
<code>void Reset_Missile(Missile *missile);</code>	מקבלת כפרמטר משתנה מסוג טיל ומחזירה אותו למיקום שיצא ממנו.
<code>void Missile_Movment(Missile *missile);</code>	תנועת הטיל.
<code>void Set_MissileLauncher(Missile_Launcher *ML, float x, float y, float z, int Type)</code>	מקבלת כפרמטר משתנה מסוג משגר טילים ומאתחלת את מאפיינו לפי הערכים שהועברו כפרמטרים.
<code>void Draw_Missile_Launcher(Missile_Launcher *ML);</code>	מקבלת כפרמטר משתנה מסוג משגר טילים ומצירת אותו.
<code>void MissileLauncher_Movment(Missile_Launcher *ML);</code>	תנועת משגר הטילים.
<code>void Set_Wheel(Wheel *w, float x, float y, float z, float R);</code>	מקבלת כפרמטר משתנה מסוג גלגל ומאתחלת את מאפיינו לפי הערכים שהועברו כפרמטרים.
<code>void Draw_Wheel(Wheel *w);</code>	מקבלת כפרמטר משתנה מסוג גלגל ומצירת אותו.
<code>void Set_Sata(Sata *s, float x, float y, float z, int Type);</code>	מקבלת כפרמטר משתנה מסוג לויין ומאתחלת אותו במיקום מסוים.
<code>void Draw_Sata(Sata *s);</code>	מקבלת כפרמטר משתנה מסוג לויין ומצירת אותו.
<code>void Tank_Launch_Missile(Tank *T, int Height);</code>	שיgor טיל מהטנק
<code>void Switch_Tank_Target(Tank *T);</code>	מעבר בין שני מיקומים לטנק
<code>void Set_Tank(Tank *T, float x, float y, float z, int Type);</code>	מקבלת כפרמטר משתנה מסוג טנק + מאתחלת מאפיינו לפי הפרמטרים שהועברו. טנק שיוצג מבניין.
<code>void Set_Tank_1(Tank *T, float x, float y, float z, float x1, float y1, float z1, int Type);</code>	מקבלת כפרמטר משתנה מסוג טנק + מאתחלת מאפיינו לפי הפרמטרים שהועברו. טנק שנע בין שני מקומות.
<code>void Draw_Tank(Tank *T)</code>	מקבלת כפרמטר משתנה מסוג טנק ומצירת אותו.
<code>void Tank_Movement_2(Tank *T)</code>	מקבלת כפרמטר משתנה מסוג טנק ומנוהלת אותו בצורה סטטית (לא יוזז).

