עבודת גמר בתכנון ותכנות מערכות

5 יח"ל – שאלון _____

ChessMaster

שם התלמיד : ניצן ויינגרט
ת.ז. : 209857382
שם הבית ספר והעיר: תיכון מו"ר (מודיעין, מכבים ורעות)
מורה: אורן גרוס
תאריך: 3.5.2022

תוכן עניינים

מסמך ייזום.....2
מבוא.....2
תיאור מוצר 2
אילוצים ודרישות 2
תיחום הפרויקט 3
מצב השוק 3
מסמך אפיון.....4
פונקציונליות המערכת.....4
אילוצים עיקריים.....4
תרשים ארכיטקטורת המערכת.....5
מסמך עיצוב.....6
סביבת פיתוח.....6
הנושאים שנדרשתי ללימוד עצמי וחקר.....6
תהליך החקר.....6
תרשים פונקציונליות של הפרויקט והקשרים ביניהם.....7
תרשים מודלים של הפרויקט והקשרים ביניהם.....8
פירוט המודלים של הפרויקט.....8
עיצוב נתונים ופרוטוקולים.....9
ממשק משתמש והוראות הפעלה.....10
קודים עיקריים של מודלים והפונקציות העיקריות.....10
ביבליוגרפיה.....12

מסמך ייזום

מבוא

תיאור של התחום שבו עוסקת העבודה

בתחום תורת המשחקים נשאלת השאלה האם יש אסטרטגיה שתמיד תביא לניצחון במשחק ללא מזל שבו כל הכלים חשופים לכלל המשתתפים. נכון להיום לא נמצא משחק כזה שאין לו "פתרון" או שהשחקן הראשון יכול תמיד לנצח או שהשני או שתמיד יהיה תיקו (לדוגמה באיקס עיגול כאשר 2 השחקנים הכי טובים תמיד התוצאה תהיה תיקו). התעלומה הגדולה היא במשחק שחמט, שבו אין מזל וכל הכלים חשופים אך עדיין לא נמצאה אסטרטגיה מנצחת עבורו. בשביל לגלות את ה"פתרון" לשחמט החלו ליצור אלגוריתמים ובינה מלאכותית שבעזרתה יוכלו להבין מהו ה"פתרון" של שחמט.

סוגי משתמשים:

שחקנים מתחילים שינסו ללמוד אסטרטגיות מהמהלכים של האלגוריתם.
שחקנים מנוסים שינסו לנצח את ה-AI.

תיאור המוצר

השרת נפתח והוא מחכה למשתמשים חדשים.
לקוח או מספר לקוחות מנסים להתחבר לשרת והוא מנהל נגד כל אחד משחק שחמט, המשחק הוא בין הלקוח לבין אלגוריתם הנמצא בשרת, לאחר כל מהלך של הלקוח, האפליקציה שולחת לשרת את המהלך שנבחר, האלגוריתם בוחר איזה מהלך הוא יבצע ושולח לו את המהלך שנבחר.

אילוצים ודרישות

בעיות שהמערכת צריכה להתמודד איתם:

- השרת צריך להתמודד עם כמה לקוחות במקביל. לשם כך השתמשתי ב-Threading על מנת להריץ כמה משחקים במקביל ולקבל הודעות מלקוחות שונים במקביל.
- חוסר תיאום בין החוקים בשרת ללקוח-הקודים לכיצד המשחק מנוהל היה שונה ולשם האחידות ולעבודה יעילה יותר יצרתי ספרייה (Core) שנמצאת בשניהם שבה יש את החוקים וחיברתי ששניהם יעבדו איתה.

תיחום הפרויקט

הפרויקט מורכב משלושה חלקים ראשיים:

1. core: מכיוון שחוקי השחמט זהים גם בלקוח וגם בשרת יצרתי ספרייה הנקראת core שבה יש את החוקים הבסיסיים, הספרייה נמצאת גם בשרת וגם בלקוח.
2. שרת- מנהל את המידע מהלקוחות ומכיל את האלגוריתם.
3. לקוח- מכיל את ממשק המשתמש ומאפשר לשחק לבצע מהלכים במשחק השחמט.

מצב השוק

מוצרים דומים בשוק:

ברחבי האינטרנט ישנם אתרים ותוכנות רבות של משחקי שחמט ושל אלגוריתמים עבורם אך עניין אותי מאוד כיצד הם עובדים ולכן בחרתי בנושא זהה.
מוצר יהיה דומה ל-Chess.com.

