

תשע"ה 2015	הנדסת תוכנה
סמסטר ב'	ארנון ברקת

תכנות ברשת .Net עבודת גמר סמסטר (פרויקט) .

תיאור כללי

משחק ה- "איקס מיקס דריקס" (*Tic tac toe*) הוא משחק לשני שחקנים. נשחק את המשחק ברשת : יהיו שני לקוחות, כלומר שני שחקנים, שונים. הם ישחקו ביניהם ברשת בעזרת שירות *Wcf*, המאוחסן כ- *Web site*. בנוסף, *Web Site* זה, יחזיק בסיס נתונים שיאפשר לעקוב, לשמור ולבצע שאילתות שונות על המשחקים, השחקנים וכך הלאה.

את הלקוחות יש לשמור, כ"א מהם בנפרד, בתיקייה אחרת.
ה- *Web Site* שכולל את השירות ובסיס הנתונים, יהיה אף הוא בתיקייה אחרת.
כלומר, סה"כ יש שלוש תיקיות.

תיקיית ה- Web Site

1. יש לפתוח פרויקט חדש מסוג *Web Site* → *New* → *File*.
2. יש לצרף אליו בסיס נתונים כפי שעשינו בכיתה.
3. יש לצרף אליו שירות מהסוג *Wcf*.

יועצים

4. לכל אחד משני השחקנים, יכולים להיות יועצים, בכמות כלשהי.
- יועצים של שחקן מסוים מייעצים רק לו כמובן.
- יועצים אלו הם גם שחקנים מקצועיים מן הסתם.

בסיס הנתונים

5. יכלול טבלאות בהתאם, כך שנוכל לשמור נתונים עבור **שחקנים**, **יועצים**, **משחקים**, **ואליפויות**, כולל קשרי גומלין ואכיפת אילוצים בהתאם. (למשל לא יתכן מספר שחקן בטבלת משחקים שאינו מופיע בטבלת שחקנים וכך הלאה).

Wcf

6. ניתן לממש כ- *SOAP* או *REST* כרצונך.
7. יש לעבוד עם שמות קבצים ומחלקות מתאימים, כלומר בעלי משמעות.
8. לפי הצורך לחלק לממשקים שונים, כלומר לעבוד עם יותר מממשק אחד.
9. שירות ה- *Wcf* יהיה אחראי לקשר של הלקוחות השונים עם בסיס הנתונים, ועל הקשר השוטף ביניהם.

תיקיית לקוח

העבודה תבוצע עם Winform.

בלקוח לא יהיה מימוש של אף שאילתה. (כל השאילתות רק ב- Web Site).
יש לממש בכל לקוח:

תפריט

10. תפריט כמקובל, המכיל גם אפשרות הפעלה לתיבת About Box ובה הפרטים של המגיש/ים.

טופס הרשמה

11. יש טופס הרשמה עבור הפרטים של השחקן ושל יועציו. יש לעבוד עם שדות שונים מסוגים שונים, כמקובל. המשתמש מחויב למלא לפחות את שמו הפרטי כדי להמשיך, ולפחות את השם הפרטי של כל יועץ, במידה וקיימים.
12. למשתמש ו/או יועץ, יש אפשרות להוסיף מידע על אליפויות שהוא השתתף בהן, ע"י בחירת אליפות קיימת, למשל מתוך פקד *DataGridView*. במידה והוא השתתף באליפות שלא הוזנה עד היום לבסיס הנתונים, הוא יכול להוסיף אליפות חדשה. אליפות חדשה כוללת מספר סידורי, תאריכים, העיר שבה היא התקיימה, וכן אפשרות להעלות תמונה עם העיר שבה התקיימה האליפות.
13. יש לעבוד עם Validating ו- *ErrorProvider* לפי מה שעשינו בכיתה כדי לוודא תקינות נתונים.

אפשרויות משחק

14. התוכנית שומרת על חוקי המשחק, לא מאפשרת לי לצייר במשבצות תפוסות, לא נותנת לי לשחק אם זה לא התור שלי, מודיעה מי ניצח ומסיימת את המשחק כשצריך, וכך הלאה.
 15. התוכנית תומכת בשתי רמות של משחק: 4 על 4, ו- 5 על 5. יש כמובן לבצע תיאום בין שני השחקנים.
 16. התוכנית תומכת במשחק מול המחשב, בשתי הרמות, לפי בחירת המשתמש. (המחשב יגריל את צעדיו, ואין צורך לממש אלגוריתם שמנסה לנצח).
 17. כל המשחקים "מוקלטים".
- לפיכך, התוכנית תאפשר שחזור משחק שהיה בעבר, ע"י בחירת משחק מתוך כל המשחקים שהיו.