<code>void Tank_Movement_1(Tank *T)</code>	מקבלת כפרטמר משתנה מסוג טנק ונהלת אותו בצורה סטאטית (עובר מנוקודה לנוקודה).
<code>void Tank_Movement(Tank *T)</code>	מקבלת כפרטמר משתנה מסוג טנק ונהלת אותו בצורה לעקב אחרי מטרתו.
<code>void Blow_Tank(Tank *T)</code>	מקבלת כפרטמר משתנה מסוג טנק ומפוץז אותה.
<code>void Set_Bullding_With_Tank(Bullding_With_Tank *b, float x, float y, float z, int Type)</code>	מקבלת כפרטמר משתנה מסוג בניין עם טנק ומתחילה אותה במקום מסוים. (הבניין והטנק)
<code>void Draw_Bullding_With_Tank(Bullding_With_Tank *b)</code>	מקבלת כפרטמר משתנה מסוג בניין עם טנק ומצירת אותו. (הבניין והטנק)
<code>void Bullding_With_Tank_Movement(Bullding_With_Tank *b)</code>	מקבלת כפרטמר משתנה מסוג בניין עם טנק ומנהלת אותו. (הבניין והטנק)
<code>void Set_Street(Street *S, float x, float y, float z, float sx, float sz, int RotateY)</code>	מקבלת כפרטמר משתנה רחוב ומתחילה אותו במקום מסוים.
<code>void Draw_Street(Street *S)</code>	מקבלת כפרטמר משתנה רחוב ומצירת אותו.
<code>void Set_Tree(Tree *t, float x, float y, float z)</code>	מקבלת כפרטמר משתנה עצם ומתחילה אותו במקום מסוים.
<code>void Draw_Tree(Tree *T)</code>	מקבלת כפרטמר משתנה עצם ומצירת אותו.
<code>void Set_Cloud(Cloud *C, float x, float y, float z)</code>	מקבלת כפרטמר משתנה ענן ומתחילה אותו במקום מסוים.
<code>void Draw_Cloud(Cloud *C)</code>	מקבלת כפרטמר משתנה ענן ומצירת אותו.
<code>void Shoot_Bomb(Helicopter *Heli)</code>	מקבלת כפרטמר משתנה מסוק ומשחררת ממנה פצצה.
<code>void Reset_Bomb(Bullet *b)</code>	מקבלת כפרטמר משתנה כדורי ומضافת אותו.
<code>void Bomb_Movement(Bullet *b)</code>	מקבלת כפרטמר משתנה כדורי ומנהלת אותו ביצירת פצצה נופל.
<code>void Calculate_Speed(Speed *speed, Location *To, Location *From, int DivideSpeed)</code>	מקבלת כפרטמר משתנה מסוג מהירות ומחשבת אליו מהירות בשלושה צירים שיקח בחשבון מתאימה בין שני מקומות.
<code>void Shoot_Bullet(Helicopter *Heli)</code>	מקבלת כפרטמר משתנה מסוק ומשחררת ממנה כדורי.
<code>void Shoot_Missile(Helicopter *Heli)</code>	מקבלת כפרטמר משתנה מסוק ומשחררת ממנה טיל.
<code>void Bullet_Movement(Bullet *b)</code>	מקבלת כפרטמר משתנה כדורי ומנהלת אותו ביצירת כדורי.
<code>void Helicopter_Missile_Movemet(Missile *m)</code>	מקבלת כפרטמר משתנה טיל ומנהלת אותו.
<code>void Draw_Bullet(Bullet *b)</code>	מקבלת כפרטמר משתנה כדורי ומצירת אותו.
<code>void Set_Aim(Aim *aim)</code>	מקבלת כפרטמר משתנה מטרה ומתחילה אותה.
<code>void Draw_Aim(Aim *aim)</code>	מקבלת כפרטמר משתנה מטרה ומצירת אותה.

