**פרויקט גמר כיתה י"א סייבר**

**שם הנושא:**

**Games Of Magic**

**שם מלא של חברי הצוות:**

**דניאל ואזנה**

**מורה מלווה : עדינה אנדן**

****

**מאי 2019**

תוכן עניינים

תקציר הפרויקט :

בפרויקט יש שני סוגים של משחק, משחק רגיל המתנהל בין שני שחקנים ומשחק טורניר אשר יש שני קבוצות של שניים בכל קבוצה ולאחר הסבב הראשון של הטורניר עולה המנצח לסבב השני ומתנהל קרב נוסף וזה שניצח את הקרב ניצח את הטורניר.

מטרת הפרויקט  ותאור מלא :

במהלך הקרב, השחקן ישתמש בדמויות אשר קיבל מראש.

זירת הקרב מחולקת לשני צדדים: הצד של השחקן והצד של היריב ולכל צד/שחקן יש טירה. מעל לזירת הקרב נמצא סרגל כלים אשר מכיל את כמות הדמויות שהשחקן יכול לשים כרגע( 2 מתוך 6 דמויות שהוא משתמש במהלך המשחק), ואתה כמות ה-mana(שיקוי אשר בעזרתו אפשר ליצור דמויות במשחק) ברגע זה.

לאורך כל המשחק השחקן מקבל mana, אלא אם כן לשחקן יש 10 mana ואז הוא לא יכול לקבל יותר. כל דמות עולה כמות מסוימת של mana, וברגע שהשחקן שם את הדמות, הוא מאבד כמות של mana על פי מחיר הדמות.

המטרה של כל שחקן היא להרוס את הטירה של השחקן השני, אבל תוך כדי להשאיר את הטירה שלו עומדת. מנצח השחקן שהטירה שלו לא נהרסה והרס את הטירה של השני.

שפת התכנות וסביבות העבודה :

שפת תכנות: bash ,python.

סביבת העבודה: PyCharm, Code Runner.

ניסוח וניתוח הבעיה האלגוריתמית :

כאשר מימשתי את הטורנירים, הבעיה האלגוריתמית הייתה לאחר שחיכינו ל4 משתמשים אשר רוצים להשתתף בטורניר ולאחר מכן להתאים לכל שחקן, שחקן אחר ולבדוק אם יש פרישה אז להתאים את המנצח של הקרב לסבב הגמר, ולאחר מכן שיש שחקן שני לגמר להוציא את שני המפסידים מן הטורניר ולהתאים למנצחים של הסבב הראשון למשחק הגמר.

בנוסף הוספתי שהמשחקים של שני השחקנים יתחילו בדיוק באותו זמן כאשר הם מחכים בדיוק לאותו זמן(חמש שניות לאחר שהסרבר התאים להם משחק) ובכך אנחנו בטוחים שהתחילו באותה שנייה.

פיתוח הפתרון בשכלול הקוד עם שפת התכנות :

שלבי הכנת הפרויקט:

1.    עיצוב מסך פתיחה והגדרת הוראות משחק ומוזיקה למסכים וכולי.

2.    הכנת גרפיקה זירה.

3.    איסוף גרפיקה דמויות.

4.    להתחיל לכתוב את הקוד(בהתחלה ע״י מסך פתיחה - דומה לאסמבלר).

5.    יצירת מחלקת דמות.

6.    יצירת כל אחת מהדמויות בעזרת ירושה ממחלקת דמות והגדרות נוספות.

7.    לרוץ על רשימת הדמויות ולעשות רשימה חדשה של אחרי תקיפות ותזוזות.

8.    לבדוק אם מקבלים לחיצת עכבר ולבדוק בלקוח אם אפשר להוסיף את הדמות ואם כן לשלוח בקשה לשרת.

9.    לבדוק האם אנחנו מקבלים מידע מהסרבר ולעדכן את התכנית שלנו בהתאם.

10. לשלוח בקשת משחק לשרת, והשרת ידאג להגיד לנו איזה צד אנחנו.

11. ליצור משתמשים(מסך יצירת משתמש, פרוטוקול, מילון בשרת, תיבת הכנסת טקסט ובדיקת סיסמה).

12. מסך ניקוד אישי(לעצב מסך, לכל משתמש יהיה ניקוד וניצחונות).

13. ליצור טבלת מובילים(לעצב מסך, לכל משתמש יהיה ניקוד, להכין מסך מובילים, להחזיר רשימה שתהיה ממוינת לפי ניקוד).

