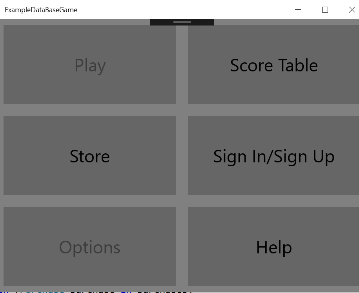
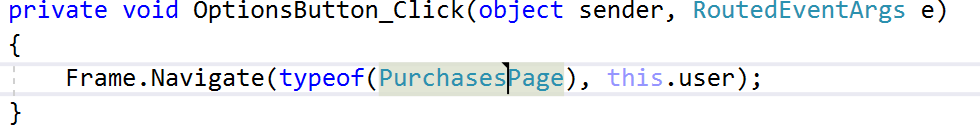
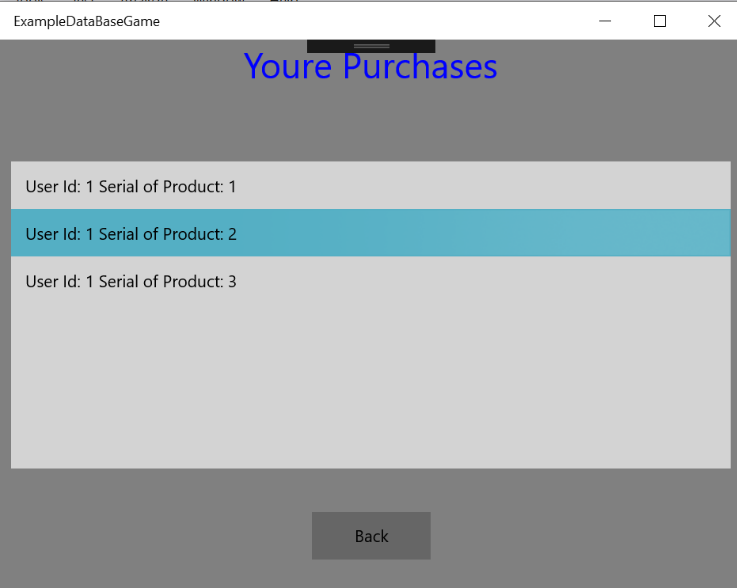
**דף עבודה מס' 1.**

**"המשתמש נרשם, חוזר לתפריט הראשי ואז בוחר באפשרות: Options. ייפתח דף הרכישות הקודמות שלו (המחסן), אשר באמצעותו יוכל לבחור בדמות או בשלב בין הדמויות או השלבים שנמצאים ברשותי, כלומר שרכש בעבר. לאחר מכן חוזר המשתמש שוב לתפריט הראשי ואז הולך לשחק עם הדמות שבחר".**

שלבי הביצוע:

1. בתפריט הראשי להכין לחצן Options:

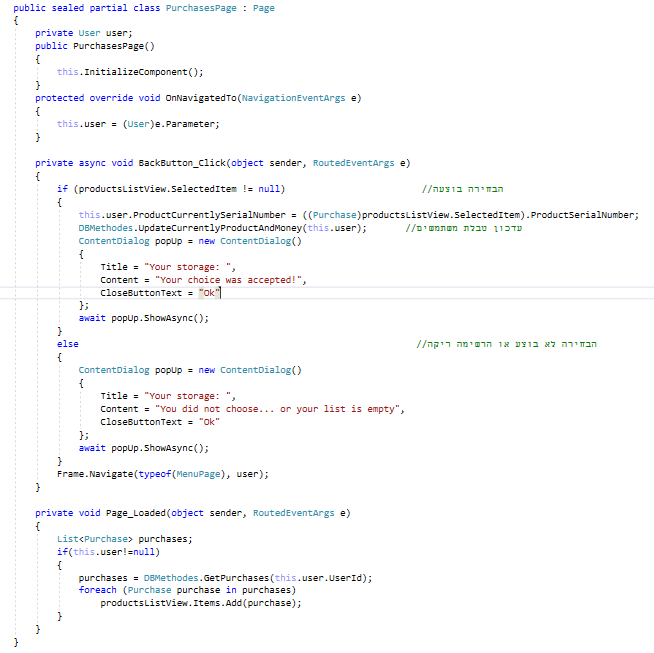
2. ב- MenuPage.cs לתכנת פעולה Click, אשר תגיב ללחיצה על הלחצן Options:

3. להוסיף דף חדש בשם PurchasesPage (דף הרכישות(. הדף יציג את הרכישות הקודמות של המשתמש שמשחק כעת. זה המראה שלו:

4. להלן קוד העיצוב של הדף XAML)):

ה- **ListView** זאת רשימה שאמורה להציג את כל המוצרים או השלבים שנרכשו בעבר.