<code>void Set_Helicopter_1(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומתחילה אותו בצורה 1.
<code>void Set_Helicopter_2(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומתחילה אותו בצורה 2.
<code>void Set_Helicopter(Helicopter *H, float x, float y, float z, int X_Or_0, char *User_Name)</code>	מקבלת כפרטמר משתנה מסוק ומתחילה אותו בצורה מתאימה לפि הדרגה של השחקן.
<code>void Draw_Helicopter(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומציירת אותו.
<code>void Draw_Robot(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומציירת אותו בצורת רובוט.
<code>void Draw_Player(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומציירת אותו לפि הדרגה של השחקן.
<code>void Draw_Waiting_Player(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומציירת אותו. (מציראת את השחקן הממתין)
<code>void Helicopter_Movement(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומנהלת אותו.
<code>void MoveLeft(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק וזמןון ללכת בצד X במינוס.
<code>void MoveRight(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק וזמןון ללכת בצד X בפלוס.
<code>void MoveForward(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק וזמןון ללכת בצד Z במינוס.
<code>void MoveBack(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק וזמןון לכת בצד Z בפלוס.
<code>void Throw_Xo(Helicopter *H)</code>	מקבלת כפרטמר משתנה מסוק ומשחררת ממנה קופסת OX אם יש בה.
<code>int BoundFunc(Box B1, Location L1, Box B2, Location L2)</code>	מקבלת שתי קופסאות והנקודות שלהם ובזקקת אם יש התנגשות ביניהם או לא.
<code>int Bound_Status(Box B1, Location L1, Box B2, Location L2)</code>	מקבלת שתי קופסאות והנקודות שלהם ובזקקת אם יש התנגשות ביניהם או לא. ואם כן יש תחזור מייצ'י צד ביחס לקופסה הראשונה.
<code>int Bound_Helicopter_Tank(Helicopter *H, Tank *T)</code>	מקבלת מסוק וטנק ובודקת התנגשות ביניהם.
<code>void Bound_Helicopter_Build(Helicopter *H, Building_With_Tank *B)</code>	מקבלת מסוק ובניין עט טנק ובודקת התנגשות ביניהם.
<code>void Bound_Helicopter_Bridge(Helicopter *H, Bridge *B)</code>	מקבלת מסוק וגשר ובודקת התנגשות ביניהם.
<code>void Bound_Helicopter_Building(Helicopter *H, Building *B)</code>	מקבלת מסוק ובניין ובודקת התנגשות ביניהם.
<code>void Bound_Helicopter_Bonus(Helicopter *H, Bonus *B)</code>	מקבלת מסוק וボונוס ובודקת התנגשות ביניהם.
<code>void Bound_Helicopter_Trailer(Helicopter *H, Trailer *T)</code>	מקבלת מסוק וקרוואן ובודקת התנגשות ביניהם.
<code>void Bound_Helicopter_XO_Table(Helicopter *H, XO_Table *B)</code>	מקבלת מסוק ושולחן OX ובודקת התנגשות ביניהם.
<code>void Bound_Tank_Tank(Tank *T1, Tank *T2)</code>	מקבלת טנק וטנק ובודקת התנגשות ביניהם.
<code>void Bound_Tank_Build_With_Tank(Tank</code>	מקבלת טנק ובניין עם טנק ובודקת התנגשות

