

פרויקט גמר במדעי המחשב – תכנון

ותכנות מערכות מונחות עצמים – 5

יחידות לימוד בפלטפורמת UWP



שם העבודה: space invaders

תכנות שפת: c#, xaml, sql

שם המגיש: מתן אלדר

תעודת זהות: 215332479

בית ספר: אורט נעמי גן יבנה

שם המורה: אולג כצנלסון

תוכן עניינים

2.....	מבוא
2.....	רקע
4.....	לוגיקת המשחק
4.....	רקע המשחק/מטרת המשחק/תהליך המשחק
5.....	הפעלת המשחק
6.....	מדריך למשתמש
6.....	דף הכניסה למשתמש
9.....	דף הרשמה למשתמש חדש
11.....	דף חנות המשחק
13.....	דף מחסן הסקינים
14.....	דף מידע למשתמשים
15.....	עמוד טבלת שיאים
16.....	דף המשחק
17.....	מדריך למתכנת
17.....	דיאגרמת הפרוייקט
18.....	Manager
20.....	SpaceShip.cs
21.....	Enemy.cs
22.....	GreenEnemy.cs
23.....	RedEnemy.cs
24.....	Enemies.cs
25.....	Point
26.....	MovingItem.cs
27.....	Bullet.cs

28.....	Session.cs
29.....	דפי המשחק
29.....	MainPage.cs
30.....	GamePage.cs
31.....	HighScore.cs
32.....	InformationPage.cs
33.....	LockerPage.cs
34.....	Login.cs
35.....	RegisterPage.cs
36.....	ResetPasswordPage.cs
37.....	Shop.cs
38.....	DataBase
38.....	דיאגרמת הפרויקט
39.....	SqlHelper.cs
40.....	EmailManager.cs
41.....	User.cs
42.....	Product.cs
43.....	איכות הפרויקט
43.....	בעיות אלגוריתמיות/הצעות לשיפור
44.....	רפלקציה
45.....	מקורות

מבוא

רקע: הפרויקט שלי נוצר בכך שנוצרה בי נוסטלגיה מהזמנים שיחקתי את המשחק space invaders כשהייתי קטן. המשחק הוא חיקוי של space invaders עם שינויים משלי. במהלך כל תקופת הפרויקט ניסיתי לצאת מהקופסא וליצור משחק מהנה ומזמין העולה על כל הציפיות שהצבתי לעצמי מהיום הראשון שתכננתי על המשחק.

את המשחק בניתי באמצעות היכולות שלי והידע שרכשתי במהלך לימודי השפה c# עם המורה שלי אולג.

המשחק נבנה בפלטפורמת UWP(Universal Window Platform) ובממצעות שלוש שפות:

- שפת Xmal : האחראית על עיצוב ממשק הפרויקט.
- שפת c# : הבונה את החלק הלוגי של המשחק.
- שפת SQL : המיועדת למסד הנתונים בפרויקט.

לוגיקת המשחק:

רקע המשחק: המשחק מתרחש בעולם עתידי בו הגיבור של כדור הארץ יוצא אל עבר החלל על מנת לחקור את ניסרתיו. במהלך מחקרו הגיבור נתקל בחיזורים משונים אשר רוצים להשמיד אותו, הגיבור מנסה להשמיד את החיזורים אך הם ממשיכים לבוא בגלים.

מטרת המשחק: מטרת המשחק היא לשרוד כמה שיותר גלים של אויבים

תהליך המשחק: המשחק הוא משחק הרפתקאות המעגלי שבכל פעם השחקן צריך להשמיד את כל האויבים, כאשר כל האויבים על המסך הושמדו יופיעו אויבים חזקים ומהירים יותר

בכל מהלך המשחק השחקן שולט על דמותו(שבחר דרך דף הלווי) באמצעות מקשי המקלדת)



-----בשביל הגנה ל 10 שניות

במשחק עצמו ישנם פיצ'רים שהשחקן יכול להרוויח במהלך המשחק/בחנות:

ניקוד - סכום הכסף והניקוד של המשתמש איתם יוכל לרכוש בחנות דמויות ורקעים.

שיא ניקוד - מספר המייצג את הניקוד הגבוהה ביותר שהמשתמש השיג במשחק אחד

מגן - מוצר אשר מעניק לשחקן 10 שניות של חסינות לפגיעות

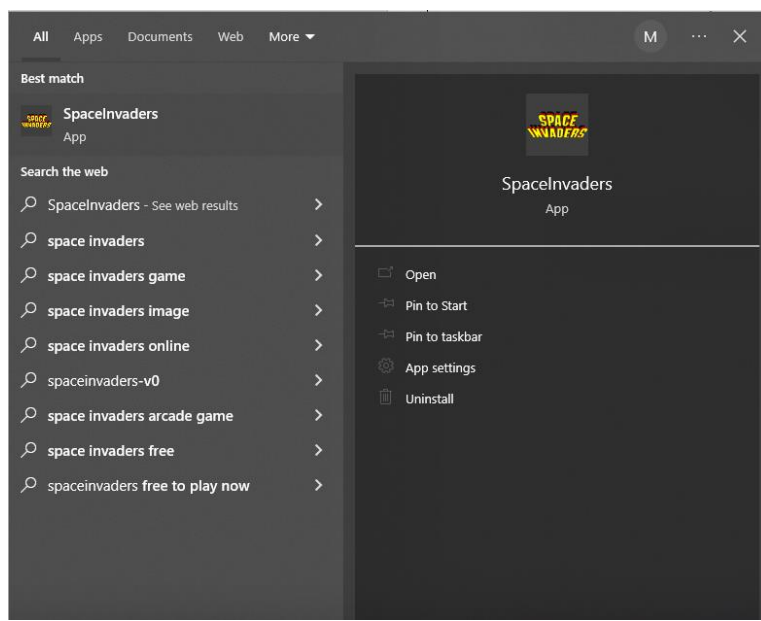
*שימו לב! משתמש חדש הנרשם לאחר, מתחיל כאשר כל פרטיו במשחק שווים לאפס מלבד חייו(השווים ל 3- בכל משחק). בנוסף לכך, המשתמש מקבל כברירת מחדל דמות ראשונה ואפס מגנים.

המשחק מתחיל בשלב הראשון בוא השחקן פוגש את החיזורים מהרמה הראשונה. כאשר השחקן הורג את כל החיזורים של על המסך קבוצה נוספת של חיזורים(שחלקם משופרים יותר) הסיטואציה חוזרת על עצמה עד שהשחקן מת. כאשר השחקן מת הנקודות מתווספות למאגר הנתונים.

בדף הלובי ישנם 3 דמויות למשחק לבחירת השחקן(במידה ורכש את השאר בחנות).

הפעלת המשחק:

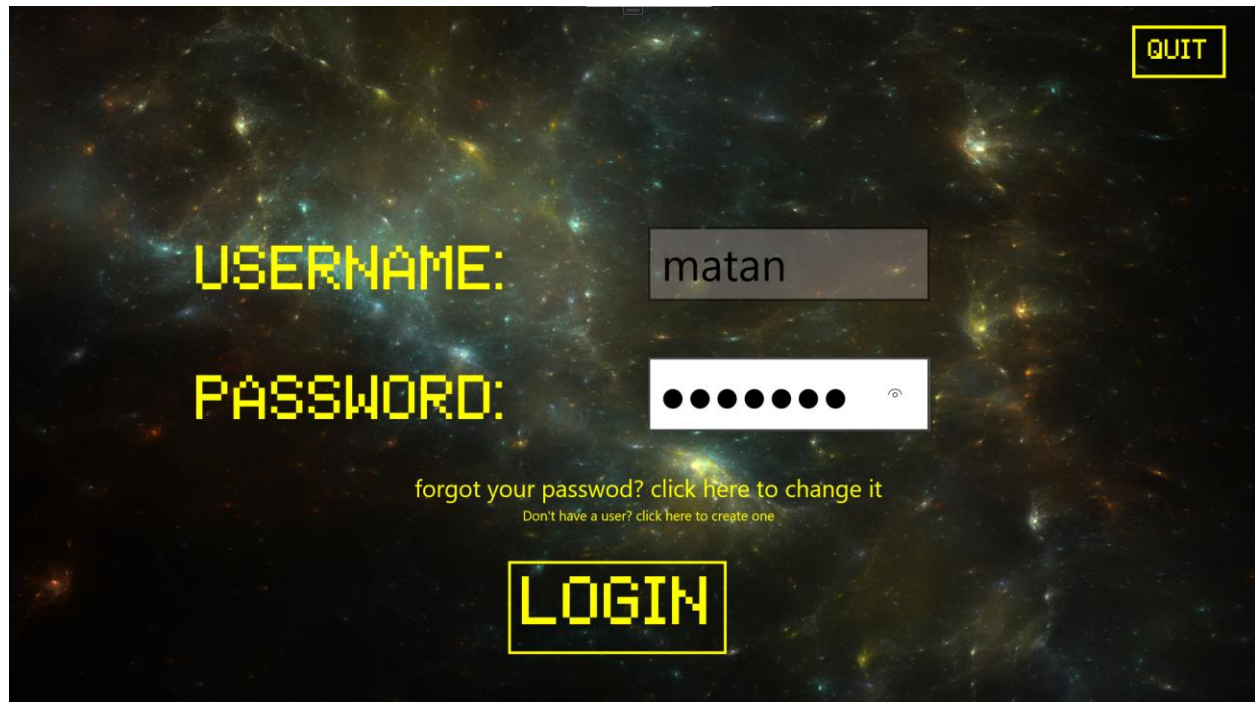
כדי להפעיל את המשחק במחשב, המשתמש צריך להקליד את שם המשחק: "SpaceInvaders" בסרגל החיפוש ולבחור באפליקציית התוכנה, לחיצה עליו תפתח ותריץ את המשחק בדף ניפרד.



