

רשתות תקשורת מחשבים - תרגיל מעשי #2

301341947 - תומר מירב - tmayrav@gmail.com

305750713 - נועם דביר - noamdvir123@gmail.com

בדומה לתרגיל המעשי הראשון, השתמשנו בפרוטוקול בינארי אותו מימשנו באמצעות מבנה, כאשר במהלך כל שליחה ערכי השדות המתאימים מחושבים ונקראים על ידי הצד המקבל. להלן השדות שבהם השתמשנו ומשמעותן: (לכל שדה הוקצתה כמות ביטים על פי הצורך)

unsigned int Misere (ביט 1) = 1 אמ"מ המשחק הוא misère, 0 אם לא.

unsigned int heapA (11 ביטים) – מוגדר להיות מספר הנקודות שבערימה A בכל זמן נתון.

unsigned int heapB (11 ביטים) – מוגדר להיות מספר הנקודות שבערימה B בכל זמן נתון.

unsigned int heapC (11 ביטים) – מוגדר להיות מספר הנקודות שבערימה C בכל זמן נתון.

unsigned int heapD (11 ביטים) – מוגדר להיות מספר הנקודות שבערימה D בכל זמן נתון, בנוסף, כאשר תוכנית הלקוח מקבלת קלט לא חוקי מהמשתמש היא מודיעה על כך לשרת באמצעות הפיכת ביט זה להיות 0.

unsigned int firstMessage (ביט 1) – שדה זה מוגדר להיות 1 בעת שליחה הודעה מהשרת ללקוח אם זאת היא ההודעה הראשונה מהשרת ללקוח, 0 אחרת.

unsigned int GameProgress (ביט 1) = מוגדר להיות 1 בעת שליחת הודעה מהשרת ללקוח אמ"מ במהלך החוקי אחרון שבוצע המשחק הגיע לסיומו ועל תוכנית הלקוח להדפיס למשתמש הודעת ניצחון או הפסד בהתאם. 0 אם המשחק עוד מתנהל (סכום הנקודות בערימות גדול מ-0)

unsigned int illegalMove (ביט 1) = 1 אמ"מ המהלך האחרון שביצע לקוח מסוים בתורו אינו חוקי, 0 אחרת.

unsigned int youLoseORyouWin (ביט 1) = 1 אמ"מ המשחק הגיע לסיומו ולקוח מסוים ניצח, 0 אחרת (לקוח אחר ניצח).

Unsigned int yourTurn (ביט 1) = 1 אם כעת תורו של שחקן מסוים לשחק, 0 אחרת (תורו של שחקן אחר).

unsigned int heapIndex (ביט 1) בעזרת ביט זה מודיעה תוכנית הלקוח לתוכנית השרת את הערימה ממנה בחר המשתמש להוריד נקודות בתורו.

unsigned int Player (ביט 1) = 1 אם לקוח מסוים הוא שחקן, 0 אם הוא צופה.

unsigned int yourTurn (ביט 1) = 1 אם תורו של לקוח מסוים, 0 אחרת.

unsigned int numPlayers (4 ביטים) = כמות השחקנים המסוקסימלית (P שמתקבל בעת פתיחת תוכנית השרת כארגומנט)

unsigned int yourNum (4 ביטים) = מספר השחקן של לקוח מסוים.

unsigned int joinedGame = 1 אמ"מ זהו התור הראשון במשחק בו צופה מסוים הפך לשחקן (לאו דווקא תורו, אלא התור הראשון בו הוא שחקן פעיל ולא צופה)

unsigned int tooManyPlayers (ביט 1) = 1 אמ"מ בעת ניסיון התחברות של לקוח מסוים, יש כבר 9 לקוחות מחוברים ולכן ההתחברות נדחת, 0 אחרת.

unsigned int type (2 ביטים) = בצד השרת - 0 אם הקלט שהתקבל מלקוח מסוים היה הודעה, 1 אם זה היה מהלך, 2 אם הלקוח ביצע QUIT. בצד הלקוח - 0 = הודעה, 1 = עדכון (כל עדכון מהשרת שהוא לא הודעה מלקוח אחר)

unsigned int moveStatus = 1 אמ"מ ההודעה שנקלטה אצל לקוח מסוים הייתה הודעת עדכון (type 1) הראשונה שהתקבלה לאחר שהלקוח ביצע את תורו.