14. ליצור טורנירים(4 שחקנים, 2 סבבים, עבודה בשרת ושהתכנית של לקוח תהיה מותאמת גם לטורניר).

תיאור המודולים של מערכת התוכנה :

**אובייקטים**:

Character – אובייקט בסיסי אשר שאר הדמויות יורשות ממנו, והוא מפעיל פעולות בסיסיות כמו תזוזה, הצגת תמונת תקיפה, הצגת תמונת הליכה, לתקוף אובייקט.

Medusa, Wizard, Minotaur, Skeleton, Golem, Knight – דמות היורשת מאובייקט ה Character וכאשר יוצרים את האובייקט הזה צריך רק לתת את הנקודה שהיא צריכה להתחיל ואת הצד שלה (ביצירת הדמות ניתנת כתובת תמונות קבועה, חיים ועוד).

Castle – אובייקט טירה, אשר מכיל חיים, תמונות לפי מצב חיים, פונקציית הצגת תמונה וחיים.

Tournament – אובייקט המייצג טורניר, כאשר הוא נוצר, מוגדרות שני קבוצות של שני המשחקים של הסבב הראשון, ולאחר ניצחון הוא פרישה המנצח נוסף לגמר ונוספות הנקודות שנשארו לטירתו לנקודות של הטורניר, כאשר יש שני שחקנים לגמר, הטורניר מארגן מחדש את המשחקים, מוחק את המפסידים מהטורניר ומתחיל משחק חדש.

User – אובייקט המייצג משתמש (מכיל שם משתמש, סיסמה, מספר ניצחונות ונקודות).

**רשתות**:

כאשר המשתמש רוצה להרשם/להכנס למשתמש הוא שולח את השם משתמש והסיסמה לשרת ולאחר מכן התכנית מחכה בrecv מפני שאנחנו לא רוצים שהשמתמש ימשיך עד שנקבל תשובה מהשרת.

לאחר שקיבלנו אישור מהשרת לגבי המשתמש נפתח מסכי הפתיחה.

מסך טבלת המובילים: כאשר יש בקשה לטבלת המובילים הלקוח שולח בקשה לסרבר והלקוח מחכה לתשובה מהסרבר בrecv, והסרבר שולח ללקוח רשימה עם שלושת המובילים לפני הניקוד.

מסך ניקוד אישי: : כאשר יש בקשה לניקוד אישי הלקוח שולח בקשה עם שם משתמש וסיסמה לסרבר, והסרבר בודק אם יש שם משתמש כזה ואם כן אז בודק שהסיסמה מתאימה, ולאחר מכן אם הכל נכון שולח ללקוח את פרטי הניקוד של המשתמש.

תחילת משחק: כאשר לקוח רוצה להתחיל משחק הוא שולח בקשה לסרבר לפי פרוטוקול, אז הסרבר מוסיף את המשתמש לרשימת המחכים למשחק, ואם הרשימה בגודל 2 מוסיף אותם לרשימה לשליחת תחילת משחק ומוציא אותם מרשימת לחכות למשחק.

תחילת טורניר: כאשר לקוח רוצה להתחיל טורניר הוא שולח בקשה לסרבר לפי פרוטוקול, אז הסרבר מוסיף את המשתמש לרשימת המחכים לטורניר, ואם הרשימה בגודל 4 יוצר אובייקט טורניר וביצירת הטורניר מכניס אותם(4 socket) בתור ערכים התחלתיים שהאובייקט יודע ליישם לרשימות והתאמות, ולאחר מכן מוסיף את כל ארבעת המחכים לטורניר לרשימה של שליחת התחלת משחק.

סבב טורניר שני: כאשר נגמר משחק של סבב ראשון בין שני שחקנים, הסרבר מוריד את המפסיד מן הטורניר ומוסיף את המנצח לסבב השני ואת הנקודות שהרוייח לאובייקט הטורניר, אם הרשימה של הסבב השני בגודל 2 אז האובייקט מתאים את המנצחים של הסבב הראשון אחד אל השני, והסרבר מוסיף אותם לרשימת התחלת משחק.

סרבר חיצוני: לפרויקט שלנו בשונה משאר הפרויקטים רצינו שחווית המשתמש תהיה כמה שיותר גבוה, בעיצוב, בכמות הדמויות, ברמת המשחק ובעיקר ברמת הרשתות, בגלל זה הקמנו שרת אמיתי שרץ כל הזמן, שבעזרתו המשתמשים אשר נרשמים ישמרו תמיד ותמיד משתמשים יוכלו לשחק במשחק שלנו ובכל מדינה בעולם.