בעמוד הבא ישנן הוראות אשר עוזרות למשתמש להתמצא במשחק עצמו.

מדריך למשתמש

דף הכניסה למשתמש:



כפתור סגירת האפליקציה

כפתור כניסה למשתמש

כפתור שכחתי סיסמא

כפתור מעבר להרשמה

חלון שכתי סיסמא ראשוני כאשר לוחצים על שכחתי
סיסמא





כפתור חזרה אחורה אשר יסגור את שכחתי סיסמא



--כפתור המשכה הבודק את תקינות האימייל ואם תקין ממשיך לחלון הבא .



--חלון שבו מכניסים את הקוד שנשלח למייל כדי לאשר את החשבון



--עמוד יצירת סיסמא חדשה



---כפתור האתחול סיסמא



שגיאה מופיע כאשר השם משתמש או הסיסמא לא נכונים

דף הרשמה למשתמש חדש:



A registration form with a space-themed background. It includes fields for USERNAME (matan793), EMAIL (matanepic79), PASSWORD (masked with dots), and CONFIRM PASSWORD (masked with dots). A 'REGISTER' button is at the bottom. A link 'already have a user? click here to login' is below the password fields. A 'QUIT' button is in the top right corner.

-כפתור יציאה מהאפליקציה

QUIT

-----כפתור הרשמה

REGISTER

-כפתור חזרה לכניסת משתמשים already have a user? click here to login

-שגיאה של סיסמאות לא תואמות

the passwords don't match. try again

- שגיאה של מייל לא תקין

Mail is not correct please try again

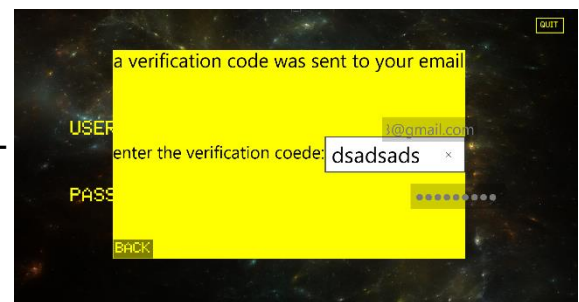
- שגיאה שהמייל קיים במאגר הנתונים

Mail already exist.

- שגיאה שהשם משתמש קיים במאגר הנתונים

Username already exist.

-חלון אישור קוד הנשלח למייל שהוכנס



A verification code window with a yellow background. It displays the message 'a verification code was sent to your email'. Below, it shows the email address 'i@gmail.com' and a text input field for the verification code containing 'dsadsads'. There are 'BACK' and 'QUIT' buttons.



-כפתור המתחיל משחק חדש

NEW GAME

-כפתור היוצא מהמשחק ומעביר אותו אל דף הכניסה

LOGOUT

-כפתור לסגירת האפליקציה

QUIT GAME

-כפתור השולח לדף מחסן הסקינים

LOCKER

-כפתור השולח את המשתמש לדף החנות

SHOP

- כפתור השולח את המשתמש לדף השיאים

HI-SCORES

INFORMATION - כפתור השולח את המשתמש אל דף המידע

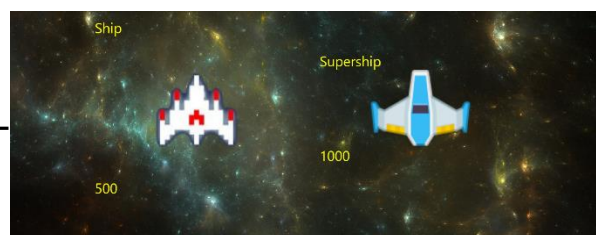
דף חנות המשחק:



-כפתור הזרחה למסך הבית



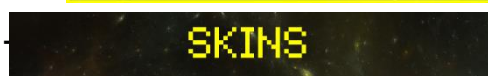
-רשימת סקינים עם מחירים.



-כפתור קנייה של סקין



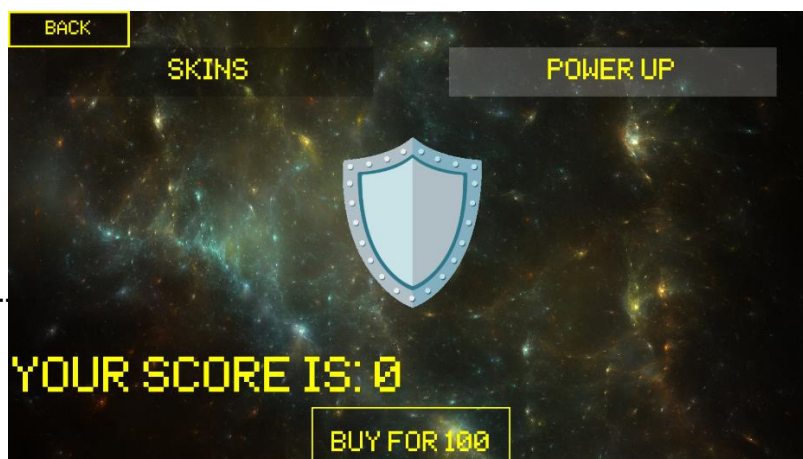
- כפתור מעבר לקניית סקינים



-כפתור מעבר לקניית מגנים



-חלון קניית מגנים



- כפתור לקניית מגן ב 100 נקודות



- שגיאה של מחסור בנקודות



- שגיאה של אי בחירת מוצר



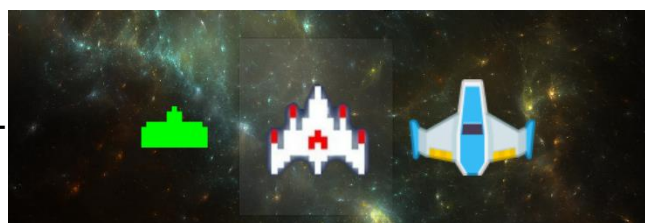
- טקסט המציג את הניקוד של השחקן



דף מחסן הסקינים:



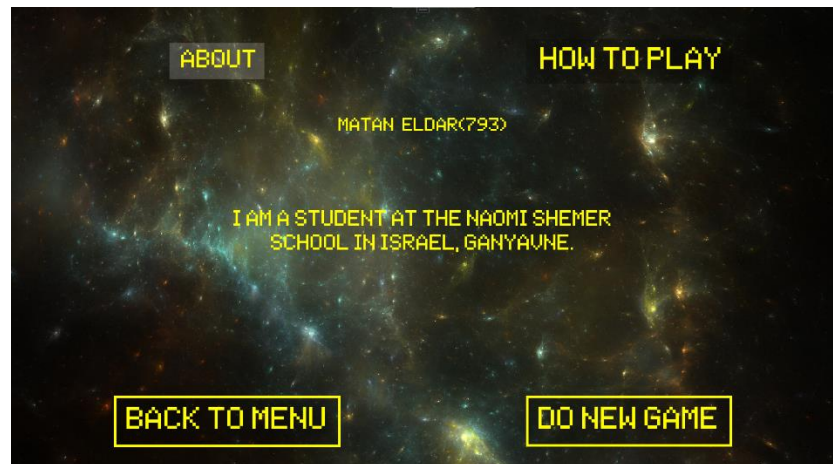
- רשימת סקינים לבחירה



- כפתור חזרה לדף הבית השומר את הבחירה של המשתמש
(אם המשתמש לא בחר סקין הסקין הנוכחי שלו ישאאר)



דף מידע למשתמשים:



-כפתור חזרה לדף הבית

BACK TO MENU

- מעבר למחשק חדש

DO NEW GAME

-מעבר למידע עליי

ABOUT

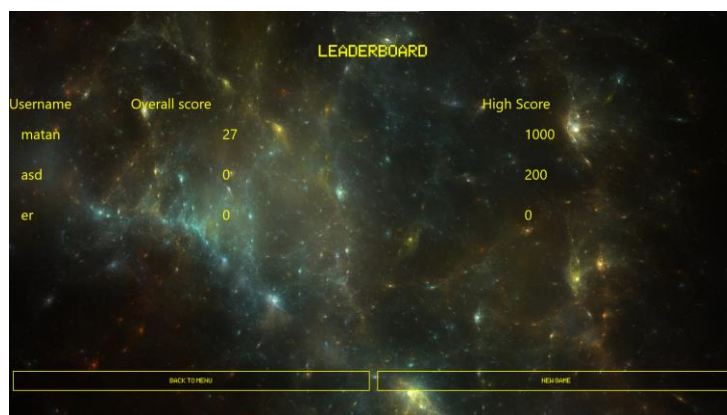
-מעבר לאיך לשחק

HOW TO PLAY

חלון ה'איך לשחק':



עמוד טבלת שיאים:



Username	Overall score	High Score
matah	27	1000
asd	0	200
er	0	0

BACK TO MENU NEW GAME

– טבלת שיאים הממוינת לפי השיא הגבוה ביותר (high score).



