

פרויקט סיום – סדנא ברשתות תקשורת

מבוא

בפרויקט זה אתם מתבקשים לממש גרסה *online* של המשחק "4 בשורה". מומלץ לחפש את החוקים המלאים של המשחק באינטרנט (במידה ואינכם מכירים אותם). הפרויקט נועד לשפר את כישורי התכנות שלכם ולחשוף אתכם לתכנות בארכיטקטורת *server-client*. כמו כן, באמצעות הפרויקט תכירו יותר לעומק את שכבות התעבורה ואת שכבת האפליקציה (שכבה 4 ו-5). נזכיר שבארכיטקטורת *server-client* יש שני צדדים הלקוח שמקבל שירות והשרת שמספק שירות.

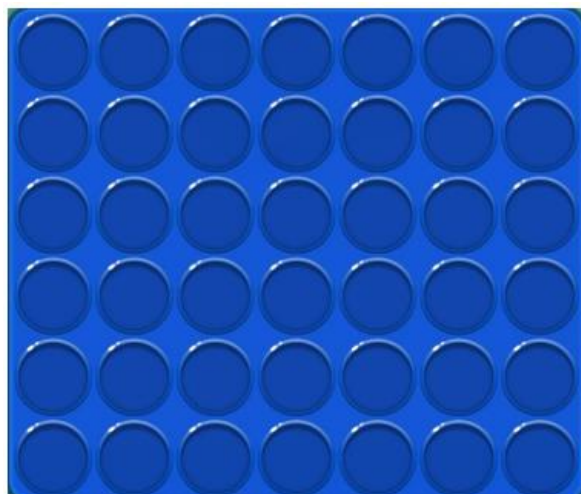
במקרה שלנו

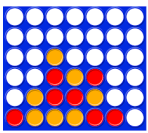
- צד הלקוח יהיה משתמש במחשב, המעוניין להשתתף במשחק.
- צד השרת יהיה המחשב שאחראי לניהול המשחק (שירותיו כוללים מתן גישה למשחק למספר לקוחות במקביל, עדכון על התקדמות המשחק, מתן אפשרות למשחק בין 2 שחקנים אחד נגד השני, השתתפות במשחק נגד לקוח וכו').



כללי המשחק

להלן הלוח ההתחלתי של המשחק:

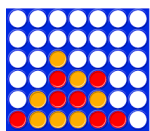




לוח המשחק מורכב מ-7 עמודות ו-6 שורות. בתחילת המשחק, כל המקומות על הלוח ריקים. המטרה של כל שחקן היא להשלים 4 תאים רצופים היוצרים קו (אלכסון/קו אופקי/קו אנכי) שאינו נחתך ע"י האסימונים של השחקן השני. בכל שלב, על השחקן אפשרות לבחור עמודה שבה הוא מעוניין להוריד את האסימון שלו. ניתן לבחור עמודה יחידה בכל תור אשר ממנה השחקן מוריד את האסימון. שימו לב כי האסימון יתפוס את המקום האחרון הפנוי בעמודה שנבחרה.
* עליכם לוודא שהמקום שנבחר תקין.

