

Magic Maze Online



מגיש: תומר דרור

תז: 328363163

שם החלופה: תכנון ותכנות מערכות

מורה: ניר סליקטר

תאריך הגשה 30.5.24



תוכן עניינים

מבוא.....	3
המשחק.....	3
מבני נתונים.....	5
תקשורת.....	7
ממשק משתמש.....	7
פרוטוקולים.....	8
מענה על דרישות.....	12
טכנולוגיות.....	14
ביבליוגרפיה.....	16
רפלקציה.....	18
קוד.....	19
נספח.....	143

מבוא

הפרויקט שלי הוא משחק רב משתמשים real time בו 1-8 שחקנים ללא יכולת תקשור צריכים לעבוד ביחד כדי לגלות את מפת הקניון, לגנוב מהחנויות ולברוח לפני שהזמן נגמר, במשחק אין תורות ואפשר לפעול מתי שרוצים על כל אחת מארבעת הדמויות, אך רק בדרכי התנועה הייחודיים הניתנים לך מתחרים נגד הזמן. זה משחק מתסכל שדורש עבודת צוות תחת לחץ זמן ועכשיו יצרתי לו גרסה אונליין.

המשחק

משחק "מאג'יק מיז" (Magic Maze) הוא משחק לוח שיתופי, שבו השחקנים משתפים פעולה כדי להוביל ארבע דמויות - קוסם, לוחם, גנב ואלף - דרך קניון קסום למציאת וגניבת הציווד שלהם מבלי להיתפס על ידי השומרים. הנה תיאור כללי של המשחק:

התחלת המשחק: כל שחקן מקבל קלף פעולה אחד או יותר, המאפשר לו לבצע פעולה מסוימת (לנוע צפונה, דרומה, מזרחה, מערבה, להשתמש בגרמי מדרגות וכו'). כל הפעולות מתבצעות בשיתוף פעולה בין השחקנים, אך אין תקשורת מילולית או סימנים פיזיים מותרים בזמן המשחק. המשחק מתחיל בלוח התחלתי אחד, ועל השחקנים להוסיף עוד חלקי לוח כדי לגלות את הקניון המלא.

מטרת המשחק: למצוא את הפריטים הדרושים לכל דמות ולהוציא אותם מהקניון דרך היציאות המיועדות לפני שנגמר הזמן.

הזמן: המשחק מתנהל תחת מגבלת זמן שמוגדרת מראש. במרכז השולחן יש שעון חול, וכשנגמר הזמן יש לבצע פעולת "עצירה" שבה השחקנים יכולים לתקשר ולתכנן את המהלכים הבאים.

מהלך המשחק: כל שחקן, בתורו, מבצע פעולה המותרת לו לפי קלף הפעולה שלו, על ידי הזזת דמות אחת בלוח. על השחקנים לשתף פעולה ולתכנן את המהלכים בצורה מושכלת כדי לאפשר לכל דמות להגיע אל הפריטים שלה ולברוח מהקניון בזמן.

סיום המשחק: המשחק נגמר כאשר כל הדמויות הצליחו לאסוף את הפריטים שלהן ולצאת מהקניון, או אם הזמן נגמר לפני שהצליחו.

מבני נתונים

במשחק המגרש כל פעם משתנה נפתח ומורכב אחרת לכן מבנה הנתונים שלי צריך להיות מותאם לכל אופציה, הדרך הפשוטה ביותר היא פשוט לשמור את כל המשבצות השונות במערך ענק פסלתי את האופציה הזאת בגלל כמות המקום בזיכרון, יש במשחק עד איזה 30 חלקי מגרש שכל אחד מורכב מ 4X4 משבצות והמערך צריך להכיל אופציה להיפתח לארבעה כיוונים מה שאומר שאני צריך מערך באורך 240X240 שזה 57600 משבצות שזה דרישה גבוהה מאוד מהזיכרון במיוחד אם צריך לשנות בזריזות ולקבל תגובה במהרה. לכן עברתי למבנה נתונים אחר המבנה הכללי הוא מערך דו כיווני דו מימדי (לא מלא) בשביל המשבצות. משבצת היא ריבוע, מכילה מצביע לכל אחד מהמשבצות שאפשר לעבור אליהן מהמשבצת הזו, כמו כן המשבצת מכילה מידע על מה אפשר לעשות במשבצת/ מה קורה כשדורכים עליה.