Username	Overall score	High Score
matah	27	1000
asd	0	200
er	0	0

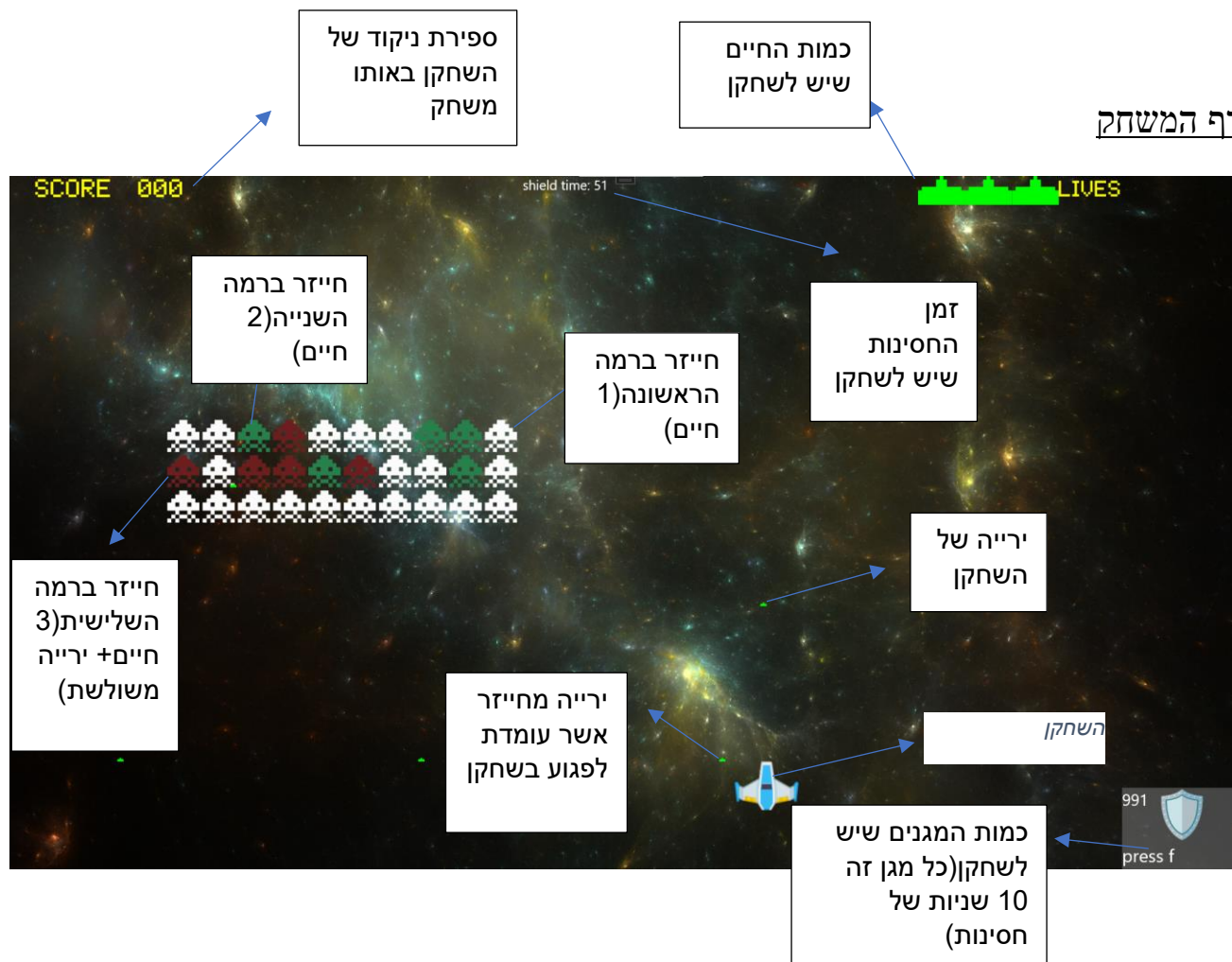
– כפתור חזרה לדף הבית



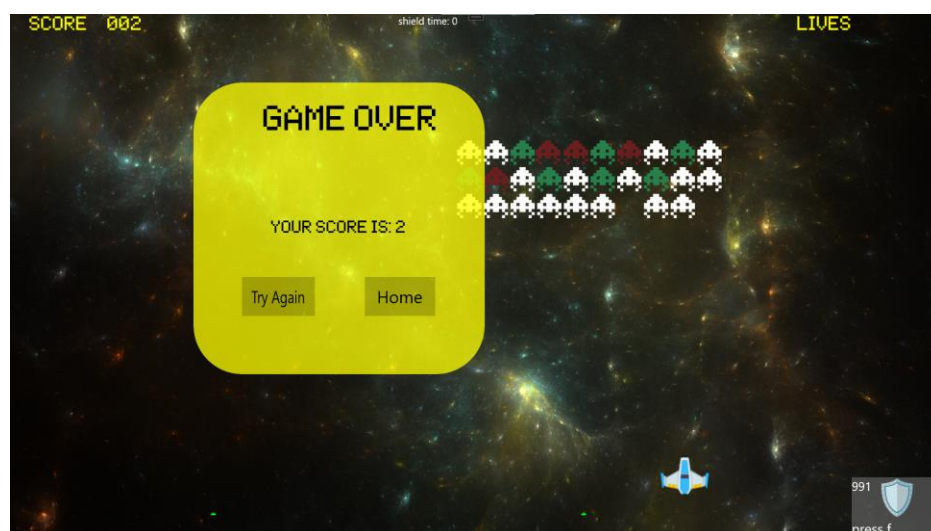
– כפתור להתחלת משחק חדש



דף המשחק



דף חנות

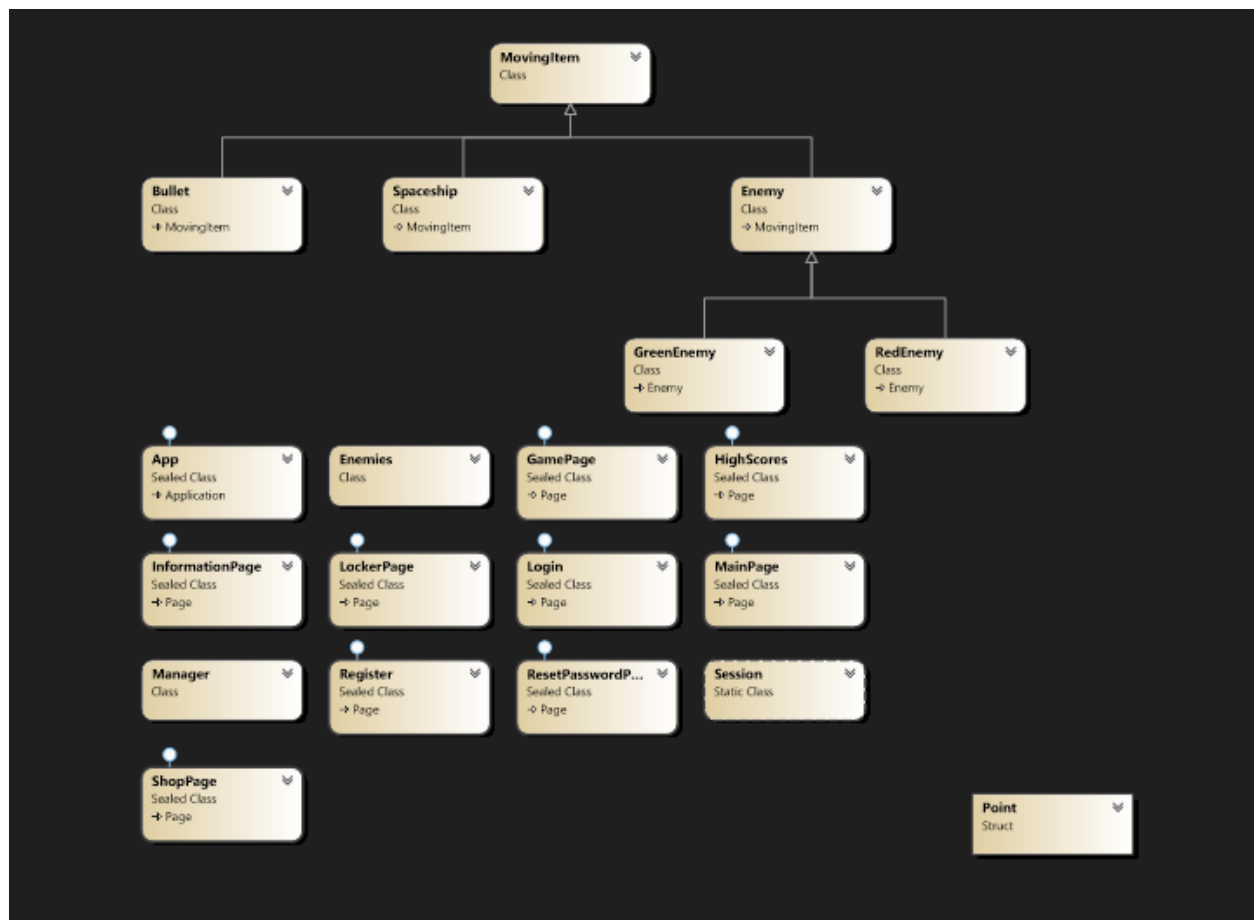




-גריד שמופיע כאשר השחקן מת, הגריד בעל לחצן נסה שוב .
 תאחל, כפתור חזרה המ חזיר את השחקן לדף הבית וטקסט
 המציג את ניקוד השחקן במשחק הנוכחי.