שלבי המשחק

- בתחילת המשחק, השחקן מקבל הודעת פתיחה שבה הוא מתבקש לבחור את סוג המשחק שהוא מעוניין לשחק. יש להציע לו 2 אופציות כדלהלן:
 - אינו מעוניין לשחק. במקרה זה, יש לבצע התנתקות מיידית מהשרת, ולסיים את תוכנית השחקן. על השרת להמשיך לפעול כרגיל.
 - משחק נגד השרת (AI).
 - משחק אחד על אחד עם שחקן נוסף (שאינו השרת).עליכם לחשוב טוב כיצד ניתן לממש אפשרות זו.
- עבור בחירה של אפשרות מספר 2, יש לשאול את הלקוח מהי רמת הקושי הרצויה של השרת. יש לאפשר בחירה בין 2 רמות קושי:
 - קלה - השרת מגריל בצורה רנדומלית את העמודה שהוא מעוניין להוריד את הכדור בתור זה.
 - קשה - השרת בוחר את המיקום בצורה מושכלת שאינה רנדומלית. עליכם לחשוב על דרך שבה השרת יכול לבחור את העמודה בדרך מושכלת
- לאחר מכן, יש לקבל מהשחקן את מספר הניצחונות הדרושים לנצח במשחק, כאשר המינימום הינו 1. אנא וודאו כי השחקן אינו בוחר מספר שלילי או שווה ל-0. במידת במידה והשחקן אינו שולח מספר חוקי מעל 5 פעמים, יש להשהות את פעילותו למשך דקה. לאחר סיום ההשהיה, יש להמשיך לשאול באופן דומה כאשר משך השהיית הלקוח במקרה של הכנסת מספר שגוי 5 פעמים רצוף מוכפלת ב-2 בכל פעם
- רמז:** זכרו מה המשמעות של *Blocking Call*, אילו פונקציות שירות בספריית *socket* הן מסוג *Blocking Call* וכיצד ניתן להשתמש ברעיון זה.
- לאחר ניצחון של אחד השחקנים, יש להציג את תוצאות הביניים (חלוקת הניצחונות בין השחקנים, מספר הסיבובים שלקח המשחק וכו'). הגדילו ראש

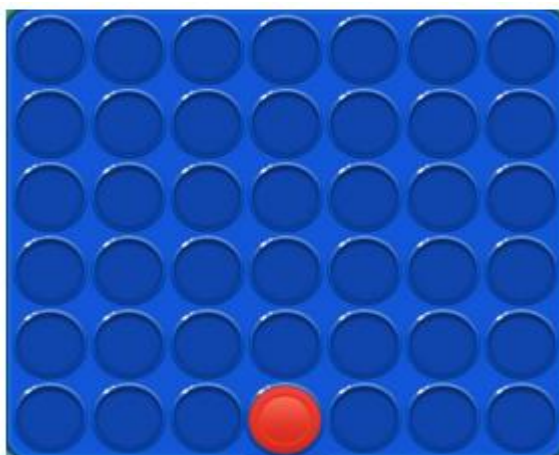


וחישבו על מדדים נוספים מעניינים שניתן להציג לשחקנים לאחר סיום משחק בודד.

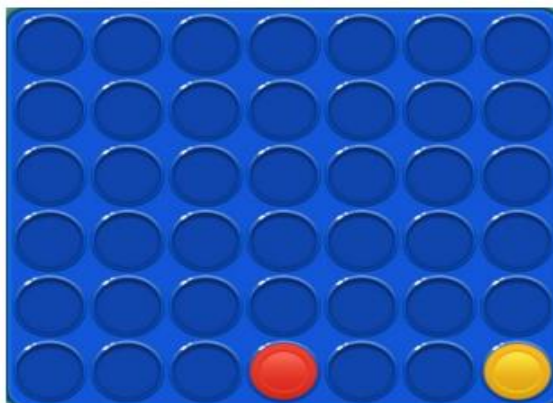
- כאשר שחקן מגיע לכמות הניצחונות הדרושה, יש להציג מדדים סופיים לגבי המשחק. יש להציג לפחות 3 מדדים.
- יש להדפיס את מצב לוח המשחק לאחר כל בחירה של שחקן. העדכון של הבחירה הנוכחית יוצג בלוח המשחק. כמו כן, יש להדפיס את הבחירה שנעשתה ע"י השחקן השני (AI) או שחקן שני אמיתי עבור האנשים שבחרנו לממש את אופציה (3).
- יש להקפיד על בדיקת מקרי קצה. הפעילו שיקול דעת במקרים בהם הבדיקה נחוצה. אין צורך לבדוק תקינות של *input*. כלומר, אין צורך לבדוק האם קיבלתם מהמשתמשים קלט מטיפוס נכון; לדוגמא קבלת *string* כאשר ציפיתם ל-*int* וכו'.