Char MSG [240] – תוכן ההודעה שנשלחת בין משתמשים – מוגבל ל-240 תווים.

Int sourceORdest – בעת שליחת הודעה מתאר את הנמען, בעת קבלת הודעה מתאר את המוען.

Int moveRejected = 1 אמ"מ שחקן מסוים שלח הודעת משחק (מהלך, type = 1) שלא בתורו, 0 אחרת.

מימשנו את הבונוס

כעת נפרט מספר מצבים שאופן טיפולם לא הוגדר בצורה דטרמיניסטית בהוראות התרגיל וכיצד בחרנו לטפל בהם:

-אם במהלך התחברות השרת קרתה תקלה באחד משלבי ההתחברות/הגדרת סוקט, תבוצע התנתקות והתוכנית תסיים את ריצתה.

-אם לקוח מסוים ביצע ניסיון התחברות אך יש 9 לקוחות שכבר מחוברים, בקשת ההתחברות תידחה.

-אם במהלך שליחה לקבלה של הודעות מהשרת ללקוח מסוים (שכבר מחובר) קרתה תקלה, אותו לקוח ינותק והסוקט שלו ייסגר.

-כאשר נשלחת הודעה (message) מלקוח א' ללקוח ב', השרת בודק האם הלקוח מוכן לקבלת הודעה (select) אם לא הוא מוותר על שליחת ההודעה וממשיך בתוכנית.

-כאשר לקוח מבצע תור (בתורו) נשלחת לכל הלקוחות הודעה אם המידע הרלוונטי לכל לקוח. במשך הזמן הזה מבוצע blocking עד שכל המשתמשים מקבלים את ההודעות.

-כאשר לקוח מבצע תור שלא בתורו, נשלחת אליו הודעה בהתאם עם blocking עד שהוא פנוי לקבל אותה.

כעת נתאר בקצרה כיצד עובד קוד השרת וקוד הלקוח:

קוד הלקוח:

בדומה לתרגיל הקודם מבוצעת התחברות לשרת ולאחר מכן נכנסים ללולאת האירועים. ללולאת האירועים מחכים באמצעות קריאה לפונקציה select להודעה מהמשתמש (stdin) או להודעה מהשרת. אם יש קלט שמוכן לקריאה בודקים אותו ושולחים הודעה מתאימה לשרת.

אם יש הודעה מהשרת שמחכה לקריאה, קוראים אותה ואת מצב המשחק ובמידת הצורך ובהתאם מדפיסים למשתמש את מה שצריך. אם התור הנוכחי הוא תורו של הלקוח, אז נכנסים ללולאה שמבקשת מהמשתמש שיכניס את התור שבחר לבצע, במידה ובחר לשלוח הודעה לשחקן אחר, שולחים אותה לשרת ומחכים לאותו משתמש שיבצע את תורו, כך עד שהוא מבצע את תורו. לאחר שביצע את תורו (בין אם המהלך היה חוקי או לא) חוזרים לתחילת הלולאה.

קוד השרת:

-ללולאת האירועים של קוד השרת מחולקת לשני חלקים: הראשון הוא בדיקה אם קיימים לקוחות חדשים המבקשים להצטרף וטיפול בבקשותיהן בהתאם למצב המשחק. החלק השני הוא קבלת הודעות מהלקוחות ותגובה בהתאם.

שלושת המצבים ששרת יכול להגיב אליהם הם העברת הודעות בין לקוחות, טיפול במהלך של לקוח ועדכון בהתאם (או ללקוח אחד או לכולם) או ניתוק לקוח (שעשה QUIT או שהתחברות איתו התנתקה מאיזשהי סיבה)

-במידה ויש תקלה בקבלת הודעה מלקוח מסוים, אותו לקוח מנותק (כפי שמתואר בסעיף הרביעי) ולאחר מכן השרת בודק אם ישנם לקוחות שהם צופים וניתן להפוך אותם לשחקנים פעילים. במידה ויש, נשלחת אליהם הודעת התחברות כמתואר בסעיף השישי.