מדריך למתכנת:

דיאגרמת מבנה הפרוייקט:



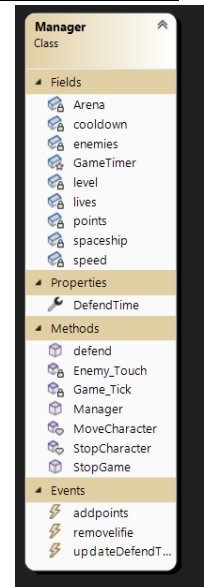
Maneger.cs

תפקיד המחלקה לנהל את פעילות המשחק: צבירת נקודות וכו(מהירות הדמות, עוצמת כוחו והזמן שיפעל), יצירת שלבים(עם רמות קושי שונות) יחד עם מכשולים ואויבים הנמחקים בהתאם, זיהוי ההתנגשות בין הדמות לאויבים ולמכשולים, זיהוי ההתנגשות בין קליע הדמות לבין האויב, זיהוי פעילות השחקן, תקתוק טיימרים נוספים וסיום המשחק.

תכונות המחלקה:

```
private Spaceship spaceship;
public event EventHandler removelifie; //מחקית לב מן השחקן
public event EventHandler addpoints; //הוספת נק
public event EventHandler updateDefendTime; //הוספת נק
private Enemies enemies;
Canvas Arena;
private int points;
private int lives;
private int level;
private uint cooldown;
protected DispatcherTimer GameTimer; //הטיימר שאחראי על תנועת הדמות
private int speed;

2 references
public int DefendTime { get; private set; }
```



פעולות המחלקה:

```
2 references
public int DefendTime { get; private set; }
/// <summary> פעולה בונה
1 reference
public Manager(Canvas Arena)...
/// <summary> פעולה אשר נקראת כל אלפית שנייה ואחראית על מעבר שלב, זמן יריית הכדו
1 reference
private void Game_Tick(object sender, object e)...
/// <summary> פעולה אשר נקראת על אלפית שנייה ובודרת אם האויב פגע בשחקן, ירייה ש
4 references
private void Enemy_Touch(object sender, EventArgs e)...
/// <summary> פעולה אשר נקראת כל פעם שהמשתמש לחץ על המקלדת שלו
1 reference
internal void MoveCharacter(VirtualKey virtualKey)...
/// <summary> פעולה אשר מוסיפה 10 לזמן החסינות
1 reference
public bool defend(...
/// <summary> פעולה אשר עוצרת את המשחק
1 reference
public void StopGame(...
/// <summary> פעולה אשר עוצרת את השחקן
2 references
internal void StopCharacter(...
```

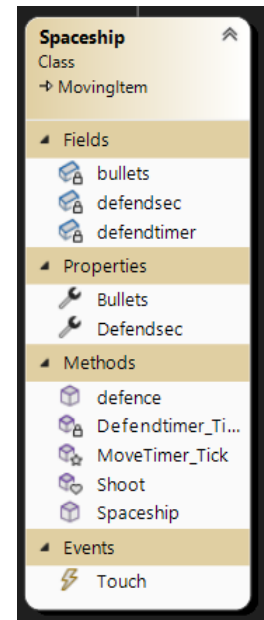
המחלקת Spaceship.cs

המחלקה מייצגת את השחקן ונראותו היא לפי הסקין הנוכחי.



```
public event EventHandler Touch;
private List<Bullet> bullets;
1 reference
public List<Bullet> Bullets { get { return bullets; } }
private int defendsec = 0;
2 references
public int Defendsec { get { return defendsec; } }
DispatcherTimer defendtimer;
```

תכונות המחלקה:



פעולות המחלקה:

```
/// <summary> פעולה כונה
1 reference
public Spaceship(double placeX, double placeY, Canvas arena, int size, Skin skin) ...
/// <summary> פעולה אשר נקראת כל אלפית שנייה ואחראית על לבדוק את גבולות השחקן זה
3 references
protected override void MoveTimer_Tick(object sender, object e)...
/// <summary> פעולה אשר גורמת לשחקן להיות חסין ב10 שניות יותר
1 reference
public void defence()...
/// <summary> פעולה אשר נקראת כל שנייה ומורידה שנייה מחזקן של החסינות
1 reference
private void Defendtimer_Tick(object sender, object e)...
/// <summary> פעולה אשר גורמת לשחקן לירות
1 reference
internal void Shoot()...
```

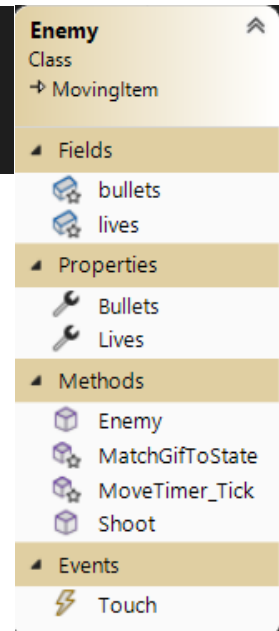
מחלקת Enemy.cs

המחלקה הינה מחלקה מופשטת שממנה יורשות מחלקות נוספות. תפקיד המחלקה הוא להיות הבסיס לכל האויבים שיבנו ויוגדרו על מסך המשחק. בנוסף המחלקה הזאת היא אויב מדרגה ראשונה.



תכונות המחלקה

```
public new event EventHandler Touch;
protected List<Bullet> bullets;
1 reference
public List<Bullet> Bullets { get { return bullets; } }
protected int lives;
```



פעולות המחלקה:

```
/// <summary> פעולה בונה של המחלקה
3 references
public Enemy(double placeX, double placeY, int speed, Canvas arena, int size) ...
/// <summary> ... פעולה שנקראת כל אלפית שנייה. הפעולה מבצעת את הפעולה הקדומה שלה וכן
7 references
protected override void MoveTimer_Tick(object sender, object e) ...
/// <summary> פעולה אשר מתאימה את התמונה של האויב למצב הנוכחי שלו
3 references
protected virtual void MatchGifToState() ...
/// <summary> פעולה אשר גורמת לאויב לירות ירייה
3 references
public virtual void Shoot() ...
```

מחלקת GreenEnemy.cs

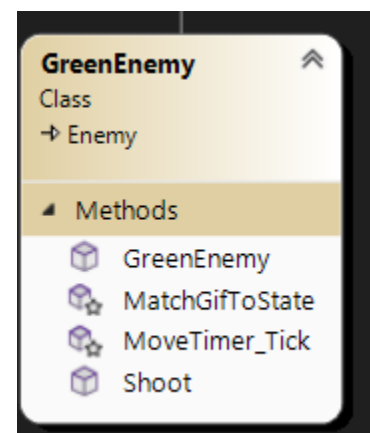
מחלקה זו היא מחלקה היורשת מ cs.Enemy ותפקידה לבנות ולהגדיר את האויב השני.

***מחלקה זאת לא מרחיבה משתנים



פעולות המחלקה:

```
public GreenEnemy(double placeX, double placeY, int speed, Canvas arena, int size) ...  
/// <summary> פעולה הנקראת כל אלפית שנייה. הפעולה מזמנת את הפעולה הקדומה לה  
6 references  
protected override void MoveTimer_Tick(object sender, object e) ...  
/// <summary> פעולה אשר מעדכנת את התמונה של האויב על פי מספר החיים שלו  
2 references  
protected override void MatchGifToState() ...  
/// <summary> פעולה אשר גורמת לאויב לירות ירייה  
2 references  
public override void Shoot() ...
```



מחלקת RedEnemy.cs:

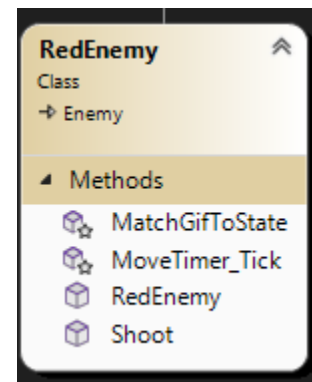
מחלקה זו היא מחלקה היורשת מ cs.Enemy ותפקידה לבנות ולהגדיר את האויב השלישי.