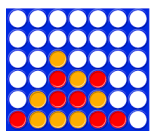
דוגמא להרצת המשחק

1. שחקן א' בוחר בעמודה מספר 4

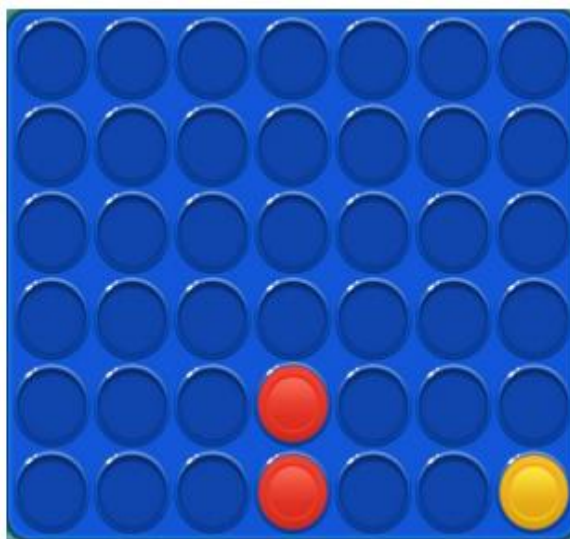


2. שחקן ב' בוחר בעמודה מספר 7

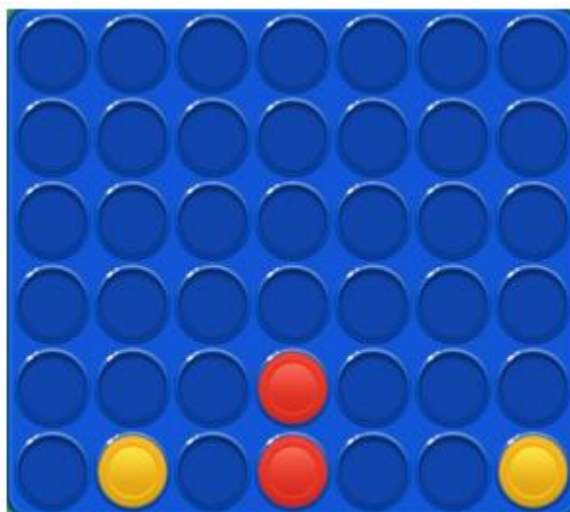




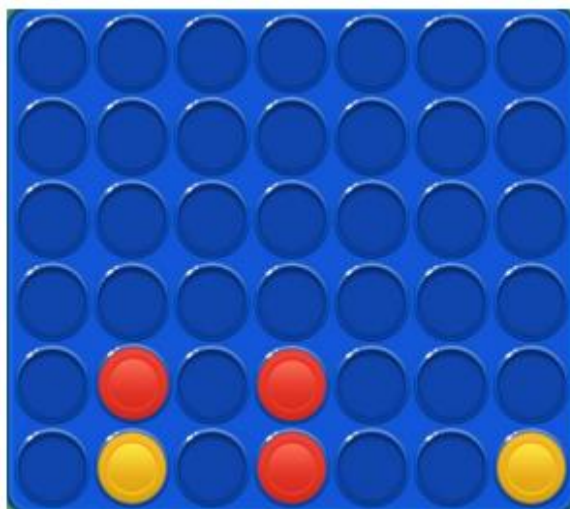
3. שחקן א' בוחר בעמודה מספר 4

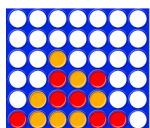


4. שחקן ב' בוחר בעמודה מספר 2

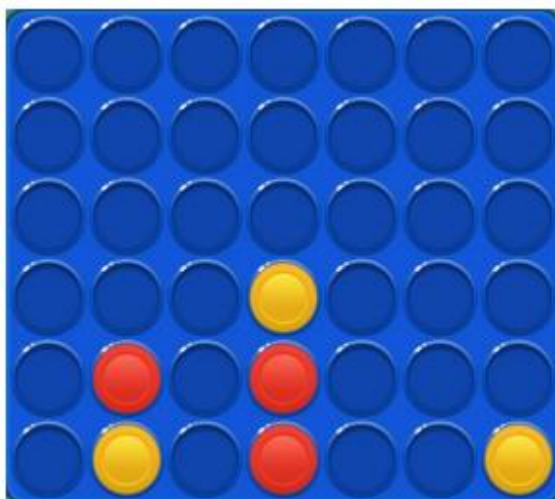


5. שחקן א' בוחר בעמודה מספר 2

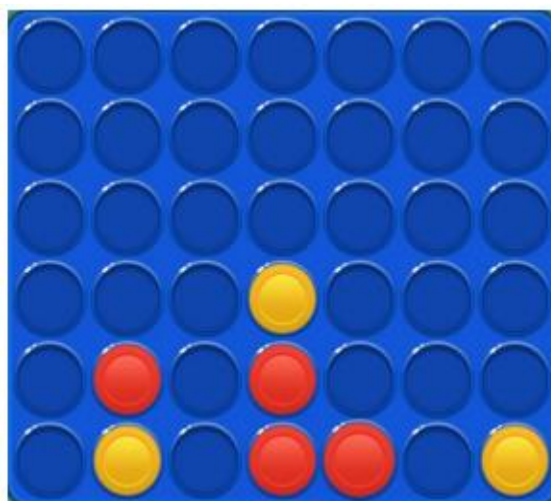




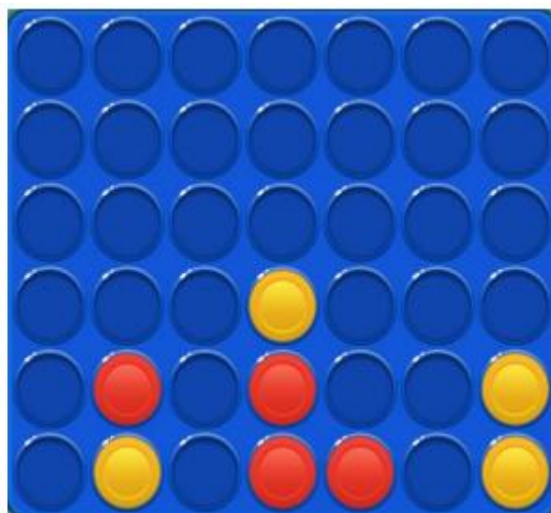
6. שחקן ב' בוחר בעמודה מספר 4

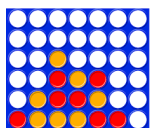


7. שחקן א' בוחר בעמודה מספר 5

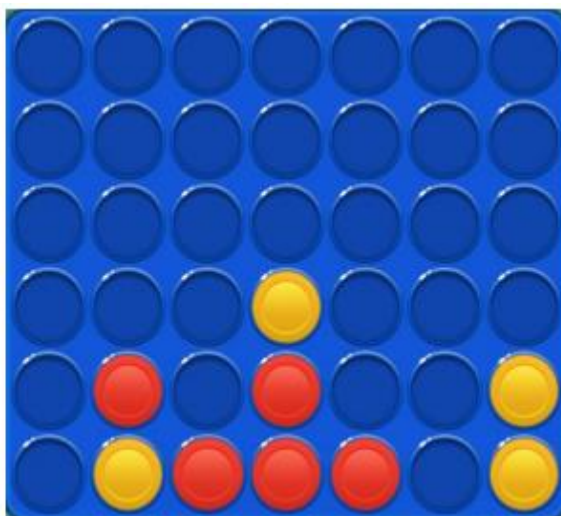


8. שחקן ב' בוחר בעמודה מספר 7

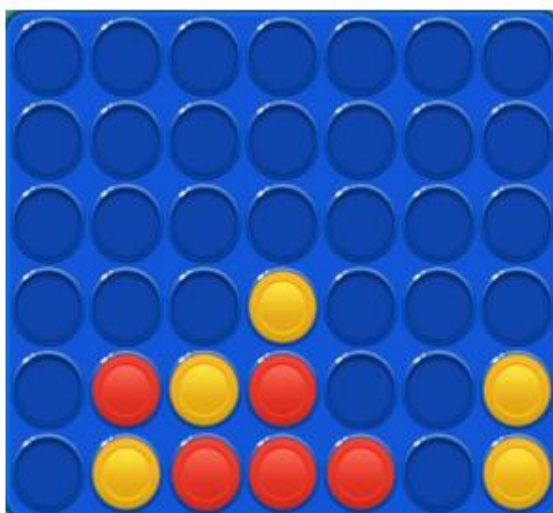




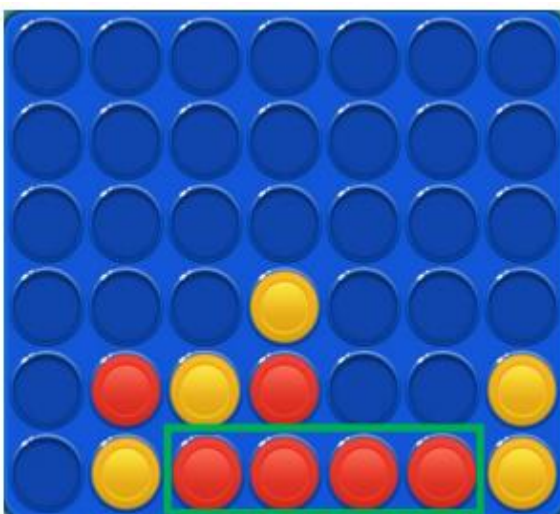
9. שחקן א' בוחר בעמודה מספר 3