מסמך אפיון

פונקציונאליות המערכת

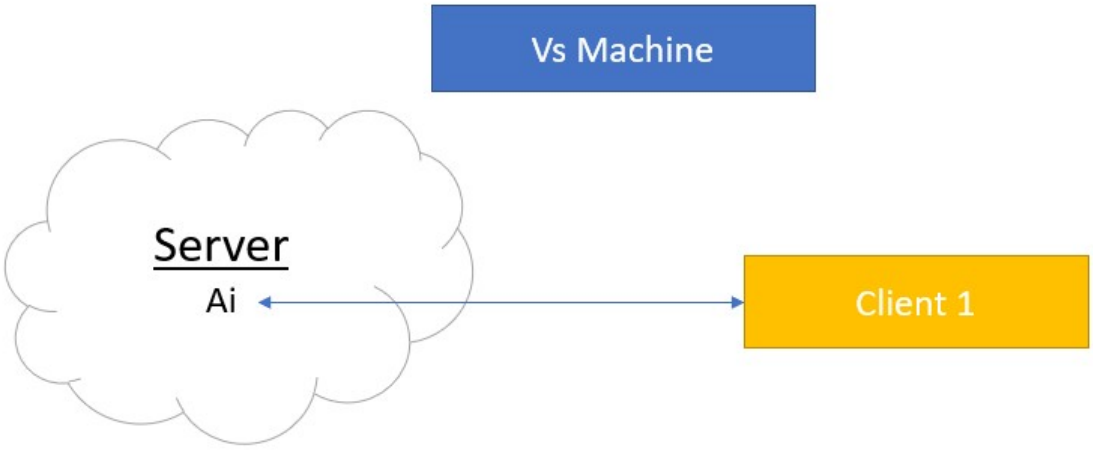
- ממשק לקוח
- אפשרות ללחוץ על חייל, לראות את המהלכים האפשריים שלו ולבצע מהלך
- התחברות למערכת - יצירת קשר בין השרת לכל אחד מהלקוחות.

כתיבה במנוע למשחקים Unity ובו אכתוב scripts ב-#C.
אשתמש בשיטת MinMax ל-AI על מנת למצוא את המהלך האופטימלי.

אילוצים עיקריים

- התוכנה רצה על מערכת הפעלה windows 10.
- המערכת דורשת חיבור לאינטרנט
- האלגוריתם יהיה פשוט יותר ולכן פחות אפקטיבי - (ככל שהאלגוריתם רואה יותר קדימה כך מספר האופציה עולה בצורה אקספודנצילית אז אני מוגבל בגלל הזמן לעומק של 4-5 מהלכים קדימה (חישוב האלגוריתם יהיה על פי 4-5 המהלכים האפשריים הבאים).

תרשים ארכיטקטורת המערכת



מסמך עיצוב

סביבת פיתוח

הפרוייקט נעשה בשפת #C בסביבת Visual Studio 2022 ובמנוע Unity.

הכלים והספריות בהם השתמשתי בהם:

- Socket – ספרייה לשימוש בתקשורת בין מחשבים שונים, השתמשתי בו על מנת לבסס תקשורת TCP.
- Time – השתמשתי בספרייה Time על מנת לבדוק את המהירות של האלגוריתם ומתי לנתק לקוח שנמצא זמן רב מדי מחובר.
- Threading – ספרייה המאפשרת שימוש בThreads.
- UnityEngine – ספרייה של המנוע הגרפי שמכיל עבודה עם אובייקטים, סצנות וscripts.

הגדרות ומושגים

- Unity – מנוע גרפי המאפשר ליצור משחקים ואפליקציות בשפת #C עבור מגוון פלטפורמות.
- Scripts – במנוע הגרפי נוצרים אובייקטים שאליהם משייכים קובצי קוד הנקראים scripts.
- Scenes – המנוע מאפשר מעבר בין סצנות לדוגמא בין תפריט ראשי לסצנה של המשחק עצמו.
- MinMax Algorithm – האלגוריתם המשמש לבחירת המהלך האופטימלי הבא.