***מחלקה זאת לא מרחיבה משתנים

פעולות המחלקה:



```
/// <summary> פעולה בונה של המחלקה
1 reference
public RedEnemy(double placeX, double placeY, int speed, Canvas arena, int size) ...
/// <summary> פעולה הנקראת כל אלפית שנייה. הפעולה מזמנת את הפעולה הקדומה לה
6 references
protected override void MoveTimer_Tick(object sender, object e) ...
/// <summary>
/// פעולה אשר מעדכנת את התמונה של האויב על פי מספר החיים שלו
/// </summary>
2 references
protected override void MatchGifToState() ...
/// <summary> פעולה אשר גורמת לאויב לירות שלוש יריות לקיוונים שונים
2 references
public override void Shoot() ...
```



מחלקת Enemies.cs

מטרת המחלקה היא לגרום לכל האוייבים על ההמסך לפעול באופן מסוכרן

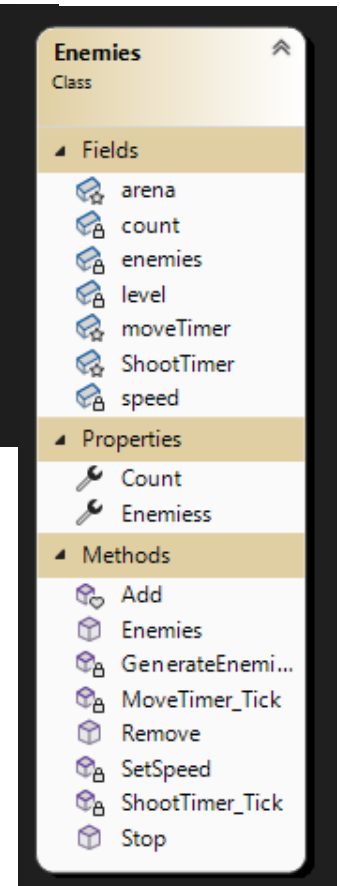
תכונות המחלקה:

```
private int speed;
private int level;
Enemy[,] enemies;

private int count;

2 references
public int Count
{
    get { return count; }
    set { count = value; }
}

9 references
public Enemy[,] Enemies { get { return enemies; } }
protected Canvas arena; //זירת המשחק
protected DispatcherTimer moveTimer; //הטיימר שאחראי על תנועת הדמויות
protected DispatcherTimer ShootTimer; //הטיימר שאחראי על תנועת הדמויות
```



פעולות המחלקה:

```
/// <summary> פעולה בונה
2 references
public Enemies(int numrow, int numcollum, int speed, Canvas arena, int level)...
/// <summary> ... פעולה מייצרת מערת של אויבים ומקנה את כמות האויבים של כל רמה לפי
1 reference
private void GenerateEnemiesByLevels(int level)...
/// <summary> פעולה אשר מוחקת את האויב
1 reference
public void Remove(ref Enemy e)...
/// <summary> פעולה הנקראת כל 2 שניות ואחראית על גרימת ירייה של אויב רגומלי
1 reference
private void ShootTimer_Tick(object sender, object e)...
/// <summary> פעולה הנקראת כל אלפית שנייה ואחריה על הזזת האויבים קבוצה
1 reference
private void MoveTimer_Tick(object sender, object e)...
/// <summary> פעולה אשר אחראית על השמת המהירות לכל האויבים
2 references
private void SetSpeed(double speed)...
/// <summary> פעולה אשר עוצרת את כל האויבים
1 reference
public void Stop()...
/// <summary> פעולה אשר מוסיפה אויב למיקום שניתן לפעולה
0 references
internal void Add(int row, int collum)...
```


סראקט Point

מבנה זה מתאר נקודה על המסך

```
struct Point
{
    public int x;
    public int y;
    /// <summary> פעולה בונה
    2 references
    public Point(int x, int y)...
```

מחלקת MovingItem.cs

מחלקה זאת מייצגת אובייקט אשר זז על המסך. מחקה זאת היא המחלקה הבסיסית אשר כל המחלקות של אובייקטים הזזים יורשות ממנה.

תכונות המחלקה:

```
protected double placeX; // מיקום אופקי
protected double placeY; // מיקום אנכי
protected Image image; // מראה הדמות
protected Canvas arena; // זירת המשחק
protected double speedX; // מהירות אופקית
protected double speedY; // מהירות אנכית
protected bool destroyed;
protected DispatcherTimer moveTimer; // הטיימר שאחראי על תנועת הדמות
public event EventHandler Touch;

5 references
public double PlaceX { get { return placeX; } set { placeX = value; } }
11 references
public double PlaceY { get { return placeY; } set { placeY = value; } }
1 reference
public double SpeedX { get { return speedX; } set { speedX = value; } }
3 references
public double SpeedY { get { return speedY; } set { speedY = value; } }
12 references
public Image Image { get { return image; } set { image = value; } }
5 references
public bool Destroyed { get { return destroyed; } set { destroyed = value; } }
```

MovingItem

Class

Fields

arena

destroyed

image

moveTimer

placeX

placeY

speedX

speedY

Properties

Destroyed

Image

PlaceX

PlaceY

SpeedX

SpeedY

Methods

GetRectangle

GoLeft

GoRight

MoveTimer_Tick

MovingItem

Stop

Events

Touch

פעולות המחלקה:

```
/// <summary> פעולה בונה
3 references
public MovingItem(double placeX, double placeY, Canvas arena, int size)...
/// <summary> פעולה אשר מחזירה אובייקט מסוג מלבן שמייצג את היקפו של האובייקט
4 references
public virtual Rect GetRectangle() => new Rect((int)this.placeX + 2, (int)this.placeY, (int)this.Image.Height, (int)this.Image.Width);
/// <summary> פעולה אשר נקראת כל אלפית שנייה ואחראית על לבדוק את גבולות האובייקט
9 references
protected virtual void MoveTimer_Tick(object sender, object e)...
/// <summary> פעולה אשר גורמת לאובייקט לזוז ימינה
2 references
public void GoRight()...
/// <summary> פעולה אשר גורמת לאובייקט לזוז שמאלה
2 references
public void GoLeft()...
/// <summary> פעולה אשר גורמת לאובייקט לעזור
2 references
public virtual void Stop()...
```

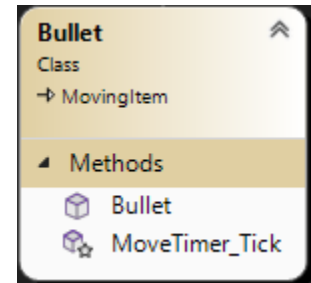
מחלקת Bullet.cs

מחלקה שתפקידה היא לבנות ולהגדיר את הכדור הנורה ע"י דמות השחקן במשחק.

***למחלקה זו אין תכונות המתווספות לMovingItem.

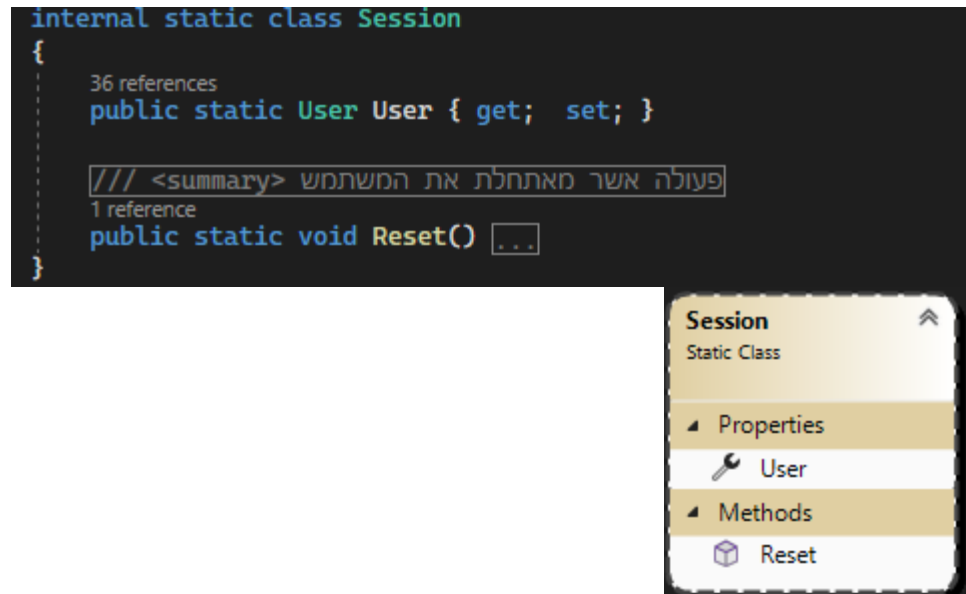
פעולות המחלקה:

```
/// <summary> פעולה אשר בונה את המחלקה
6 references
public Bullet(double placeX, double placeY, Canvas arena, int size, bool IsPlayerShooting, int speedX) ...
/// <summary> פונקציה הנקראת כל אלפית שנייה ומטרתה היא להציג את הכדור
3 references
protected override void MoveTimer_Tick(object sender, object e) ...
```



מחלקת Session.cs

מחלקה זו היא מחלקה סטטית אשר מכילה אובייקט מטיפוס משתמש. מטרת המחלקה הזו היא לאחסן בזיכרון המשחק את פירטי המשתמש.



דפי המשחק (כחלק מהמשחק Space Invaders)

מחלקת דף הבית MainPage.cs

מחלקה זו מייצגת את הדף הראשי, משם אפשר לגשת לכל דפי המשחק.

פעולות המחלקה:

```
/// <summary> פעולה בונה
0 references
public MainPage()...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על הכפתור משחק חדש
1 reference
private void playBtn_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור היציאה
1 reference
private void Quit_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור השיאים
1 reference
private void highScoreBtn_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור המידע
1 reference
private void informationBtn_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור היציאה מהמשחק
1 reference
private void logout_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת בכל פעם שהעכבר נכנס לתכום של הכפתור
3 references
private void playBtn_PointerEntered(object sender, PointerRoutedEventArgs e)...
/// <summary> פעולה אשר נקראת בכל פעם שהעכבר יוצא מתכום של תיבת הכפתור
3 references
private void playBtn_PointerExited(object sender, PointerRoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור המחסן
1 reference
private void lockerpbtn_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור החנות
1 reference
private void shoprpbtn_Click(object sender, RoutedEventArgs e)...
```

מחלקת הדף GamePage.cs

מחלקה זו מייצגת את דף המשחק של האפליקציה, באמצעות פעולות המחלקה המשחק ניבנה בשביל שהמשתמש יכול לשחק ולהרוויח נקודות ולשחק בנעימות.