<code>*T,Bullding_With_Tank *B)</code>	בינויים.
<code>void Bound_Tank_Bridge(Tank *T,Bridge *B)</code>	מקבלת טנק ו גשר ובודקת התנגשות בינם.
<code>void Bound_Tank_Building(Tank *H,Building *B)</code>	מקבלת טנק ו בניין ובודקת התנגשות בינם.
<code>void Bound_Tank_Train(Tank *tank,Train *T)</code>	מקבלת טנק ו רכבת ובודקת התנגשות בינם.
<code>void Bound_Helicopter_XOBox(Helicopter *H , XO *xo)</code>	מקבלת מסוק ו קופסת XO ובודקת התנגשות בינם.
<code>void Bound_XOBox_Bridge(XO *xo,Bridge *bri)</code>	מקבלת גשר ו קופסת XO ובודקת התנגשות בינם.
<code>void Bound_XOBox_Building(XO *xo,Building *B)</code>	מקבלת בניין ו קופסת XO ובודקת התנגשות בינם.
<code>void Bound_XOBox_Build_With_Tank(XO *xo,Bullding_With_Tank *B)</code>	מקבלת בניין עם טנק ו קופסת XO ובודקת התנגשות בינם.
<code>int Bound_XOBox_Table(XO *xo,XO_Table *table)</code>	מקבלת שולחן XO ו קופסת XO ובודקת התנגשות בינם.
<code>void Bound_Bonus_Bridge(Bonus *bonus,Bridge *bri)</code>	מקבלת בונוס ו גשר ובודקת התנגשות בינם.
<code>void Bound_Bonus_Building(Bonus *bonus,Building *B)</code>	מקבלת בונוס ו בניין ובודקת התנגשות בינם.
<code>void Bound_Bonus_Build_With_Tank(Bonus *bonus,Bullding_With_Tank *B)</code>	מקבלת בונוס ו בניין עם טנק ובודקת התנגשות בינם.
<code>int Bound_Missile_Tank(Missile *missile,Tank *T,int Size)</code>	מקבלת טיל ו טנק ובודקת התנגשות בינם.
<code>int Bound_Missile_Bullding_With_Tank(Missile *missile,Bullding_With_Tank *B,int Size)</code>	מקבלת טיל ו בניין עם טנק ובודקת התנגשות בינם.
<code>void Bound_Missile_Bridge(Missile *missile,Bridge *B,int Size)</code>	מקבלת טיל ו גשר ובודקת התנגשות בינם.
<code>int Bound_Missile_Building(Missile *missile,Building *B,int Size)</code>	מקבלת טיל ו בניין ובודקת התנגשות בינם.
<code>int Bound_Missile_Train(Missile *missile,Train *T,int size)</code>	מקבלת טיל ו רכבת ובודקת התנגשות בינם.
<code>int Bound_Bomb_Tank(Bullet *bomb,Tank *tank,int Size)</code>	מקבלת כדורים ו טנק ובודקת התנגשות בינם.
<code>int Bound_Bomb_Bullding_With_Tank(Bullet *bomb,Bullding_With_Tank *B,int size)</code>	מקבלת כדורים ו בניין עם טנק ובודקת התנגשות בינם.
<code>void Bound_Bomb_Bridge(Bullet *bomb,Bridge *B,int size)</code>	מקבלת כדורים ו גשר ובודקת התנגשות בינם.
<code>int Bound_Bomb_Building(Bullet *bomb,Building *B,int Size)</code>	מקבלת כדורים ו בניין ובודקת התנגשות בינם.
<code>int Bound_Bomb_Train(Bullet *bomb,Train *T,int size)</code>	מקבלת כדורים ו רכבת ובודקת התנגשות בינם.
<code>void Set_Bridge(Bridge *B,float x,float y,float z,int Count,float eachSize,float height,int Horizon)</code>	מקבלת כפרטמר משטנה גשר ומתחילה אותו לפי אורך , גובה , אופקי או عمودי לפי שאר המשתנים שהפונקציה מקבלת.
<code>void Draw_Bridge(Bridge *B)</code>	מקבלת כפרטמר משטנה גשר ומצירת אותו.
<code>void Set_XO_Box(XO *xo,float x,float y,float z, int X_OR_0)</code>	מקבלת כפרטמר משטנה קופסת XO ומתחילה אותו.
<code>void Draw_XO_Box(XO *xo)</code>	מקבלת כפרטמר משטנה קופסת XO ומצירת אותו.