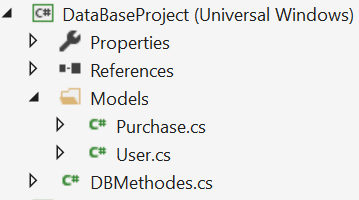
5. להלן קוד מאחור (code behind) של הדף: PurchasesPage.cs:



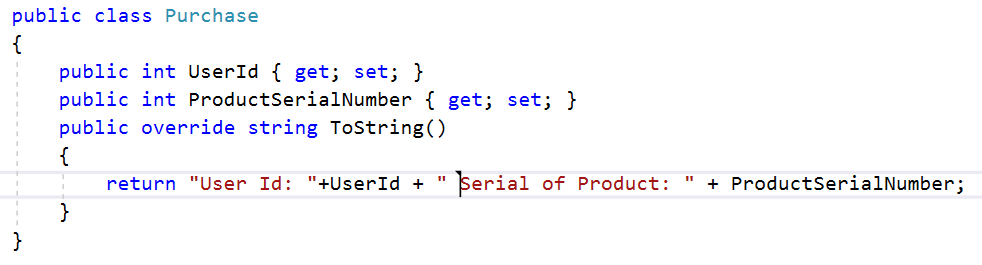
6. בפעולה Page\_Loaded, כפי שנראה בצילום, מופעלת הפעולה:

purchases = DBMethodes.GetPurchases(this.user.UserId);

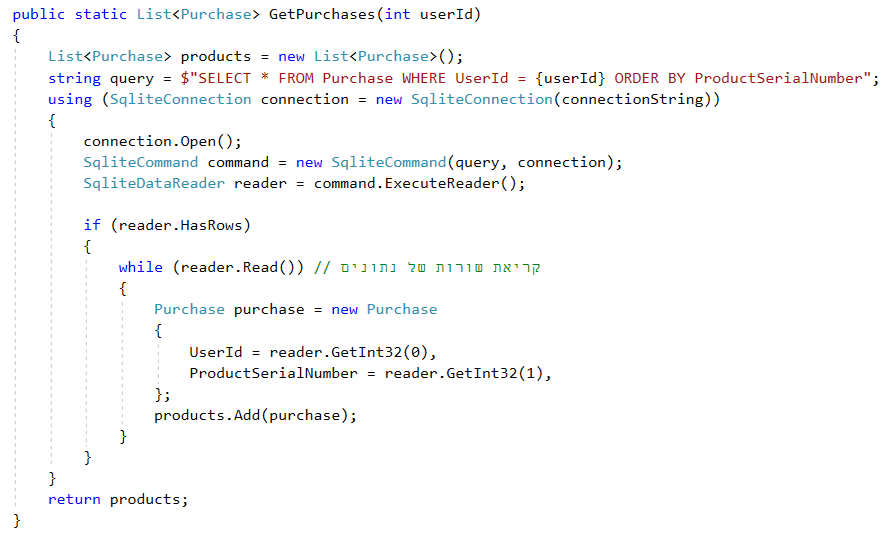
נעבור כעת לפרויקט השני: DataBaseProject.

7. נוסיף מחלקה חדשה Purchase.cs בתיקיה Models בדומה למחלקה User.cs:

זה תוכן המחלקה:

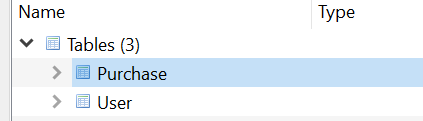


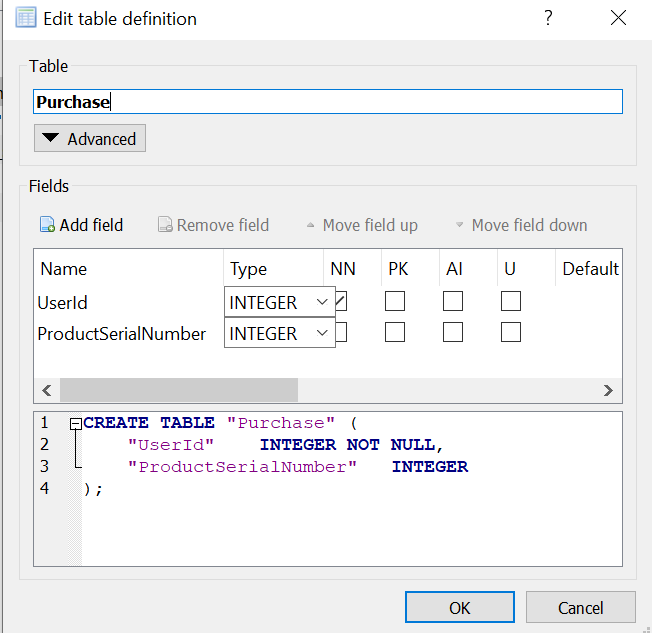
8. כעת נעבור למחלקה סטטית DataBaseMethods.cs. נוסיף במחלקה פעולה הבאה:

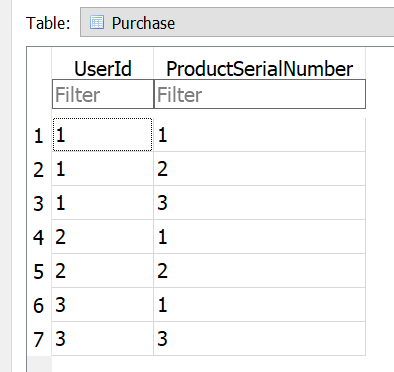
הפעולה מבצעת שאילתה:

string query = $"SELECT \* FROM Purchase WHERE UserId = {userId} ORDER BY ProductSerialNumber";

פירוש הדבר: מתבצעת קריאת כל השורות מהטבלה Purchase של המשתמש userId.

9. מכאן נובעת המסקנה שכעת יש ליצור טבלה בשם הנ"ל ולמלא אותה. נעבור לתוכנית Browser for SQLite, שבאמצעותה הקמנו בעבר טבלה User ונקים טבלה נוספת בשם Purchase.

זה מבנה הטבלה:

10. נמלא את הטבלה שיצרנו:

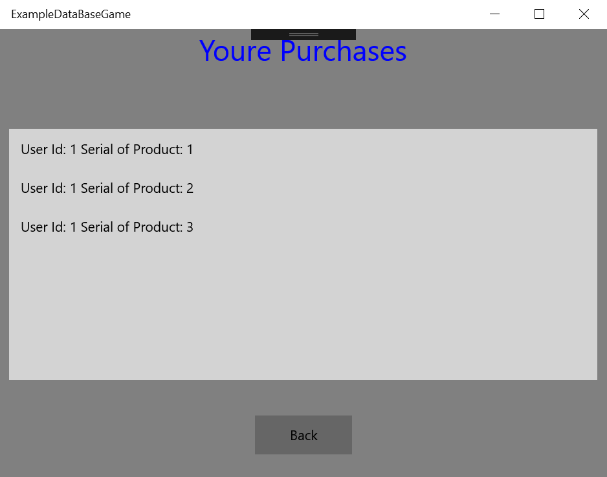
המשמעות היא שבבעלותו של המשתמש 1 נמצאות דמויות שמספריהן 1,2,3. המשתמש 2 רכש בעבר דמויות 1,2, והמשתמש 3 רכש 1,3.

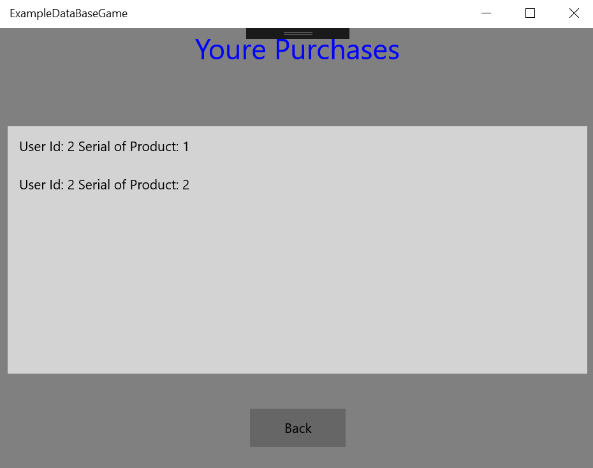
11. נשמור את ה- DB ונחזור לפרויקט של המשחק למחלקה PurchasesPage.cs לפעולה Page\_Loaded. בפעולה הזו רשמנו את השורות האלו:

purchases = DBMethodes.GetPurchases(this.user.UserId);

foreach (Purchase purchase in purchases)

productsListView.Items.Add(purchase);

