

תש"פ 2020	הנדסת תוכנה
סמסטר ב'	ארנון ברקת

תכנות ברשת .Net עבודת גמר סמסטר (פרויקט)

תיאור כללי

נגדיר משחק לשני שחקנים, שדומה למשחק הדמקה, באופן הבא :
לוח המשחק הוא בגודל של 8 על 4, ונראה כך :

		●		●
●			●	
	●			●
●		●		

משחקים באופן דומה למשחק הדמקה המקורי עם שינויים קלים :
אכילת יריב בהתאם לחוקים הרגילים אבל מותר לאכול רק קדימה, ורק חייל אחד בתור מסוים.
כל צד חייב לשחק כאשר הגיע תורו.
מטרת המשחק היא להגיע ראשון לשורה הראשונה בצד של היריב, מספיק עם אחד החיילים, ולא דווקא לאכול את כל שחקני היריב כמו במשחק המקורי.
בנוסף, אני מנצח גם אם הצלחתי לחסום את היריב כך שהוא לא יכול לשחק למרות שזה התור שלו.

נשחק את המשחק ברשת.
כל שחקן (לקוח) משחק כנגד ה-Server, כאשר ה-Server עונה לכל שחקן על צעד המשחק שלו בצעד נגדי.
ה-Server מרכז בנוסף גם את בסיס הנתונים ואת אתר האינטרנט של המשחק.
אין קשר בין המשחקים של הלקוחות השונים.

יש להציג בפרויקט 3 תיקיות שונות לפי הפירוט הבא :

1. תיקיית לקוח ראשון שמשחק את המשחק.
2. תיקיית לקוח שני שמשחק את המשחק. (יכול להיות זהה ללקוח הראשון)
3. תיקיית פרויקט של *ASP.NET Core Web Application (Razor Pages)*, ובו יהיה אתר האינטרנט של המשחק.
באתר זה נבצע פעולות כגון תפעול בסיס הנתונים, רישום שחקנים למשחקים, הפעלת שאילתות וקבלת תוצאות, וכך הלאה. בנוסף, דפי ה-Server של ה-Razor ישגרו לכל שחקן את צעד התשובה של ה-Server.

כלומר, סה"כ יש להגיש שלוש תיקיות.

בסיס הנתונים

4. בפרויקט בסיס נתונים מרכזי שנמצא בפרויקט ה-*Razor*.
5. לקוח יחזיק בסיס נתונים רק כדי לשחזר משחקים (ראה בהמשך). את שאר המידע הוא יקבל ע"י פניה ל-*Server*. בסיס הנתונים יכלול טבלאות בהתאם, לפי מה שצריך, כולל קשרי גומלין וכיוצ"ב.

Web API Core

6. התקשורת השוטפת בין כל לקוח לבין השרת תבוצע בעזרת *Web Api Core* כפי שנלמד בכיתה.

אתר האינטרנט

7. האתר יכיל:
 - דף הרשמה:
 - האתר יציג טופס הרשמה עבור שחקנים שנרשמים למשחק.
 - יש לעבוד עם מספר שדות שונים מסוגים שונים, כמקובל, כולל שדות חובה.
 - יש לבצע *Validation* כדי לוודא שיש ערכים בשדות החובה.
 - משחק חדש מאופשר מיד רק עם סיום מלא של ההרשמה.
 - דף/דפים נוספים בהתאם, בהם נציג תוצאות של שאילתות מבוקשות, כפי שיפורט בהמשך.
 - דף 'אודות' ובו פרטי המגשים.
 - יש לאפשר *link*-ים מתאימים שיאפשרו מעבר בין הדפים השונים.

לקוח

8. עבור כל לקוח נציג את לוח המשחק, ובנוסף חלוניות אחרות בהתאם לצורך.

לוח המשחק

9. אילוץ: יש לשלב אנימציה יצירתית של הצורות ו/או של המשבצות אליהן מגיע חייל.
10. ניתן לממש את הלוח / לקוח, באחת מהדרכים הבאות, כרצונך:
 - ב-*Winform*: ואז יש לעבוד עם הפקודות הגרפיות שלמדנו בכיתה, המחלקה *Graphics*, עבודה עם *Bitmap* בהתאם לצורך, *Timer* וכך הלאה.
 - ב-*Wpf*: יש לעבוד עם *Animation* וכי'.