<code>void XO_Box_Movement(XO *xo)</code>	מקבלת כפרטמר משתנה קופסת XO ומנהלת אותו.
<code>void Building_Type_0(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Building_Type_1(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Building_Type_2(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Building_Type_3(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Building_Type_4(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Building_Type_5(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Building_Type_6(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Set_Building(Building *B, float x, float y, float z, int Type)</code>	מקבלת כפרטמר בניין ומתחילה אותו לפי הסוג שלו.
<code>void Set_Wall(Building *B, float x, float y, float z, float Size_x, float Size_y, float Size_z)</code>	מקבלת כפרטמר בניין ומתחילה בצורה של קיר עם הממדים.
<code>void Draw_Building(Building *B)</code>	מקבלת כפרטמר בניין ומצiertה אותו לפי הסוג שלו.
<code>void Build_Movement(Building *B)</code>	מקבלת כפרטמר בניין ומתחילה אותו.
<code>void Set_Button(Button *btn, float x, float y, float Side, float r, float g, float b, char *str)</code>	מקבלת כפרטמר פתוח ומתחילה אותו.
<code>void Draw_Button(Button *btn)</code>	מקבלת כפרטמר פתוח ומצiertה אותו.
<code>int ClickButton(Button btn, int x, int y)</code>	מקבלת כפרטמר פתוח ובודקת אם נלחץ עליו.
<code>void Type_Name(char *str, char ch)</code>	מקבלת כפרטמר מחרוזת אותן ומוסיפה אותן לסוף המחרוזת.
<code>void Remove_Char(char *str)</code>	מקבלת כפרטמר מחרוזת ומוחקkt האות الأخيرة.
<code>GameList *Create_Game(float x, float y, char *str1, char *str2, char *Winner)</code>	מקבלת כפרטמר מקום בمسך ושלושה מחרוזות ומקרה חוליה עם הנתונים האילו.
<code>GameList *Add_GameList(GameList *head, GameList *NewGameList)</code>	המקבלת כפרטמר ראש רשומה וחוליה ומוסיפה חוליה לסוף הרשימה.
<code>void Draw_Gamelist(GameList *Game_List)</code>	המקבלת כפרטמר רשומה ומצiertה אותה.
<code>void Draw_String_int(float r, float g, float b, float x, float y, float z, char *s, int val)</code>	המקבלת כפרטמר מחרוזת ומדפיסה על המסך אותה עם משתנה int .
<code>void Draw_String_float(float r, float g, float b, float x, float y, float z, char *s, float val)</code>	המקבלת כפרטמר מחרוזת ומדפיסה על המסך אותה עם משתנה float .
<code>void Draw_String(float r, float g, float b, float x, float y, float z, char *s)</code>	המקבלת כפרטמר מחרוזת ומדפיסה על המסך.
<code>void Draw_Square(float x, float y, float Side, float r, float g, float b)</code>	המקבלת כפרטמר מדיי כיכר ומצiertה אותו על המסך.
<code>void Set_Bouns(Bonus *B, float x, float y, float z, int Type)</code>	מקבלת כפרטמר בונוס ומתחילה אותו.
<code>void Draw_Bonus(Bonus *B)</code>	מקבלת כפרטמר בונוס ומצiertה אותו.
<code>void Bonus_Movement(Bonus *B)</code>	מקבלת כפרטמר בונוס ומנהלת אותו.
<code>void Trailer_Launch_Missile(Trailer *T, int Height)</code>	שיגור טיל מהקרואן
<code>void Set_Trailer(Trailer *T, float x, float y, float z, int Type)</code>	מקבלת כפרטמר קרוואן ומתחילה אותו.

<code>void Draw_Trailer(Trailer *T)</code>	מקבלת כפרטמר קרוואן ומצירת אותו.
<code>void Trailer_Movement(Trailer *T)</code>	מקבלת כפרטמר קרוואן ומנהלת אותו.
<code>void Set_Train(Train *T, float x, float y, float z, float x1, float y1, float z1)</code>	מקבלת כפרטמר רכבת ומתחילה אותו.
<code>void Draw_Train(Train *T)</code>	מקבלת כפרטמר רכבת ומצירת אותו.
<code>void Train_Movement(Train *T)</code>	מקבלת כפרטמר רכבת ומנהלת אותו.
<code>int CheckXO(Box *B)</code>	מקבלת כפרטמר קופסה ומחזירה אם היא X או O
<code>int Check_Winner(XO_Table *table)</code>	מקבלת כפרטמר שולחןOX ובודקת אם יש מישחו ניצח.
<code>void Set_XO_Table(XO_Table *table, float x, float y, float z)</code>	מקבלת כפרטמר שולחןOX ומתחילה אותה.
<code>void Draw_XO_Table(XO_Table *table)</code>	מקבלת כפרטמר שולחןOX ומצירת אותה.
<code>int User_Exists(char *Name)</code>	מקבלת כפרטמר שם משתמש ובודקת אם רשום.
<code>User* Get_User(char *Name)</code>	מקבלת כפרטמר שם משתמש ומחזירה הנותונים שלו.
<code>void In_User(User *user)</code>	מקבלת כפרטמר משתמש ושומרת הנתונים שלו.
<code>void Print_User(char *Name)</code>	מקבלת כפרטמר שם המשתמש ומצירת הנתונים שלו על המסך.
<code>int New_User(char *User_Name, char *Password)</code>	מקבלת כפרטמר שם משתמש וסיסמה ובודקת אם לא רשום אך תרשים אותו.
<code>int Login(char *User_Name, char *Password)</code>	מקבלת כפרטמר שם משתמש וסיסמה ובודקת אם רשום תאפשר לו כניסה.
<code>void Set_Stars(Star *s, float x, float y, float z)</code>	מקבלת כפרטמר כוכבית ומתחילה אותה.
<code>void Draw_Star(Star *s)</code>	מקבלת כפרטמר כוכבית ומצירת אותה.
<code>Box *Check_Row(Box *B1, Box *B2, Box *B3)</code>	מקבלת כפרטמר שלושה קופסאות (שורה או עמודה בשולחןOX) ובודקת אם אפשר לנצח בשורה זו.
<code>int Get_Move(Box *B1, Box *B2, Box *B3)</code>	מקבלת כפרטמר שלושה קופסאות (שורה או עמודה בשולחןOX) ומ核查ה אם אם המחשב שחקן נצח.