תכונות המחלקה:

```
private Manager myManager;  
private int points;
```

דף:

The screenshot displays the source code of the `GamePage` class in a dark-themed editor. The code includes several methods with XML documentation comments in Hebrew. The sidebar on the right shows the class hierarchy for `GamePage`, which is a sealed class and a page. It lists the fields `myManager` and `points`, and the methods `AddPoints`, `Button_Click`, `Button_Click_1`, `CoreWindow_KeyDown`, `CoreWindow_KeyUp`, `DefendKeyDown`, `GamePage`, `MyManager_updateDefendTime`, `Page_Loaded`, and `RemoveLife`.

```
/// <summary> פעולה בונה  
0 references  
public GamePage()...  
/// <summary> פעולה אשר מוסיפה נקודות לטקסט של הנקודות  
2 references  
private void AddPoints(object sender, EventArgs e)...  
/// <summary> פונקציית נקראת כאשר דף נטען והפונקציה אחראית על שיוך פונקציות לאיר ...  
1 reference  
private void Page_Loaded(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על המקלדת ובודקת אם הוא לחץ על הכ ...  
2 references  
private void DefendKeyDown(CoreWindow sender, KeyEventArgs args)...  
/// <summary> פעולה אשר מעדכנת את הטקסט של זמן החסינות  
2 references  
private void MyManager_updateDefendTime(object sender, EventArgs e)...  
/// <summary> פעולה אשר מעדכנת את התקסט של מספר החיים  
2 references  
private void RemoveLife(object sender, EventArgs e)...  
/// <summary> פונקצייה אשר נקראת כאשר המשתמש עוצב את מקשי המקלדת  
2 references  
private void CoreWindow_KeyUp(Windows.UI.Core.CoreWindow sender, Windows.UI.Core.KeyEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על המקלדת  
2 references  
private void CoreWindow_KeyDown(Windows.UI.Core.CoreWindow sender, Windows.UI.Core.KeyEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על הכפתור חזרה לבית  
1 reference  
private void Button_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על הכפתור נסה שוב  
1 reference  
private void Button_Click_1(object sender, RoutedEventArgs e)...
```

דף מחלקת HighScores.cs

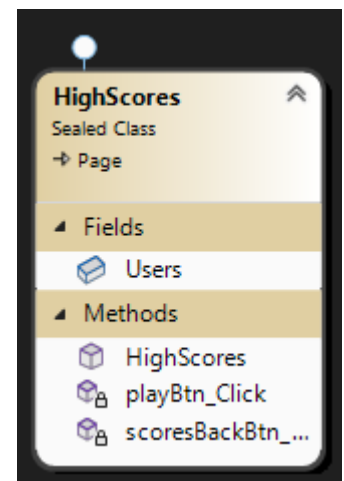
מחלקה זו מייצגת דף אשר מציג את טבלת השיאים של המשחק הממוינים בסדר יורד לפי הניקוד הגבוהה ביותר

תכונות המחלקה:

```
public List<User> Users;
```

פעולות המחלקה:

```
/// <summary> פעולה בונה  
0 references  
public HighScores()...  
/// <summary> פעולה אשר נקראת כאשר השחקן לוחץ על הכפתור חזרה הביתה  
1 reference  
private void scoresBackBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקרא כאשר השחקן לוחץ על הכפתור משחק חדש  
1 reference  
private void playBtn_Click(object sender, RoutedEventArgs e)...
```

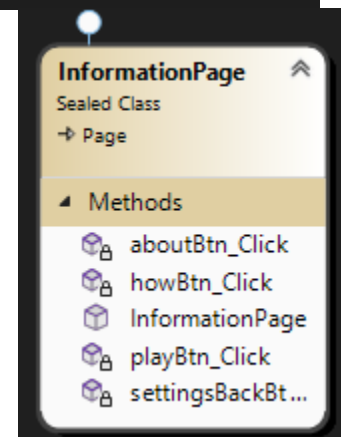


מחלקת דף InformationPage.cs:

מחלקה זו מייצגת דף אשר מציג מידע על מתכנת המשחק, וגם מידע ע איך לשחק במשחק.

פעולות המחלקה:

```
/// <summary> פעולה בונה  
0 references  
public InformationPage()...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כתור האין לשחק  
1 reference  
private void howBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על הכפתור מידע עלי  
1 reference  
private void aboutBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על העפתור חזרה הביתה  
1 reference  
private void settingsBackBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקרא כאשר השחקן לוחץ על הכפתור משחק חדש  
1 reference  
private void playBtn_Click(object sender, RoutedEventArgs e)...
```

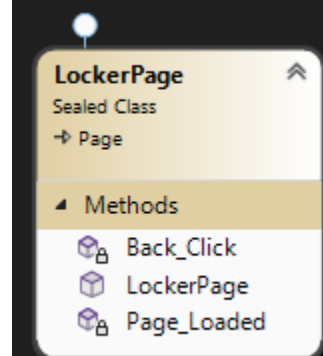


מחלקת דף LockerPage.cs:

מחלקה זו מייצגת דף אשר מציג את הסקינים הקנויים של המשתמש

פעולות המחלקה:

```
/// <summary> פעולה בונה
0 references
public LockerPage()...
/// <summary> פעולה אשר נקראת כאשר העמוד נטען
1 reference
private void Page_Loaded(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על העפתור חזרה הביתה
1 reference
private void Back_Click(object sender, RoutedEventArgs e)...
```



מחלקת דף Login.cs:

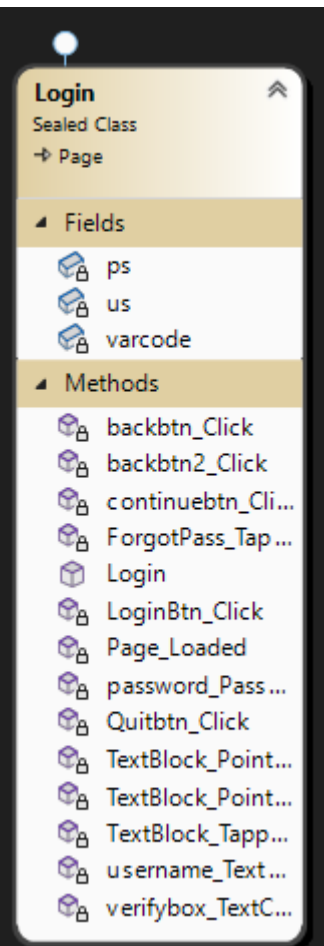
מחלקה זו מייצגת דף אשר מאפשר למשתמשים להיכנס לחשבונם

תכונות המחלקה:

```
bool us = false;  
bool ps = false;  
private string varcode;
```

פעולות המחלקה:

```
/// <summary> פעולה בונה  
0 references  
public Login()...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור הכניסה למשתמש  
1 reference  
private void LoginBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כל פעם שתיבת הטקסט של השם משתמש משתנה  
1 reference  
private void username_TextChanged(object sender, TextChangedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור היציאה  
1 reference  
private void Quitbtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על תיבת הטקסט של הרשמה  
1 reference  
private void TextBlock_Tapped(object sender, TappedRoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת בכל פעם שהעכבר יוצא מתוכו של תיבת טקסט  
6 references  
private void TextBlock_PointerEntered(object sender, PointerRoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת בכל פעם שהעכבר יוצא מתוכו של תיבת טקסט  
6 references  
private void TextBlock_PointerExited(object sender, PointerRoutedEventArgs e)...  
  
/// <summary> פעולה אשר נקראת כל פעם שתיבת הסיסמא משתנה  
1 reference  
private void password_PasswordChanged(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר הדף טען  
1 reference  
private void Page_Loaded(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על תיבת הטקסט של שכחתי סיסמא  
1 reference  
private void ForgotPass_Tapped(object sender, TappedRoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על העפתור חזרה הביתה  
1 reference  
private void backbtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על הלחצן המשך  
1 reference  
private void continuebtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כל פעם שתיבת הטקסט של האישור קוד משתנה  
1 reference  
private void verifybox_TextChanged(object sender, TextChangedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על העפתור חזרה השני הביתה  
1 reference  
private void backbtn2_Click(object sender, RoutedEventArgs e)...
```



דף מחלקת Register.cs:

מחלקה זו מייצגת דף אשר מאפשר למשתמשים לפתוח חשבון על מנת לשחק ולהשתמש בפיצרים של המשחק

תכונות המחלקה: `private string email = "";`

פעולות המחלקה:

```
/// <summary> פעולה בונה  
0 references  
public ResetPasswordPage()...  
/// <summary> פעולה אשר נקראת כאשר המעוד נטען אחרי שעברו אליו  
0 references  
protected override void OnNavigatedTo(NavigationEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור היציאה  
1 reference  
private void Quitbtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור האיתחול סיסמא  
1 reference  
private void resetbtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כל פעם שתיבת הסיסמא משתנה וגם תיבת האישור סיסמא  
2 references  
private void newpass_PasswordChanged(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר בודקת את עמידות הסיסמא  
1 reference  
public bool CheackPassword(out string exception)...
```

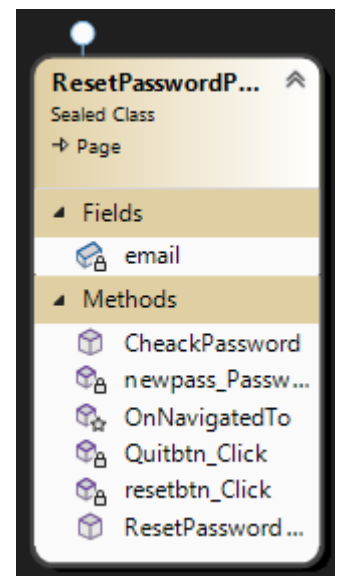
מחלקת דף ResetPasswordPage.cs

מחלקה זו מייצגת דף אשר מעניק למשתמש אפשרות לעדכן את סיסמתו לאחר ששכח אותה.