ארבעת הדמויות של כלי המשחק ממוקמות על משבצות ולכל אחד מצביע למשבצת עליה הוא נמצא. כיוון שיש רק ארבעה דמויות וכיוון שצריך לנהל את איזה חלק מגרש צריך להיפתח יש גם אובייקט של מגרש שמכיל את הדמויות, את החלקים הפתוחים, את החלקים שעוד צריכים להיפתח וכל הפורטלים שיש לכל אחד מהדמויות. האובייקט הזה במבנה של סינגלטון.

ישנה בעייה גדולה הנובעת מהמבנה נתונים, דמיין ויש לי שמונה חלקי מגרש (4X4) במעגל ונפתח חלק מגרש נוסף במרכז אני צריך לדעת לחבר אותו גם לשאר חלקי המגרש וצריך למצוא את המשבצות אליהן לחבר לכן אני שומר גם את חלקי המגרש כדי להקל על החיפוש למשבצת הנפתחת. לגבי מציאת החלקים לחיפוש אני חיפשתי מיגוון דרכים לחיפוש אך האופציה הכי טובה שמצאתי בלי לשנות את מבנה הנתונים הייתה בסיבוכיות זמן של $O(n \log n)$ וסיבוכיות מקום של $O(n)$ ולכן

הבנתי שיהיה יותר פשוט לשנות את מבנה הנתונים ולשמור רשימה של החלקי מגרש שנפתחו. כפי
מה שאמרתי יש אובייקט של חלק מגרש הוא מכיל את המיקום שלו בשביל החיבור עליו הוסבר קודם,
את חלקי המגרש הסמוכים אליו, ורשימת הפתיחות שמתחברות לחלקים אחרים. בשביל הבנייה של
החלק מגרש אני משתמש בטבלת גיבוב כדי להתאים בין xy זמני ויחסי לחלק מגרש למשבצת וככה
אני יודע שאני לא יוצר עותקים של חלק מגרש. בעייה נוספת היא בעיית הסיבוב, את החלקים צריך
לחבר בכל אוריינטציה לכן צריך להיות אופציה לסובב את החלק מגרש, אם חלק המגרש כבר קיים
לסובב אותו זה בעייה לכן יש מבנה נתונים נוסף שזה קדם חלק מגרש שמאחסן מערך דו מימדי של
מספרים המאחסנים מידע על המשבצות כגון מה המשבצת עושה ואיפה נמצאים הקירות.

ראה נספח בשביל תמונה של מודל של המבנה נתונים

תקשורת

התקשורת מורכבת משני חלקים, תקשורת בין השרת לצרכן ובין הצרכן לממשק משתמש. התקשורת ב-Winsock לשרת רב משתמשים היא חיונית לניהול יעיל של חיבורים מרובים בו-זמנית. Winsock, ממשק תכנות יישומים (API) עבור תקשורת ברשתות במערכות הפעלה של Windows, מאפשר לשרת לקבל ולנהל חיבורים מרובים מלקוחות שונים באמצעות מנגנונים כמו sockets. שרת רב משתמשים משתמש ב-Winsock כדי ליצור socket ולהאזין לחיבורים נכנסים. כאשר מתקבל חיבור חדש, השרת יכול להקצות thread נפרד או להשתמש במנגנון אסינכרוני לטיפול בכל חיבור, מה שמאפשר ניהול חיבורים מרובים ביעילות גבוהה יותר. IPC (תקשורת בין תהליכים) באמצעות זיכרון משותף (shared memory) בין הלקוח לבין הפרונטאנד היא טכניקה נפוצה להעברת נתונים בצורה מהירה ויעילה בתוך מערכת הפעלה אחת. באמצעות shared memory, תהליכים שונים יכולים לגשת לאותו אזור זיכרון ולקרוא או לכתוב נתונים בו ללא צורך במנגנוני תקשורת מסובכים כמו sockets או pipes. זה משפר את ביצועי התקשורת בין הלקוח לפרונטאנד, מכיוון שהעברת הנתונים נעשית ישירות בזיכרון ללא צורך במעברי הקשר נוספים, ומפחיתה את זמני ההשהיה ואת העומס על המעבד.