Wpf ולוח המשחק

18. לוח המשחק, יופיע כחלונית חדשה.
- כדי לשדרג את היכולות הגרפיות של המשחק, נממש את לוח המשחק והצורות (איקס או עיגול) כ- *UserControl* מסוג *Wpf*, כפי שנלמד בכיתה, שישובץ בתוך הפרויקט. יש לשלב אנימציה יצירתית של הצורות, בעת שהן מתווספות ללוח המשחק בכל מהלך, בעזרת *Storyboard* מתאים.

Wpf ושאילתות

19. כל לקוח, יוכל לבצע מספר שאילתות, על בסיס הנתונים שנמצא ב- *Web Site*.
- התוצאה של כל שאילתה, תוצג בחלונית נפרדת, אותה יש לממש כ- *UserControl* מסוג *Wpf*.
- החלונית תכיל *DataGrid* של *Wpf*, בהתאם, שיציג את התוצאות.
- ההפעלה עצמה של השאילתה / בחירת ערכים לשאילתה - תבוצע מחלונית אחרת, מהסוג הרגיל של *Winform*.

שאלות

20. הצגת כל השחקנים עם כל הפרטים.
21. הצגת כל המשחקים עם כל הפרטים.
22. הצגת כל האליפויות עם כל הפרטים כולל התמונה אם יש.
23. אני בוחר שחקן מתוך *Combo* שמציגה את כל השחקנים שיש, ואז מקבל כל המשחקים שהוא שיחק. עבור כל שחקן, יש להציג בתיבה, את שמו, ואז רווח, ואז את מספר הזהות שלו.
24. כנ"ל אליפויות שהוא השתתף.
25. אני בוחר משחק מתוך *Combo* שמציגה את כל המשחקים, ואז מקבל את כל השחקנים ששיחקו בו.
26. אני בוחר משחק מתוך *Combo* שמציגה את כל המשחקים, ואז מקבל את כל היועצים שנתנו ייעוץ באותו משחק, ולאיזה שחקן כ"א מהם ייעץ.
27. אני בוחר אליפות מתוך תיבת *Combo* שמציגה את כל האליפויות שהיו, ואז מקבל את כל השחקנים שהשתתפו באותה אליפות.
28. כל שחקן, בכמה משחקים הוא שיחק.
29. כל עיר, כמה אליפויות היא אירחה עד היום.

עדכונים ומחיקות

30. יש לאפשר עדכון פרטים עבור כל מידע שהוזן ע"י המשתמש.
31. מחיקה של שורה בודדת: כרגיל.
- מחיקה של שורות מרובות: ע"י בחירה ברשת של התא שמכיל את השם שרוצים למחוק, ואז יימחקו כל הרשומות שבהן יש את אותו שם באותה עמודה.

Threads

32. לכל לקוח, קיימת האפשרות להוסיף זמן השהייה לכל המתודות שמשיבות לשאלות, שנמצאות בבסיס הנתונים. זמן ההשהייה הוא אחיד לכל המתודות, ויהיה 3 שניות, ומטרתו לתרגל מצב שבו התקשורת עם בסיס הנתונים איטית.
33. יש לוודא ש- *Gui* של התוכנית מגיב תמיד ובכל מצב, גם אם אנו במצב 3 השניות. יש לממש בשתי דרכים שונות: בחצי מהמקומות המתאימים יש להציג פתרון עם *Task*, ובחצי האחר פתרון אחר כגון *ThreadPool*, למשל.
34. המשתמש יכול לצאת מהתוכנית בכל שלב, אפילו באמצע שאלות.
- במקרה כזה, יש לוודא שכל ה- *Threads* בתוכנית סיימו לגמרי את מה שהם היו צריכים לעשות, ואז ניתן לסיים את התוכנית. אין לבצע *Abort* על *Threads* רצים או להפריע להם.
- כדי לדעת שהן הסתיימו, יש לעבוד למשל עם *ManualResetEvent* וכך הלאה, כפי שלמדנו בכיתה.

כללי

35. פרטים רבים נותרו "פתוחים", לשיקולו האישי של הסטודנט.
- אתה יכול לבצע מטלות מהנ"ל, שאין עבורן פירוט מיוחד, בכל דרך, ובלבד שתהיה התאמה סבירה וכמקובל, למטרת התוכנית.

הגשה והגנה

36. יש לשים לב שהעבודה כולה תרוץ בהצלחה ובאופן מידי.
המועדים המדויקים להגשה ולהגנה, יתפרסמו באתר, ויהיו סביב ה- 01.09.2015.

37. הגשה: בזוגות או ביחידים. יש להגיש באתר תיקייה אחת מזופזפת, ובה שלושת התיקיות הנ"ל.
יש לעבוד לפי כל מה שעשינו בכיתה, למשל *Linq-To-Sql* וכך הלאה.

38. הגנה: כל אחד לבד.
כל אחד צריך להיות בקיא בכל העבודה כולה גם אם עשה אותה עם עוד מישהו.
יהיה מועד אחד של הגנה.

ה ה 3 f ח ה !