הנושאים שנדרשתי ללימוד עצמי וחקר

- שימוש במנוע המשחק Unity – ניהול האובייקטים שעל המסך ושימוש בסצנות בשביל מסכים שונים (התחברות, משחק, ניצחון)
- שימוש ב-sockets ב-#C בשביל תקשורת בין הלקוח והשרת.
- שימוש ב-Threding כדי שהשרת ינהל מספר משתמשים במקביל.
- MinMax Algorithm – הדרך שבה המחשב יודע איזה מהלך לבצע.

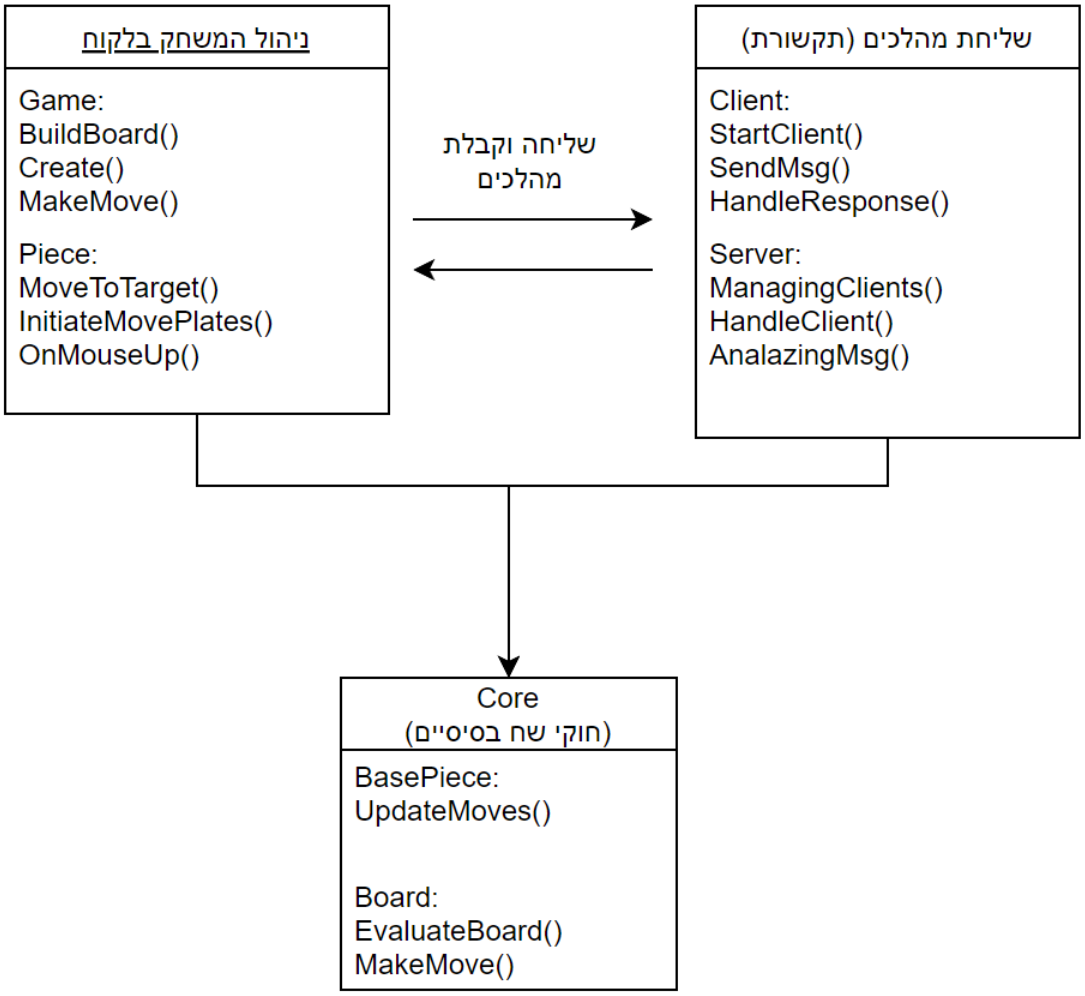
תהליך החקר

שימוש במנוע המשחק Unity:

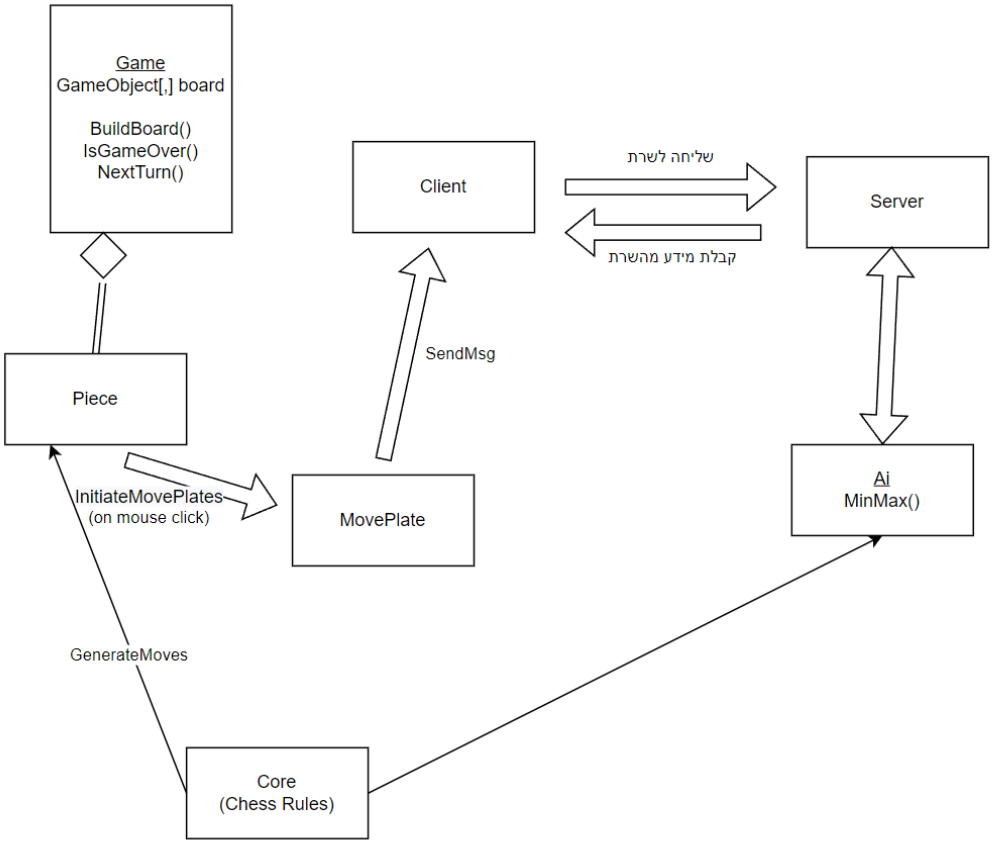
בתחילה ראיתי איך עשו פרויקטים אחרים בUnity ולמדתי את הדברים הבסיסיים, איך ליצור אובייקטים ואיך לסגור אותם, התחלתי לכתוב את צד הלקוח ובמהלך הכתיבה בדקתי באינטרנט כיצד לעבוד עם קלט מהעכבר ועם החלפת סצנות.

שימוש ב-sockets ב-#C ליצור חיבור בין השרת: לאחר חיפוש ממושך באינטרנט יצרתי שרת ולקוח בקובץ אחר על מנת לבחון את אמינות של הספרייה ולאחר מכן שילבתי בקובץ גם multiThreading

תרשים פונקציונליות של הפרויקט והקשרים ביניהם



תרשים מודלים של הפרויקט והקשרים ביניהם



פירוט המודלים של הפרויקט

Core

BasePiece – מחלקה לכל כלי משחק, מחלקה זו פשוטה יותר מPiece מכיוון שהיא צריכה לעבוד מהר כשהAi מייצר מיליוני אפשרויות על מנת לבחור את האופציה המיטבית. המחלקה מכיל את הלוגיקה לאפשרויות למהלך הבא של אותו שחקן.

Board - לוח מצומצם יותר על מנת שיעבוד מהר יותר.
Point - מחלקה פשוטה של x ו-y.
Move - מחלקה שמכילה נקודת התחלה ונקודת סיום.

Server

ManagingClients()

הפעולה הראשית שפותחת שרת ומחכה למשתמשים להתחבר, לאחר החיבור היא משייכת לכל אחד Thread וsocket ומפעילה את הפעולה HandleClient לכל Thread

HandleClient()
פעולה שמנהלת את ההודעות ללקוח מסויים, היא מקבלת ממנו מידע ומפעילה את AnalizingMsg ובעזרתו שולחת מידע חזרה ללקוח.

AnalaizingMsg() - פעולה המנתחת את ההודעה מהלקוח ושולחת הודעה מתאימה בחזרה.

