

קורס מורים מובילים תש"ע, פיתוח הפרויקט: אימאן דיאב

## פרויקטים שהוכנו על-ידי משתתפי קורס מורים מובילים תש"ע

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי

ללא קבלת אישור מראש מצוות הפיתוח

# משחק קלפים

## 21 Card Game

פיתוח הפרויקט : אימאן דיאב

ניתן להשתמש בחומרים לצורך הוראה בלבד.

לא ניתן לפרסם את החומרים או לעשות בהם כל שימוש מסחרי ללא קבלת אישור מראש מצוות הפיתוח.

## תיאור הפרויקט משחק קלפים 21

במשחק הקלפים 21 יש שני שחקנים: המחשב והמשתמש Dealer. הוא שמערבב את הקלפים בתחילת כל משחק ע"י פעולה פנימית במחלקה "דילר" ומחלק לכל שחקן שני קלפים מתוך חבילת הקלפים Deck של המשחק המכילה 52 קלפים – שהוא מערך מסוג "קלף". ערכי הקלפים הם כערכי המספרים שהם מייצגים למעט הקלף Ace שערכו יכול להיות 1 או 11, לפי מצבו של השחקן ולטובתו במשחק. ערכי הקלפים עם תמונה הם 10. כאשר השחקן מחזיק בידו בקלפים Hand, ערכם של הקלפים בידו הוא סכום הקלפים שהוא מחזיק. בתחילת המשחק ה"דילר" מחלק לכל שחקן קלף בכל סיבוב, כל עוד סכומם לא עולה על 21. שחקן יכול להמשיך לבקש מהדילר עוד קלפים עד שהסכום מגיע ל- 21 או יותר. בסוף המשחק "מנצח" מי שסכום הקלפים שבידו הגבוה ביותר **אך קטן** מ 21. במקרה ששני השחקנים עברו את ה- 21, אזי המנצח הוא מי שסכום הקלפים בידו הוא הנמוך ביותר. אם שני הסכומים שווים אז המחשב תמיד מנצח.

קהל יעד: כל מי שמתעניין במשחקי מחשב – קלפים

### מבנה הפרויקט :

הפרויקט בנוי מארבע מחלקות ותוכנית ראשית שתשתמש במערך של עצם (קלף) ותממש את המשחק ע"י לולאת כל-עוד שמאפשרת חזרה על המשחק כל עוד המשתמש רוצה בכך.

מחלקות שייעשה בהן שימוש :

1. מחלקת קלף Card
2. מחלקת חפיסה Deck
3. מחלקת מחלק הקלפים Dealer
4. מחלקת שחקן Player

מבנה הנתונים שנשתמש בו הוא מערך של קלפים באורך 52 המייצג חפיסת קלפים.

## המחלקות

### המחלקה "קלף"

```
public class card
{
    private string name;//שם הקלף
    private int val; // ערך הקלף
    private string type; //סמל הקלף

    // פעולה בונה
    public card() { }
    public card(string name, int val, string type)
    {
        this.name = name;
        this.val = val;
        this.type = type;
    }
    //get & set
    public string Name
    {
        get { return this.name; }
        set { this.name = value; }
    }
    //get & set

    public int Value
    {
        get { return this.val; }
        set { this.val = value; }
    }
    //get & set

    public string Type
    {
        get { return this.type; }
        set { this.name = value; }
    }
    // ToString

    public override string ToString()
    {
        return "name="+this.name + " " + "type="+this.type + " value=" +
this.val+"\n";
    }
} //of card
```

