**Architectural pattern - MVC**

הפרויקט שלנו נכתב בארכיטקטורת MVC.

הפרויקט מחולק לשלושה packeges: Model, View, Controller

**Model:**

במודל אנחנו מחזיקים את ה-state הנוכחי של האפליקציה.

ה-state של האפליקציה שלנו ב-runtime הוא מערך של שחקנים שמוצג למשתמש בטבלה. השחקנים נשמרים בקובץ בינארי (ע"י הממשק Serializableשמממשת המחלקה Player) שמצורף לתיקיית הפרויקט. המשתמש רשאי להוסיף שחקן לרשימת השחקנים, ולשמור מחדש את הקובץ (לדרוס את קובץ ה-default)

במחלקה Database (שנמצאת ב-Model) האפליקציה מחזיקה את מערך השחקנים הנוכחי שמוצג למשתמש בטבלה. מחלקה זו מממשת מספר מתודות שקשורות למערך השחקנים כמו כתיבתו לקובץ, קריאה של קובץ בינארי של שחקנים וטעינתו למערך השחקנים במחלקה Database (טעינת ה-state הנוכחי מהקובץ)

**Controller:**

החוט המקשר בין ה-View ל-Model.

יש מופע יחיד של ה-Controller ב-View (במחלקה MainFrame). כל בקשה שקשורה ב-state של האפליקציה (מערך השחקנים) מתבצעת אך ורק דרך ה-Controller. ה-View מעולם לא יפנה ישירות ל-Database ישירות. ה-Controller בעצם מעביר את המידע ל-Database ומתאים אותו לפי מה שה-Model יודע לקבל.

המחלקה MainFrame מחזיקה מופע יחיד של Controller. מחלקה זו "מאזינה" (Observer pattern) לשינויים שמתרחשים בשאר ה-Components של האפליקציה אותם היא מכילה. כאשר מתרחש שינוי שמצריך שינוי ב-State האפליקציה ( = במערך השחקנים), התקשורת מול ה-Database class תתבצע ע"י מתודה ב-Controller.

**View:**

מכיל את ה-gui של האפליקציה. המחלקה MainFrame מהווה את ה-Frame הראשי של האפליקציה שמכיל בתוכו את כל שאר ה-Components (מסך החיפוש, מסך הוספת השחקן, טבלת השחקנים וכו'...)