ראה נספח בשביל תמונה של מודל התקשורת

ממשק משתמש

הממשק משתמש מאפשר גישה לשחקן אין לי הרבה מה להרחיב מעבר להגיד שעשיתי אותו ב-pygame כי פחות אכפת לי מהמהירות של זה להתעדכן כי אם זה שוגה זה רק במסך שלך ולא משפיע על הרצת המשחק. ראה נספח בשביל תמונה של חלק מגרש וגם את זרימת האפליקציה.

פרוטוקולים:

פקודות קיימות:

- Login •
- Register •
- Start •
- Move •
- getCharacter •
- Get •
- Open •
- CommunicationError •

פקודות תוך משחקיות:

פקודות המועברות דרך השרת לכל שאר המשתתפים, פקודות אלו מייצגות פעולה במשחק שעל הדמיות לעשות.

לכל הפקודות יש את המבנה הבא:

command	variables(might be	Players Movement
---------	--------------------	------------------

	(multiple	Ability
--	-----------	---------

Command : אחת מהפקודות הבאות

Move •

getCharacter •

Get •

Open •

Variables : משתנה מפקודה לפקודה ראה למטה לכל אחד

Players Movement Ability : היכולת תנועה של השחקן מבצע הפעולה, מפורק ככה

canMoveUp	canMoveDown	canMoveRight	canMoveLeft	canUsePortals	canUseEscalators	canOpenFieldPiece
-----------	-------------	--------------	-------------	---------------	------------------	-------------------

Move - בקשה להזיז דמות למיקום ספציפי

Move	variables	משמעות
	color	צבע הדמות המוזזת

	pos	ה id של הtile
--	-----	---------------

דוגמא

move&green\$12\$1\$0\$1\$0\$1\$0\$1

מבקש להזיז את הדמות הירוקה לtile שהid שלו הוא 12

getCharacter - בקשה לקבלת הid של דמות. בקשה זו נשלחת רק מהFE לBE

getCharacter	variables	משמעות
	color	צבע הדמות

דוגמא

getCharacter&green\$1\$0\$1\$0\$1\$0\$1

מבקש לקבל את מיקום הדמות הירוקה

get - בקשה לקבלת צבע של דמות במיקום מסוים. בקשה זו נשלחת רק מהFE לBE

getCharacter	variables	משמעות
	כסד	Id של ה tile

דוגמא

get\$12\$1\$0\$1\$0\$1\$0\$1

מבקש לקבל את mcg הדמות במיקום 12

Open - בקשה לפתיחת חלק מגרש חדש

משמעות	variables	open
צבע הדמות שפותחת את החלק	color	

דוגמא

open&green\$1\$0\$1\$0\$1\$0\$1

מענה על דרישות

תכנות מונחה עצמים:

קיים בקוד המחלקות הבאות: main, field, fieldPiece, Character, Player, movementAbility, Tile, Constants, Utils, PreFieldPiece, Server.