תכונות המחלקה: `private string email = "";`

פעולות המחלקה:

```
public ResetPasswordPage()...
/// <summary> פעולה אשר נקראת כאשר המעוד נטען אחרי שעברו אליו
0 references
protected override void OnNavigatedTo(NavigationEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור היציאה
1 reference
private void Quitbtn_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור האיתחול סיסמא
1 reference
private void resetbtn_Click(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר נקראת כל פעם שתיבת הסיסמא משתנה וגם תיבת האישור סיסמא
2 references
private void newpass_PasswordChanged(object sender, RoutedEventArgs e)...
/// <summary> פעולה אשר בודקת את עמידות הסיסמא
1 reference
public bool CheackPassword(out string exception)...
```

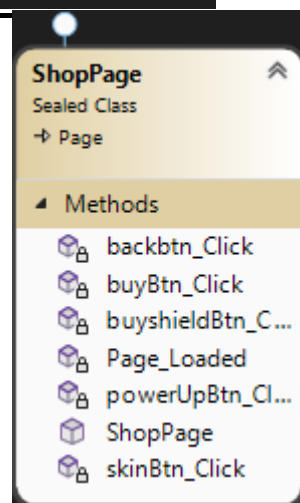


מחלקת דף ShopPage.cs

מחלקה זו מייצגת דף אשר המשתמש יכול לקנות סקינים שיראה במשחק עצמו ומגן אשר יגן עליו מפגיעות

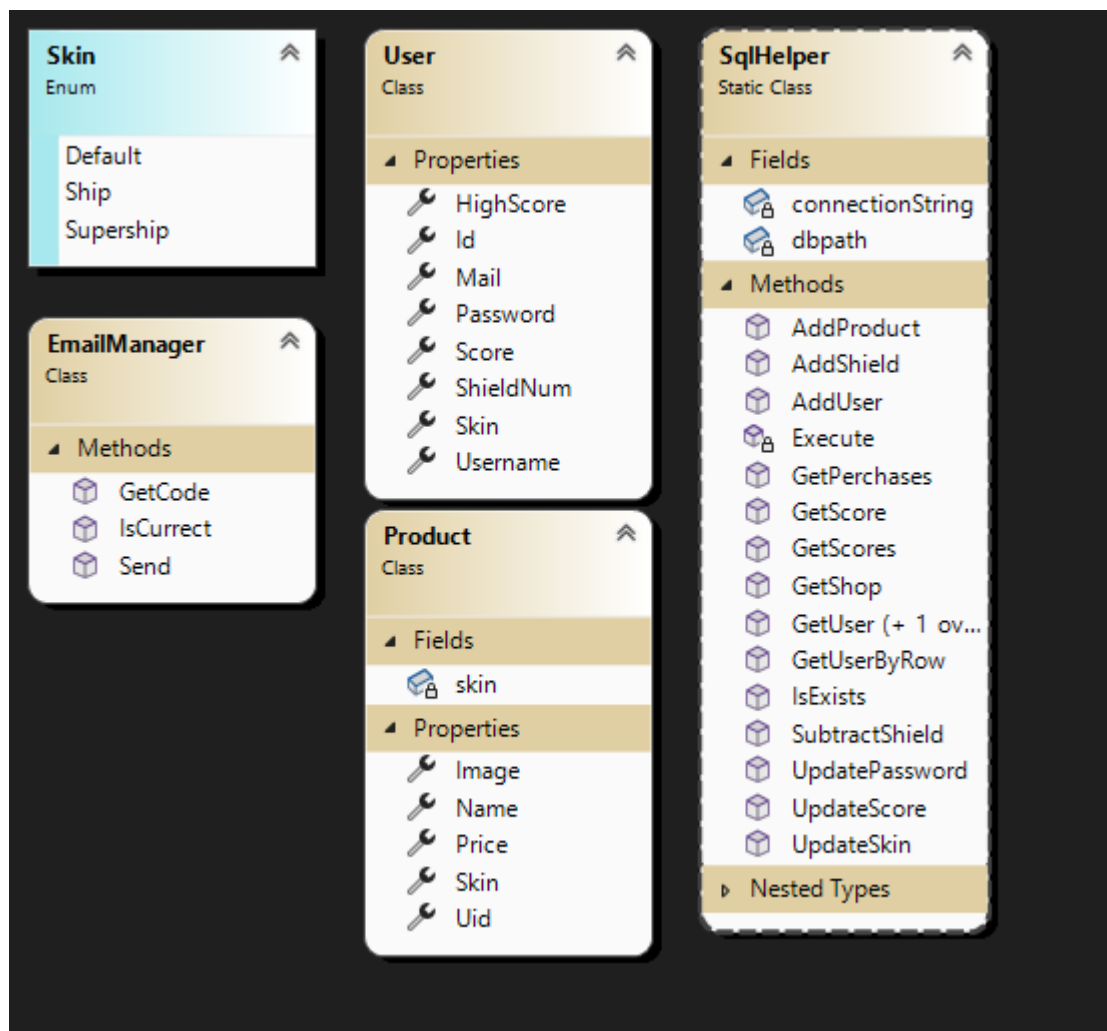
פעולות המחלקה:

```
/// <summary> פעולה בונה  
0 references  
public ShopPage()...  
/// <summary> פעולה אשר נקראת כאשר העמוד נטען  
1 reference  
private void Page_Loaded(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על העפתור חזרה הביתה  
1 reference  
private void backbtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור קניית הסקין  
1 reference  
private void buyBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור החלפת הצפייה לסקינים  
1 reference  
private void skinBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור ההחלפה למגן  
1 reference  
private void powerUpBtn_Click(object sender, RoutedEventArgs e)...  
/// <summary> פעולה אשר נקראת כאשר המשתמש לוחץ על כפתור קניית המגן  
1 reference  
private void buyshieldBtn_Click(object sender, RoutedEventArgs e)...
```



DataBase

דיאגרמת הפרוייקט:



מחלקת SqlHelper.cs

מחלקה האחראית על החיבור של מסד הנתונים לתוכנה. ועל שיכתוב ועדכון הנתונים בה.

תכנון המחלקה:

```
static string dbpath = ApplicationData.Current.LocalFolder.Path; // מיקום מסד הנתונים
private static string connectionString = "Filename=" + dbpath + "\\MyDatabase.db"; // מחזורת הקישור עם מסד הנתונים
```

פעולות המחלקה:

```
/// <summary> פעולה אשר מבצעת שאילתה אשר היא מקבלת </summary>
7 references
private static void Execute(string query)...
/// <summary> פעולה אשר מחזירה רשימה של משתמשים המסודרים לפי שיום הגבוה במשחק </summary>
1 reference
public static List<User> GetScores()...
/// <summary> ... פעולה אשר מחזירה את המשתמש אשר קיים במאגר הנתונים, אם המשתמש לא קי </summary>
2 references
public static User GetUser(string username, string password, string mail)...
/// <summary> פעולה אשר מחזירה את הניקוד הכללי של השחקן </summary>
0 references
public static int GetScore(int Id)...
/// <summary> ... מחזיר את הידור על פי סוג העמודה (שם משתמש, סיסמה, מיל ועוד). לפי א </summary>
1 reference
public static User GetUserByRow(UserRowType type, string value)...
/// <summary> פעולה אשר בודקת אם המשתמש קיים במסד הנתונים לפי העמודה </summary>
3 references
public static bool IsExists(UserRowType type, string value)...
/// <summary> ... הפעולה מחזירה את המשתמש הקיים במבנה הנתונים על פי ערכים שהוכנסו אם </summary>
1 reference
public static User GetUser(string username, string password)...
/// <summary> פעולה אשר מחזירה רשימה של אובייקטים מטיפוס מוצר אשר המשתמש לא רכש </summary>
1 reference
public static List<Product> GetShop(int Id)...
/// <summary> פעולה אשר מחזירה רשימה של אובייקטים מטיפוס מוצר אשר המשתמש רכש </summary>
1 reference
public static List<Product> GetPerchases(int Id)...
/// <summary> פעולה אשר יוצרת משתמש במסד הנתונים </summary>
1 reference
public static User AddUser(string username, string password, string mail)...
/// <summary> פעולה אשר עדכנת את מספר הסקין של המשתמש על פי המספר המזהה שלו </summary>
1 reference
public static void UpdateSkin(int Id, int skinId)...
/// <summary> פעולה אשר מעדכנת את הניקוד הכללי והניקוד הגבוה של השחקן </summary>
3 references
public static void UpdateScore(int Id, int score, int highscore)...
/// <summary> פעולה אשר יוצרת רכישה של מוצר </summary>
2 references
public static void AddProduct(int Id, int productId)...
/// <summary> פעולה אשר מעדכנת את הסיסמה של המשתמש לפי המספר המזהה </summary>
1 reference
public static void UpdatePassword(int Id, string newpass)...
/// <summary> פעולה אשר מוסיפה 1 למספר המגנים של המשתמש </summary>
1 reference
public static void AddShield(int id)...
/// <summary> פעולה אשר מחסירה 1 ממספר המגנים של המשתמש </summary>
1 reference
public static void SubtractShield(int id)...
/// <summary> אינם המייג את עמודות הטבלה של המשתמשים </summary>
6 references
public enum UserRowType...
```

מחלקת EmailManager.cs

מחלקה זו אחראית על שליחת מיילים ומשתמש ובדיקת תקינות המייל.