## המחלקה "חפיסה"

```

public class deck
{
    public card[] hand = new card[52];
    private string owner;
    // פעולות בונות הראשונה ריקה והשנייה מאתחלת את המערך לחפיסה מסודרת
    public deck() { }
    public deck(string name)
    {
        this.owner = name;
        //card[] hand = new card[52];
        if (this.owner == "dealer")
        {
            for (int i = 0; i <= 9; i++)
            {
                this.hand[i] = new card(" " + (i+1), i + 1, "clubs");
            }
            this.hand[10] = new card("J", 10, "clubs");
            this.hand[11] = new card("K", 10, "clubs");
            this.hand[12] = new card("Q", 10, "clubs");
            for (int i = 13; i <=22; i++)
            {
                hand[i] = new card(" " + (i - 12), i - 12, "diamonds");
            }
            this.hand[23] = new card("J", 10, "diamonds");
            this.hand[24] = new card("K", 10, "diamonds");
            this.hand[25] = new card("Q", 10, "diamonds");
            for (int i = 26; i < 36; i++)
            {
                this.hand[i] = new card(" " + (i - 25), i - 25, "hearts");
            }
            this.hand[36] = new card("J", 10, "hearts");
            this.hand[37] = new card("K", 10, "hearts");
            this.hand[38] = new card("Q", 10, "hearts");
            for (int i = 39; i < 49; i++)
            {
                this.hand[i] = new card(" " + (i - 38), i - 38, "spades");
            }
            hand[49] = new card("J", 10, "spades");
            hand[50] = new card("K", 10, "spades");
            hand[51] = new card("Q", 10, "spades");
        }
        else
        {
            for (int i = 0; i <= 51; i++) this.hand[i] = new card("E", 0, "E");
        }
    }
}

public override string ToString()

```

```

{
    string st = "";
    for (int i = 0; i < 52; i++)
        st = st + this.hand[i].ToString();
    return this.owner+"\n"+st;
}
} //end of deck

```

## המחלקה "דילר"

```

class dealer
{
    private string name;
    private deck gd=new deck("dealer");
    // פעולות בונות
    public dealer() { }
    public dealer(string name,deck gd)
    {
        //card[] de = new card[52];
        this.name = name;
        this.gd = new deck(name);
    }
    public int numOfCards()
    { //counts the num of cards in the deck
        int i, count = 0;
        for (i = 0; i <= 51; i++)
            { if (this.gd.hand[i].Name != "E") count++; }
        return count;
    }
    public card giveCard(string nameOfPlayer)
    { // the dealer gives the nameOfPlayer a card
        int place = this.numOfCards();
        card c = this.gd.hand[place-1];
        this.gd.hand[place-1] = new card("E", 0, "E");
        return c;
    }
    // מערבבת את ההפיסה
    public void shuffle()
    {
        deck d = new deck("dealer");
        int i, y, x; card tempc;
        Random rnd = new Random();
        for (i = 0; i <= 1000; i++)
        {
            x = rnd.Next(52); y = rnd.Next(52);
            tempc = gd.hand[x]; gd.hand[x] = gd.hand[y]; gd.hand[y] = tempc;
        }
    }
    public override string ToString()

```

```

    {
        return this.gd.ToString();
    }

} // end of dealer

```

## המחלקה "שחקן"

```

class player
{
    private string name;
    public deck deckinhand = new deck(""); //the deck in the hand of the player
    public player(string name)
    { //constructor of a player that takes playername and sets the deck to be empty
        this.deckinhand = new deck(name);
        this.name = name;

        for (int i = 0; i <= 51; i++)
        {
            this.deckinhand.hand[i] = new card("E", 0, "E");
        }
    }
    public int numOfCards()
    { //returns the num of cards in the hand of the player
        int i, count = 0;
        for (i = 0; i <= 51; i++)
        { if (this.deckinhand.hand[i].Name != "E") count++; }
        return count;
    }
    public card getCard(dealer deal)
    { //the player gets a card from the dealer
        int place = this.numOfCards();
        card c = deal.giveCard(this.name);
        this.deckinhand.hand[place] = c;
        return c;
    }
    public override string ToString()
    {
        return this.deckinhand.ToString();
    }
}

} // end of player

```

## התוכנית הראשית

```

class Program
{
    static char yesOrNo(char ans)
        // הפונקציה מקבלת תו
        // הפונקציה היא מסננת קלט המחזירה קלט תקין
    {