תיעוד הקוד:

1. היה לנו בעיה כאשר רצנו בשרת על הרשימות של הסוקטים שאנחנו רוצים לשלוח להם הודעה ולאחר ששלחנו אנחנו מחקנו אותם מהרשימה, אם מוחקים באמצע לולאה זה הורס את הלולאה וגורם ללולאה לדלג על תאים מסוימים ברשימה.

פתרונינו היה להעתיק כל רשימה שאנחנו רוצים לרוץ עליה ונרוץ על הרשימה המועתקת וכאשר אנחנו רוצים למחוק איבר מסוים נמחק מהרשימה המקורית.

2.כאשר אנחנו רצים על הרשימות של הדמויות ומזיזים אותם ומורידים להם חיים אז הקוד נראה כך:

my\_creachers\_new = run\_creachers(my\_creachers, enemy\_creachers, enemy\_castle)

enemy\_creachers\_new = run\_creachers(enemy\_creachers, my\_creachers, my\_castle)

my\_creachers = my\_creachers\_new

enemy\_creachers = enemy\_creachers\_new

עשינו שתי רשימות חדשות בשביל שמתי שאנחנו רצים על הרשימה השנייה זה לא יהיה עם השינויים שקרו כבר ברשימה, ורק אחרי שרצנו על הכול בשביל הסבב אנחנו מעדכנים את הרשימות.

הערכת הפתרון לעומת התכנון והמלצות לשיפורו :

אם היה לי יותר זמן הייתי רוצה ליצור גם גרסת יישומון למשתמשי האיפון בעזרת Xcode ו Swift.

בהתחלה התכנון היה לעשות משחק רגיל בין שני שחקנים, לאחר שראיתי שהספקתי וביצעתי את התכנון שלי, חשבתי איך אוכל לשפר את הפרויקט מבחינת חווית משתמש ורשתות ולאחר מכן הוספתי מסך רישום/כניסת משתמש וטבלת ניקוד אישי וטבלת מובילים ולאחר שכתבתי ועיצבתי את הכול והכל עבד כשורה חשבתי איך אוכל עוד יותר לשפר את המשחק מבחינת רשתות בעיקר ומבחינת חווית המשחק עצמו, אז שמעתי לעצתה של מורתי עדינה ובניתי טורניר.

לבסוף ראיתי שהפרויקט שלי מלא בתוכן כמו שרציתי ולא עומד בדרישות הרגילות אלא מעבר.

תיאור של הממשק למשתמש - הוראות הפעלה :

אם רוצים לבדוק את הפרויקט, ראשונה מריצים את הקובץ server\_game.py

ולאחר מכן את Games Of Magic.py פעמיים ל משחק רגיל ו - 4 פעמים למשחק טורניר.

אם רוצים לשחק במשחק עצמו מריצים רק את Games Of Magic.py והוא כבר מתחבר לשרת החיצוני.

כאשר הקובץ מורץ יעלה מסך רישום או כניסת משתמש, לאחר רישום או כניסה יעלה מסך פתיחה אשר יש באפשרותו מספר אופציות:

1. כפתור PLAY – משחק רגיל בין שני שחקנים.

2. כפתור  - סיפור המשחק.

3. כפתור  - הוראות המשחק.

4. כפתור  - טבלת שלושת המובילים.

5. כפתור  - מסך ניקוד אישי(ניקוד ומספר ניצחונות).

6. כפתור  - משחק טורניר(שני סבבים, ארבעה שחקנים, בכל משחק שני שחקנים, משתתפי הסבב השני הם מנצחי הסבבים הראשונים).

מבט אישי של העבודה ותהליך הפיתוח :

למדתי מהפרויקט, את היכולות התכנות שלי לעומק, הבנה באובייקטים, ה ״מוח״ שמאחורי הפיתון, עבודה מרוכזת ומייעלת, איך לעבוד בפרויקט, איך לתכנן את העבודה בצורה מסודרת, הבנה בגרפיקה ולמידה עצמאית לעומק.

ביבליוגרפיה :

Stack overflow -<https://stackoverflow.com/>

-  גבהים רשתות<http://data.cyber.org.il/networks/networks.pdf>

קוד התוכנה :

https://github.com/daniel842/finall-game-cyber.git