פעולות המחלקה:

```
/// <summary> length פעולה אשר מחזירה קוד רנדומלי באורך  
2 references  
public static string GetCode(int length)...
```

```
/// <summary> פעולה אשר שולחת מייל  
2 references  
public static void Send(string email, string subject, string body)...
```

```
/// <summary> פעולה אשר בודקת אם המייל תקין  
1 reference  
public static bool IsCorrect(string mail)...
```


מחלקת User.cs

מחלקה זו מייצגת משתמש על פי מאגר הנתונים

תכונות המחלקה:

```
14 references
public int Id { get; set; } // מספר מזהה
5 references
public string Username { get; set; } // שם משתמש
3 references
public string Password { get; set; } // סיסמה
3 references
public string Mail { get; set; } // מייל
16 references
public int Score { get; set; } // ניקוד כללי
11 references
public int HighScore { get; set; } // שיא הניקוד
6 references
public Skin Skin { get; set; } // סקין (מראה)
8 references
public int ShieldNum { get; set; } // מספר מגנים

public enum Skin
{
    Default,
    Ship,
    Supership
}
```

טבלה במסד הנתונים:

מספרו של המשתמש	שם המשתמש	סיסמת המשתמש	המייל של המשתמש	הניקוד הכללי של המשתמש	שיאו של המשתמש	המספר הסידורי של הסקין הנוכחי של המשתמש	מספר המגנים של המשתמש
Id	Username	Password	Mail	Score	HighScore	SkinId	ShieldNum
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
18	matan	123	matan793793793@gmail.com	40	1000	2	991
23	asd	matanM793	s32479@nomishemer.ort.org.il	0	200	0	4
25	er	miravM123	matanepic793@gmail.com	0	0	0	0

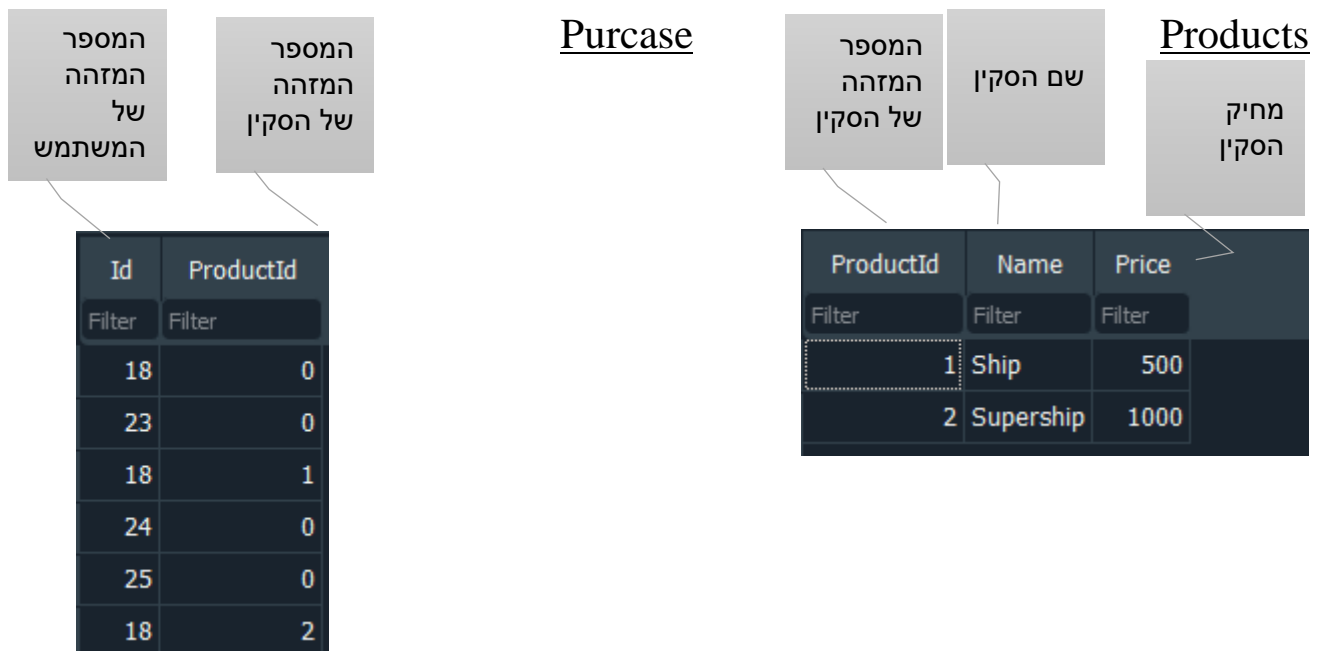
מחלקת Product.cs

מחלקה זו מייצגת מוצר בין עם זה בחנות ובין עם זה במחסן(מפני שבמחסן אין חשיבות למחיר המחיר יהיה 0) עשיתי דבר זה לצורכי נוכיות ויעילות.

תכונות המחלקה:

```
public int Uid { get; set; } /// מספר המזהה של המשתמש
private Skin skin; /// מזהה הסקין הפרטי
6 references
public Skin Skin { get { return skin; } // מזהה הסקין הפומבי
    set
    {
        skin = value;
        Image = new Image();
        Image.Width = 100;
        Image.Height = 100;
        Image.Source = new BitmapImage(new Uri($"ms-appx:///Assets/Spaceship1/{value}.png"));
    }
}
6 references
public Image Image { get; set; } /// תמונת הסקין
2 references
public string Name { get; set; } /// שם הסקין
4 references
public int Price { get; set; } /// מחיר הסקין
```

טבלאות במאגר הנתונים



איכות הפרויקט:

בעיות אלגוריתמיות:

במהלך העבודה נאלצתי לפתור מספר בעיות בפרויקט, המרכזית ביניהן הייתה פתיחת הפרויקט והרצתו, מפני שאת הפרויקט יצרתי בתוך דיסק און קי הייתי צריך להריץ אותו ממספר מחשבים שונים ובכך הקושי של הבעיה גדל.

בנוסף בעמוד המשחק כששמתי את ה canvas בגריד והקטנתי את המשח האויבים ירדו מתחת למסך.

הצעות לשיפור

- הוספת מצבי משחק נוספים שבו אחד מהם הוא מצב בוא שחקן אחד נלחם מול שחקן שני
- משחק בו שני שחקנים משחקים דרך הרשת
- הוספת שידרוגים נוספים בנוסף למגן

רפלקציה

בתחילת הדרך כשמעתי שאנחנו מתבקשים ליצור פרויקט גמר של משחק ל-5 יחידות הנוספות במדעי המחשב היו לי מלא רעיונות, שאיפות וחששות שאני לא יצליח לעמוד בכל המטרות שהצבתי לעצמי.

הפחד הכי גדול שלי במהלך העבודה היה שאני לא יצליח להגיש את הפרויקט בזמן ההגשה ושאני לא יהיה מרוצה ממנו, בסופו של דבר פחדים אלו הניעו אותי וגרמו לי להעלות המוטיבציה לעשות את הפרויקט הכי טוב שאני יכול.

במהלך הפרוייקט נתקלתי בבעיות קשות שאפילו אחת מהן סיכנה את הגשת הפרויקט שלי אך בעזרת מורי אולג עברתי אותה בשלום.

הבנתי שאני יכול ליותר ממה שציפיתי מעצמי ושיכולותי גדולות.

במהלך כל תקופת הפרויקט ולפניו אולג עזר לי באופן משמעותי מאוד, בלי עזרתו והדרכתו לא הייתי מצליח להמשיך להתקדם בפרויקט ולהגיע למצב שאני נימצא בו היום.

מקורות

המשחק מעוצב באומנות מקורית ובאנימציה שרובן נוצרו על ידי בעזרת שתי התוכנות:
Photoshop ו- Adobe Illustrator

שאר המקורות שנעזרתי בהם להכנת הפרויקט:

- אולג כצנלסון - המורה שלנו למקצוע מדעי המחשב.
- גוגל תמונות.
- חברי למגמה