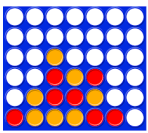
10. שחקן ב' בוחר בעמודה מספר 3



11. שחקן א' בוחר בעמודה מספר 6



כפי שניתן לראות, קיבלנו כעת 4 בשורה => השחקן **האדום** ניצח את **הצהוב**



דגשים

- אתם אחראים על יצירת התחברות אמינה בין השרת ללקוחות. השתמשו בפונקציות שירות שמומשו בספריית *Socket* בצורה נבונה
- יש לאפשר ל-5 שחקנים להתחבר לשרת בכל זמן נתון. יש לוודא כי ניתן להריץ את השחקנים במקביל. השתמשו במספר *TERMINALS* כדי לוודא זאת. עבור שחקן נוסף שמנסה להתחבר לשרת במקביל ל-5 שמטופלים ע"י השרת, שלחו לו הודעת דחייה בה מצוין שמכסת השחקנים הגיעה למלואה
- השרת הינו אוטומטי ואינו משתמש אנושי (AI). האלגוריתמיקה של השרת עבור המקרה של רמה "קשה" נתונה לשיקולכם. יש לוודא כי ברמה הזו השרת ממומש בדרך השונה מבחירה רנדומלית של המיקום על הלוח.
- איננו דורשים ממכם שתכתבו את הפלט ל-*console* בצורה ספציפית והקוד אינו יועבר בדיקה אוטומטית. כלומר אין לכם צורך להיות זהירים ולשים לב לכל אנטר, רווח וכו'. אנו מעוניינים שתשקיעו את מיטב הזמן שלכם בדברים הבאים:
 - כתיבת קוד איכותי ומתועד.
 - תיעוד נלווה ב-PDF.
 - הבנת החומר התיאורטי מאחורי הקוד.

הנחיות מימוש

- אתם צריכים לממש את הקוד מערכת הפעלה *linux* בשפת *python*.
- אנחנו ממליצים שתלמדו קצת דרך האינטרנט לפני שתתחילו לכתוב קוד, זו שפה שיותר פשוט להבין מהר במיוחד כאשר יש היכרות עם שפות תכנות אחרות.
- להלן קישור לאתר מומלץ ללימוד *python* <https://www.w3schools.com/python>.
- אסור להשתמש בספריות שאינן סטנדרטיות לתכנות מסוג *socket*.
- *programming*. לדוגמה ב-*python* עליכם להשתמש בספריית *socket*.

הנחיות הגשה

- שימו את כל הקבצים בקובץ *ZIP* עם שם המגיש ותעודות זהות
- הגישו דרך תיבת ההגשה בהתאם ללוחות הזמנים המצוינים.
- בכל קוד כתבו בהתחלה את שם המגיש + ת.ז.