```

```

char c=ans;
while (!(c.Equals('y')) && !(c.Equals('n')))
{
    Console.WriteLine("type 'y' or 'n'");
    c = char.Parse(Console.ReadLine());
}
return c;
} // end of yesOrNo
static void Main(string[] args)
{
    int pl = 0; int com = 0; card c,cc;
    Console.WriteLine("IF U WANT TO START A GAME TYPE 's' IF NOT TYPE 'end'");
    string newgame = Console.ReadLine();
    //loop for a new game
    while (newgame == "s")
    {
        player p = new player("player");
        player comp = new player("computer");
        deck dealerdeck = new deck("dealer");
        dealer deal = new dealer("dealer", dealerdeck);
        deal.shuffle();// מערבבים את החפיסה
        c=p.getCard(deal);// שחקן מקבל קלף
        Console.WriteLine("A card for the player      {0}", c.ToString());
        cc=comp.getCard(deal);
        Console.WriteLine("A card for the computer    {0}", cc.ToString());
        // מחשב מקבל קלף
        int numofcardsp = p.numOfCards();
        int numofcardsc = comp.numOfCards();
        int sump = c.Value;
        int sumc = cc.Value;
        Console.WriteLine("Your total score= {0}", sump);
        c=p.getCard(deal);
        // שחקן מקבל עוד קלף
        Console.WriteLine("Another card for the player    {0}", c.ToString());
        cc=comp.getCard(deal);// מחשב מקבל עוד קלף
        Console.WriteLine("Another card for the computer    {0}", cc.ToString());
        numofcardsp = p.numOfCards();
        Console.WriteLine("Your total score= {0}", sump);
        Console.WriteLine("The computer total score= {0}", sumc);
        numofcardsc = comp.numOfCards();
        sump = sump +c.Value; ;
        sumc =sumc+ cc.Value;
        char ch;
        bool flag = true;
        // הצגת סכום כולל של שחקן

        while ((sumc < 21) && (sump < 21)&&flag)
        {
            Console.WriteLine("The computer total score = {0}", sumc);
            Console.WriteLine("HI player , your total score = {0}", sump);

```

```

Console.WriteLine("Do you want a card ?type 'y' or 'n'");
ch = char.Parse(Console.ReadLine());
// מסננת קלט
ch = yesOrNo(ch);
if (ch == 'y')
{
    c = p.getCard(deal);
    Console.WriteLine("You have got the card      {0} ", c.ToString());
    numofcardsp = p.numOfCards();
    int cardvaluep = c.Value;
    if ((21 - sump >= 11) && (cardvaluep == 1)) cardvaluep = 11;
    sump = sump + cardvaluep;
}
if ((21 - sumc > 21 - sump) || (ch=='n'))
{
    cc=comp.getCard(deal);
    Console.WriteLine("The computer have got the card      {0} ",
        cc.ToString());
    Console.WriteLine("The computer score = {0}", sumc);
    Console.WriteLine("Press 'Enter' to continue");
    Console.ReadLine();
    numofcardsc = comp.numOfCards();
    int cardvaluec = cc.Value;
    if ((21 - sumc >= 11) && (cardvaluec == 1)) cardvaluec = 11;
    sumc = sumc + cardvaluec;
}
if (pl == sump && com == sumc) flag = false;
}
Console.WriteLine("Result for player = {0}", sump);
Console.WriteLine("Result for compter = {0}", sumc);
if ((sump > 21) && (sumc > 21)) Console.WriteLine
    ("The winner is computer");
if ((sump <= 21) && (sumc > 21)) Console.WriteLine
    ("The winner is player");
if ((sump > 21) && (sumc <= 21)) Console.WriteLine
    ("The winner is computer");
if ((sump <= 21) && (sumc <= 21))
{
    if (sump < sumc)
        Console.WriteLine("The winner is computer");
    if (sump > sumc) Console.WriteLine("The winner is player");
    if (sump == sumc) Console.WriteLine("Tiko");
}
Console.WriteLine("IF U WANT TO START A GAME
    TYPE 's' IF NOT TYPE 'end'");
newgame = Console.ReadLine(); pl = 0; com = 0;
}
Console.ReadLine();
}

```