המשמעות היא שבעת פתיחת הדף תופיע ברשימה בשם productsListView הרשימה של הדמויות של המשתמש רכש. כעת נפעיל את התוכנית עם המשתמש userId=1. זה מה שנראה אחרי שנרשם ונלחץ על הלחצן Options:

לעומת זאת אם נכנס כמשתמש userId=2, אזי זה מה שיופיע:

במילים אחרות, ההצעה תלויה בטבלה Purchase. לכל משתמש תוצע הצעה שונה בהתאם לרכישות הקודמות שלו.

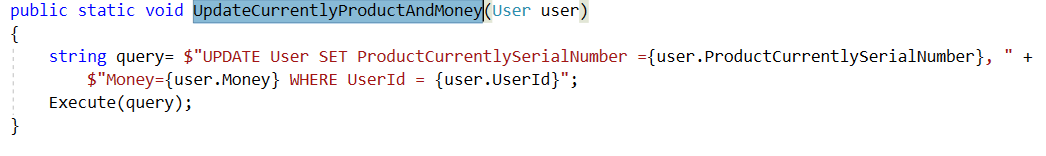
**12**. כעת נותר לבחור את אחת הדמויות ולחזור לתפריט הראשי. נחזור למחלקה PurchasesPage.cs. בדף ישנו מקש Back. יש לרשום את הפעולה:



הפעולה תגיב ללחיצה על הלחצן Back.

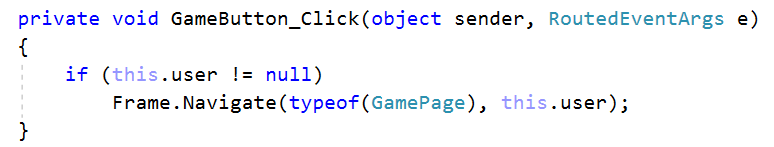
הסבר: השורה הראשונה גורמת לעדכון המשתנה user בהתאם לבחירת השחקן. כעת נותר לבצע עדכון גם בטבלה User ולאחר מכן לחזור לתפריט הראשי. השורה:

DBMethodes.UpdateCurrentlyProductAndMoney(this.user);

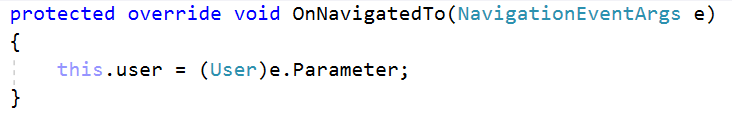
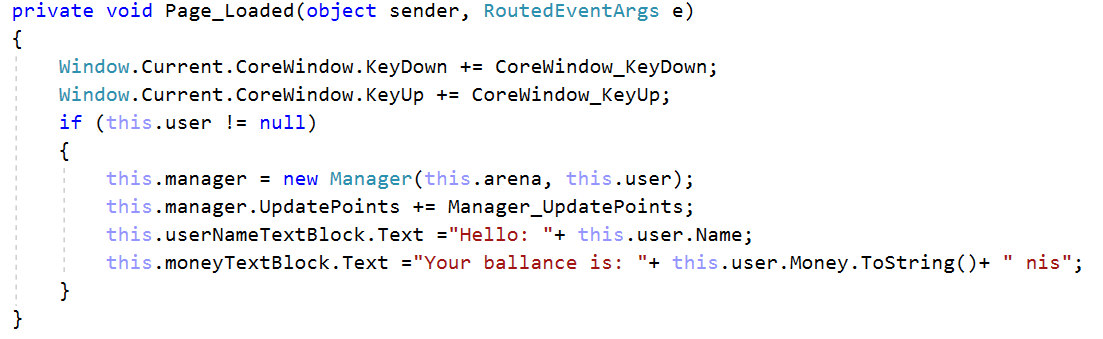
מפעילה את הפעולה UpdateCurrentlyProductAndMoney שנמצאת במחלקה DBMethodes. נעבור למחלקה DBMethodes ונוסיף שם את הפעולה הבאה:

הפעולה תעדכן את הדמות הנוכחית בטבלה User.