Client - מנהל את התקשורת מול השרת, שולח ומקבל מידע ומנתח את המידע שהגיע.

Game- מחלקה ראשית שמייועדת לניהול המשחק את הלקוח, המחלקה מכילה את יצירת הלוח הוויזואלי, ביצוע של מהלכים חדשים ויצירה של כלי משחק חדשים על הלוח.

MovePlate – מחלקה שמשוייכת למהלכים האופציונאלים כאשר הלקוח בוחר כלי משחק מסויים, אם העכבר נלחץ על MovePlate המהלך יתבצע ויישלח לשרת בעזרת Client.cs.

Piece - מחלקה השייכת לכל כלי משחק בצד הלקוח, המחלקה מכילה פעולה שנוגעות לוויזואליות של כלי המשחק, כמו יצירת מהלכים אפשריים, התקדמות לכיוון היעד (כאשר נבחר מהלך).

עיצוב נתונים ופרוטוקולים

פירוט הפרוטוקולים

לאחר שהלקוח מבצע מהלך, הלקוח שולח את המהלך (Move) בפורמט הבא:
1_x1, y1_x2, y2
1 מסמן את סוג ההודעה - זוהי הודעה של מהלך שהתבצע על ידי הלקוח
x1, y1 זו נקודת ההתחלה של האובייקט ו-x2,y2 היא נקודת הסיום של האובייקט.

ממשק משתמש והוראות הפעלה

הוראות הפעלה

1. להריץ את קוד ה-#C של השרת דרך visual studio על מחשב שמחובר באותה רשת כמו הלקוחות
2. הרצת קובץ ה-exe אצל מספר מחשבי לקוח
3. חיבור הלקוח למחשב המוגדר כשרת

לאחר ההכנה ניתן לשחק בכך שלוחצים על כלי משחק, לאחר מכן יופיעו המהלכים האפשריים של אותו השחקן וכאשר העבר יילחץ על אחת מהאפשרויות השרת יחזיר לו את המהלך שלו וחוזר חלילה עד שיש מנצח!!!

קודים עיקריים של מודלים והפונקציות העיקריות

הרצת השרת:

1. הפעולה ManagingClients נקראת בתחילת הקוד והיא פותחת את השרת ומחברת משתמשים לThread שמריץ את הפעולה HandleClient.

```
1 reference
public static void ManagingClients()
{
    //main command which handles new clients and creates thread for each
    List<Thread> threads = new List<Thread>();
    IPAddress ipAddress = IPAddress.Parse("10.100.102.61");
    IPEndPoint localEndPoint = new IPEndPoint(ipAddress, SERVER_PORT);
    // Create a Socket that will use Tcp protocol
    Socket serverSocket = new Socket(ipAddress.AddressFamily, SocketType.Stream, ProtocolType.Tcp);
    serverSocket.Bind(localEndPoint);
    int id = 0;
    while (true)
    {
        serverSocket.Listen(10);
        Socket handler = serverSocket.Accept();
        RemoveFinishedThreads(threads);
        threads.Add(new Thread(() => HandleClient(handler)) { Name = "t" + id });
        threads.Last().Start();
        id++;
    }
}
```

2. הפעולה HandleClient מקבלת את הsocket ומנהלת את המשחק עם הלקוח.

```
1 reference
public static void HandleClient(Socket clientSocket)
{
    Board board = new Board();
    try
    {
        //handle sockets for each client
        string data = null;
        byte[] bytes = null;
        byte[] msg;
        while (true)
        {
            bytes = new byte[1024];
            if (data != "")
            {
                int bytesRec = clientSocket.Receive(bytes);
                data = Encoding.ASCII.GetString(bytes, 0, bytesRec);
                Console.WriteLine("{0}: {1}", Thread.CurrentThread.Name, data);

                msg = Encoding.ASCII.GetBytes(AnalizingMsg(data, board));
                clientSocket.Send(msg);
            }
            Thread.Sleep(100);
        }
    }
    catch
    {
        clientSocket.Shutdown(SocketShutdown.Both);
        clientSocket.Close();
        Console.WriteLine("close");
    }
}
```

ביבליוגרפיה

דרך להעריך את הניקוד ללוח מסויים	https://www.chessprogramming.org/Simplified_Evaluation_Function
Unity	https://unity.com/
Chess.com	/https://www.chess.com
MinMax Algorithm	https://en.wikipedia.org/wiki/Minimax