אלגוריתמייקה:

1. תנועת הטיל:

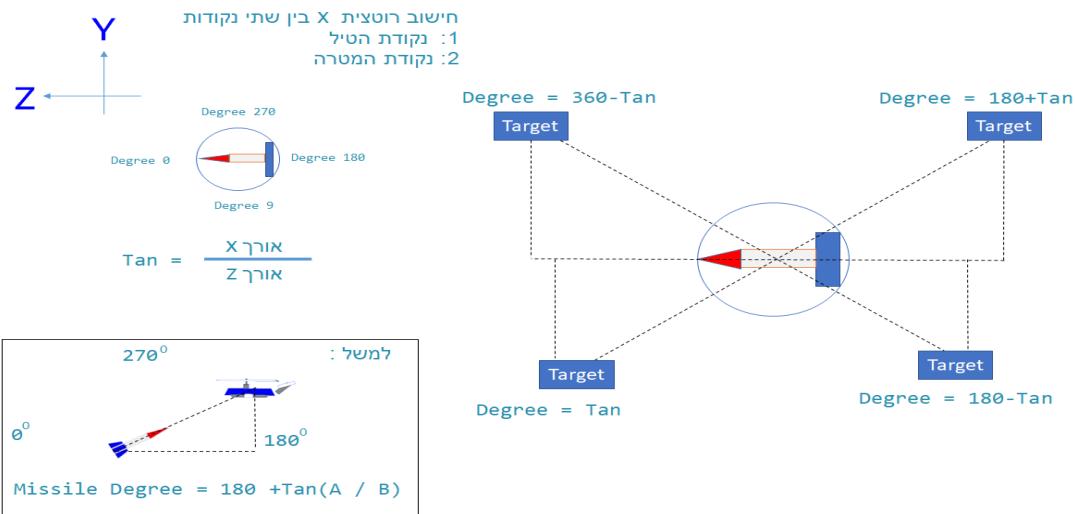
אם הטיל פועל כלומר מכוון למסוק מחשבים רוטציה שלו ובאיזה ציון נע וללאן לכת.

2. חישוב רוטציה:

מחשבים את זוויות הרוטציה דרך השימוש בחישוב atan בין שתי נקודות. הנקודהה הראשונה מסמנת את האובייקט שצרי להציב על אובייקט אחר. והנקודהה השנייה היא מיקום האובייקט השני.

- רוטציה על ציר X:

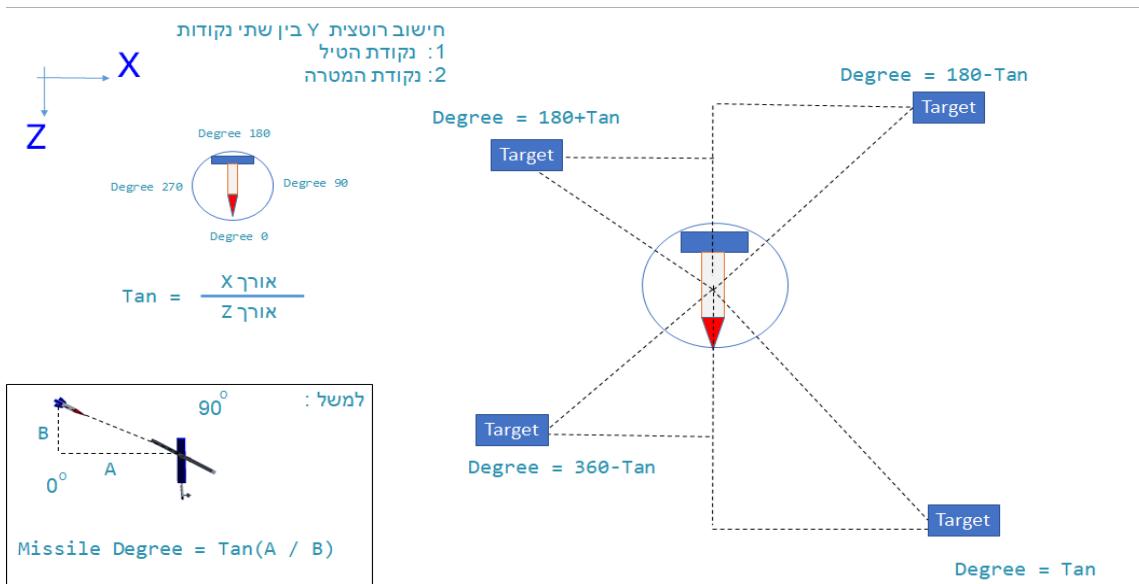
מחשבים atan לפרש הנקודות בציר Y חלקו הפרש הנקודות בציר Z.



- רוטציה על ציר Y:

מחשבים atan לפרש הנקודות בציר X חלקו הפרש הנקודות בציר Z.

לפי הציור הבא:

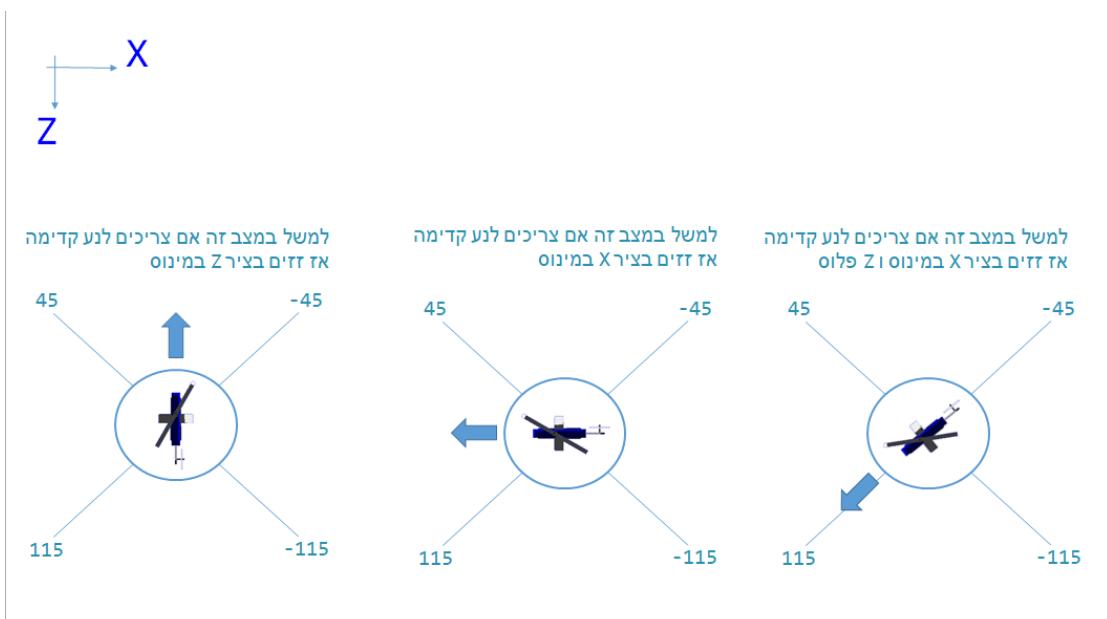


3. התנגשות המסוק:

התנגשות המסוק כמו ההתנגשות הרגילה לשאר האובייקטים אבל למסוק מחשבים מאייה צד יש התנגשות. חשבוה זווית ההתנגשות בצד לקבוע את מסלול תיקון כיוון הטיסה הנכון.