13. כעת ברצוני לוודא שזאת הפעולה שנמצאת במחלקה MenuPage.cs ומגיבה ללחיצה על המקש Play:



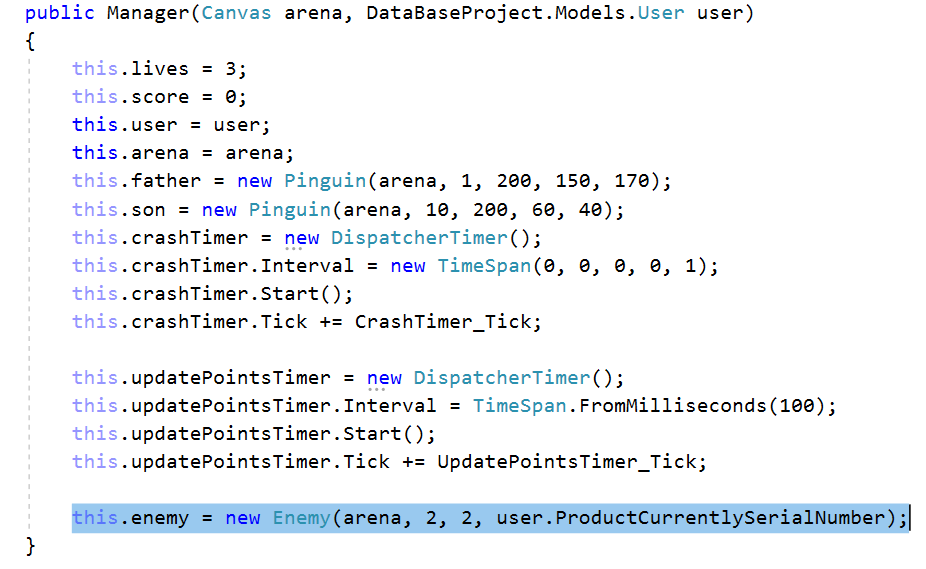
14. במחלקה GamePage.cs צריכה להיות הפעולה הבאה:

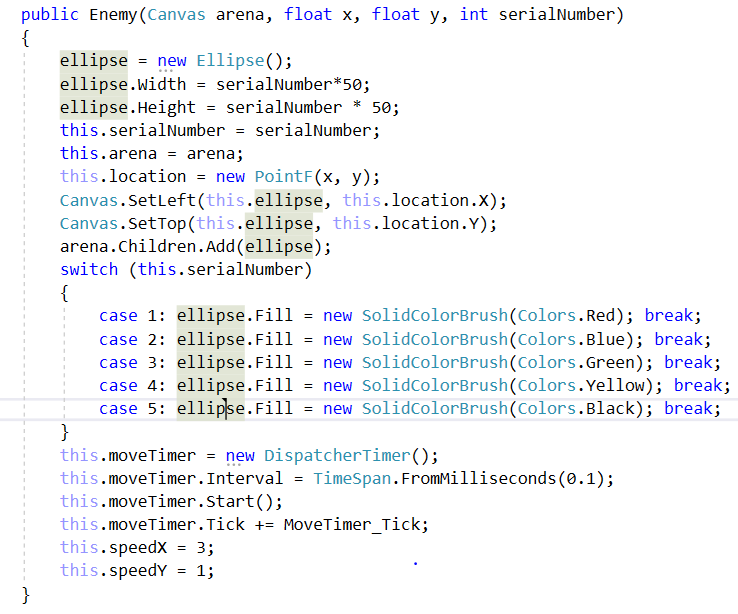
15. באותה המחלקה GamePage.cs צריכה להיות הפעולה הבאה:

הפעולה Page\_Loaded תפעיל Manager ותעביר לו הן את ה- arena והן את ה- user המעודכן, הכולל את בחירת המשתמש בדף Options.

16. נעבור למחלקה Manager.cs:

זאת הפעולה הבונה של Manager:

בשורה המודגשת נבנית דמות התואמת את בחירת המשתמש.

17. נעבור למחלקה Enemy:

כאן ניתן לראות שהדמות מקבלת מספר סידורי ובהתאם לכך בוחרת נראות שונה.