

רשתות תקשורת מחשבים – תרגיל מעשי 1כתובות מייל לתקלותAmitdaniel7@gmail.com antonb@mail.tau.ac.ilתיאור התכנית

למשחק שני קבצים עיקריים: nim.py , nim-server.py
 קובץ השרת רץ בצד אחד ומקבל את הפרמטרים לגדלי הערימות (1-1000) ופרמטר אופציונלי נוסף ל – port number (ברירת מחדל 6444) .
 קובץ המשחק nim.py ירוץ בהתאם אחרי השרת כאשר עבורו ניתנים שני פרמטרים אופציונליים :
 Port number – שצריך להיות זהה לזה שניתן בשרת
 Host Name – כתובת או שם דומיין של מכונת ההרצה של השרת

פרוטוקול תקשורת

התקשורת ממומשת בפרוטוקול בינארי הרץ מעל TCP ומטופל בקובץ functions.py בו נמצאות הפונקציות לשליחת וקבלת ההודעות: pack_send , recv_unpack. פונקציות אלה נעזרות ב – my_sendall , my_recvall לשליחת וקבלת המידע, תוך כדי שמטפלות באריזת ופתיחת המידע. כל ההודעות הנשלחות הינן בפורמט של <char, Integer> בגודל 5 בתים.
 מעבר לצעדי המשחק בין השרת ללקוח סיכמנו על קודים מסויימים :

הודעה	כיוון	משמעות
<W, 0>	שרת -> לקוח	המשך משחק
<W, 1>	שרת -> לקוח	לקוח ניצח
<W, 2>	שרת -> לקוח	שרת ניצח
<M, 0>	שרת -> לקוח	צעד לא חוקי
<M, 1>	שרת -> לקוח	צעד חוקי
<Q, 0>	לקוח -> שרת	יציאה מהמשחק
<A, 1001>	לקוח -> שרת	קלט מהמשתמש בפורמט לא תקין
<A/B/C, int>	שרת -> לקוח	עדכון על גדלי הערימות
<A/B/C, int>	לקוח -> שרת	צעד המשתמש

השגיאות מהפונקציות מטופלות במקום בו קראנו להן בהתאם.

מבנה התכנית

כפי שתיארנו תחילה, שני הקבצים המרכזיים להרצת המשחק הם `nim.py` , `nim-server.py`.
קובץ `functions.py` אחראי על אריזת המידע, שליחתו, קבלת המידע, ופיענוחו.
בנוסף בתיקיית הפרויקט שני קבצי עזר נוספים :
`clientGameLogic.py` - קובץ עזר לניהול המשחק. מכיל את כל הלוגיקה של צד הלקוח במשחק.
`customException.py` - קובץ המכיל מחלקה בודדת שמטרתה לאפשר זריקת חריג מתאים משלנו.

מקרי קצה מיוחדים

התנתקות צד - במקרה של נפילת הלקוח השרת מאתחל את נתוני המשחק ומחכה להתחברות לקוח חדש. במקרה של התנתקות השרת הלקוח ידפיס את ההודעה הנדרשת לאחר הצעד הבא ויסיים את המשחק.

סגירת התכנית בעת ריצה – גם בלקוח וגם בשרת אנו תופסים את חריגים שעלולים להתקבל בסגירה יזומה כגון : `KeyboardInterrupt`, `EOFError`, `IOError`.

אנו תופסים חריגים אלו כדי לצאת מהתכנית בצורה נקיה ולסגור את ה `socket` המתאים.
טיפול בקלט – קלט בפורמט תקין `<char, integer>` יועבר ויטופל בצד השרת ולעומת זאת קלט בפורמט לא תקין (כגון "blah") יזוהה ויטופל בצד הלקוח תוך כדי עדכון השרת (שליחת `<A, 1001>`).
לא תקין (כגון "blah") יזוהה ויטופל בצד הלקוח תוך כדי עדכון השרת (שליחת `<A, 1001>`).