4. תנועת המסוק:

כאשר מזיזים את המסוק בעזרת העכבר חיברים לדעת באיזה זווית כיוון המסוק כדי שתיהיה ההזזה נconaה לפי הציור הבא:



5. הפיצוץ:

אם יש צורך להשמיד משהו עקב סיום החיים שלו או מעתיקים לאובייקט המשמיד את מיקום האובייקט צריך להשמיד אותו.

6. תנועת הטנק:

כאשר הטנק עוקב אחרי המסוק מחשבים הרוטציה המתאימה והמהירות. בתנועת הטנק אם התנגש עם בניין מסוים מחשבים לאיזה כיוון הטנק ימשיך לפי זווית ההתנגשות.

אם הייתה התנגשות בין שני טנקים מחשבים מי הטנק הקרוב ביותר מאפשרים לו להמשיך והשני עוצר

7. שחזור מול מחשב:

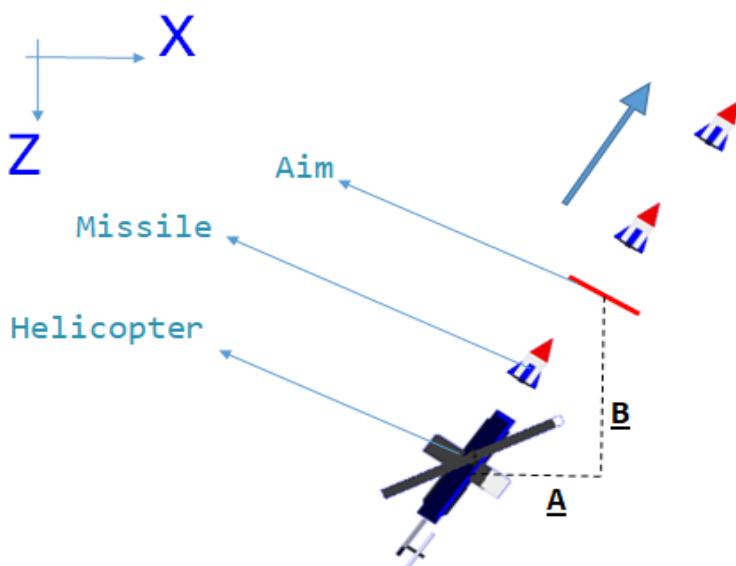
בפעם הראשונה שמשחק בוט **Bot** ישם בעדיפות ראשונה את העיגול במשבצת האמצעית אם היה פניו אמ' לו בוחר משבצת אקרואית.

מחשבים אם ה **Bot** יכול לנ匝ח בשורה. עמודה. אלכסון (כלומר אם יש שני עיגולים בשתי משבצות צמודות) אז ישם העיגול במשבצת השלישית שמהווה ניצחון לו.

במידה ולשחקן יש כבר שני איקסים במשבצות צמודות ה **Bot** צריך לחשב איך ליחסם את האפשרות להרכיב שורה. עמודה. אלכסון.

8. חישוב זווית לטיל או לכדור שיוצא ממסוק:

כאשר השחקן משליך כדור/טיל מהמסוק צריכים לחשב זווית הcador דרך חישוב היחס בין המטרה והמסוק בשלושה הצירים **Z,Y,X** מבחינות מרוחק ותנוועה. לפי הציור הבא:



$$\text{Missile.speed.x} = \text{(מהירות יחסית למטרה/A)}$$

$$\text{Missile.speed.z} = \text{(מהירות יחסית למטרה/B)}$$

*כל שהייתה המהירות היחסית
למטרה גדולה יותר, תנועת הcador
והטיל איטית יותר.