אפשרויות משחק

11. התוכנית שומרת על חוקי המשחק, לא מאפשרת לי להגיע למשבצת תפוסה, לא נותנת לי לשחק אם זה לא התור שלי, מודיעה מי ניצח ומסיימת את המשחק כשצריך, וכך הלאה.
 12. התוכנית לא מאפשרת משחק אלא אם שחקן נרשם לפני כן באתר.
- עבור כל שחקן, התוכנית מציגה את פרטיו בתחילת המשחק, ע"י קבלתם מ-*Server*.

13. כאמור קודם, כל שחקן משחק כנגד ה- *Server* הנמצא בפרויקט ה- *Razor*.
ה- *Server* יגדיל צעדים חוקיים, ואין צורך לממש אלגוריתם שמנסה לנצח.
14. כל המשחקים "מוקלטים". לפיכך, התוכנית תאפשר שחזור משחק כלשהו שהיה בעבר.
במשחק משוחזר, אין צורך לשמור את משך הזמן המקורי שעבר בין תור ל- תור, כלומר כל הצעדים ישוחזרו
במרווחי זמן שווים וסבירים, אולם השחזור יתמוך בשאר הדברים שהיו במשחק המקורי כגון אנימציה, צבעים,
תנועה וכו'.
לצורך השחזור יש לעבוד עם בסיס נתונים שנמצא אצל הלקוח. יש לאפשר בחירת משחק מתוך המשחקים שהיו.
(כל לקוח שומר רק את המשחקים שלו)
ניתן לבחור לעבוד על בסיס הנתונים עם אחד המודלים שלמדנו בכיתה
Entity Framework / LINQ to SQL / ADO.NET

אתר האינטרנט ושאלות

15. יש לאפשר תצוגה של תוצאות שאלות באתר האינטרנט של המשחק.
להלן השאלות שהאתר יתמוך בהן :

שאלות

16. הצגת כל השחקנים עם כל הפרטים.
17. הצגת כל המשחקים עם כל הפרטים.
18. אני בוחר שחקן מתוך *Combo* שמציגה את כל השחקנים שיש, ואז מקבל כל המשחקים שהוא שיחק.
19. כל שחקן, בכמה משחקים הוא שיחק.

עדכונים ומחיקות

- בנוסף, האתר גם יאפשר עדכונים ומחיקות.
יש לבצע תמיכה בשני מצבים לפי בקשת המשתמש באתר :
20. עדכון פרטים של שחקן.
21. מחיקה של משחק או של שחקן.

Task

22. כאמור קודם, לכל לקוח, קיימת האפשרות לשחזר משחקים.
בסעיף זה, נתרגל מצב שבו הלקוח זקוק לזמן רב כדי לעבד את המידע שהתקבל.
יש להוסיף, אצל הלקוח, אפשרות להפעלה של מצב 'תרגולי' שכזה לפי בחירה.
בעת בחירה במצב זה, יש להוסיף השהייה של 3 שניות אצל הלקוח מרגע קבלת התשובה מבסיס הנתונים.
יש לראות ש- ה- *Gui* של התוכנית נתקע עקב ההשהיה.
23. כעת, יש לפתור את הבעיה, כלומר יש לוודא ש- ה- *Gui* של התוכנית מגיב תמיד ובכל מצב, גם אם אנו במצב 3
השניות. יש לבצע ע"י עבודה עם *Task* -ים כפי שנלמד בכיתה.

כללי

24. 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.
25. יש לעבוד בפרויקט רק עם מודלים כלליים ואפשרויות כלליות שהראינו בכיתה / מצגת וכו'.

הגשה והגנה

26. יש לוודא מראש, שהעבודה כולה תרוץ בהצלחה ובאופן מידי על המחשב שבו אתה מציג את העבודה.
27. המועד המדויק להגנה, יתפרסם באתר, **נא לראות את ההודעה באתר**.
28. יש להעלות לאתר כמטלה, תיקייה אחת ובה קובץ ה-*zip*, שבו יש את 3 התיקיות של הפרויקט.
(הגשה עד מוצ"ש 29.08.2020, כולל). (נא להסיר את תתי התיקיות *.bin, .obj*)
29. **הגשה**: בזוגות או ביחידים. (מי שעושים בזוג רק אחד מבני הזוג יגיש באתר).
הגנה: כל אחד לבד.
כל אחד צריך להיות בקיא בכל העבודה כולה גם אם עשה אותה עם עוד מישהו.

ה ה 3 f ח ה !