תקשורת

יש שרת מרבה לקוחות המחבר בין כל המשתמשים
יש תקשורת memory share IPC לתקשר בין frontend ו backend (מוסבר בטכנולוגיות)

מערכת הפעלה

יש שימוש בthread בשביל שתוכל לקבל מהשרת פקודות תוך כדי שיש תקשורת עם frontEnd
והרצת המשחק עצמו ושימוש בthreads בשרת בשביל השרת המרובה לקוחות

אבטחה

כדי להבטיח את אבטחת המידע המועבר על ידי שילוב של AES ו-RSA בהבטחת נתונים.
AES (Advanced Encryption Standard) הוא אלגוריתם הצפנה סימטרי, כלומר אותו מפתח משמש גם להצפנה וגם לפענוח של הנתונים. AES הוא מהיר ויעיל לשימוש עם כמויות גדולות של

נתונים. הוא נחשב לבטוח מאוד כאשר משתמשים בו עם מפתחות באורך מתאים (128, 192, או 256 סיביות).

RSA (Rivest-Shamir-Adleman) RSA הוא אלגוריתם הצפנה אסימטרי, כלומר הוא משתמש בזוג מפתחות: מפתח ציבורי להצפנה ומפתח פרטי לפענוח. RSA מבוסס על הבעייתיות של פירוק מספרים גדולים לגורמים ראשוניים, והוא איטי יחסית לאלגוריתמים סימטריים כמו AES, במיוחד כשמדובר בכמויות גדולות של נתונים. השילוב של AES ו-RSA

ממשק משתמש

במשחק יש ממשק משתמש ב-pygame כדי לתת למשתמש דרך להפעיל את המשחק.

טכנולוגיות

:Front End

:Pygame

Pygame היא ספרייה פופולרית ב-Python לפיתוח משחקי מחשב ויישומים אינטראקטיביים. היא מספקת כלים קלים לשימוש לעבודה עם גרפיקה, צלילים ואירועים. Pygame נבנית על ספריית SDL (Simple DirectMedia Layer) ומאפשרת יצירה של יישומים גרפיים עם ממשק פשוט ונוח.

:IPC memory share

תקשורת בין תהליכים (IPC) הוא מנגנון המאפשר לתוכנות שונות לתקשר זה עם זה ולסנכרן בתוך אותו מכשיר. memory share היא אחת השיטות המשמשות עבור IPC. השיטה הזו מקצה את אותו קטע זיכרון מוגדר מראש לכמה תוכנות.

Backend

:++C

++C היא שפת תכנות כללית בעלת יכולות מתקדמות המאפשרות תכנות מונחה עצמים, תכנות פרוצדורלי ותכנות גנרי. היא התפתחה מתוך שפת C ונשמרת התאימות שלה עם C במידה רבה.

C++ נמצאת בשימוש נרחב לפיתוח מערכות תוכנה מורכבות, כולל מערכות הפעלה, דפדפני אינטרנט, משחקים, תוכנות למכשירים משובצים ועוד.

הסיבה שאני משתמש בC++ היא כהשפה מהירה ביותר וחשוב לי במשחק הזמן תגובה ויפעל חלק בהשוואה ל python שימוש בC++ מהיר פי 10-100 בשימוש בכלי בסיס של השפה.

Winsock Cpp:

Winsock (Windows Sockets) הוא ממשק תכנות יישומים (API) המשמש לתקשורת רשת במערכות הפעלה Windows. זהו סט של פונקציות וספריות המאפשרות לתוכנות לתקשר דרך פרוטוקולים כמו TCP/IP ו-UDP.

SQL - Database:

הוא מנוע מסד נתונים קל משקל, פשוט לשימוש ובעל יכולות רבות. הוא מאוד פופולרי בעיקר בעקבות היכולת שלו להיות מוטמע באפליקציות קטנות ובעקבות זאת הוא משמש רבות פעמים כמסד נתונים עבור אפליקציות ניידות וכלי אינטרנט. SQLite3 נתמך על מגוון רחב של פלטפורמות ומערכות הפעלה, והוא מצוי בקוד המקור של פייתון, בין היתר, כדי לתמוך במסדי נתונים מוטמעים. המאפיינים הבולטים של SQLite3 כוללים קלות בהתקנה ושימוש, תמיכה ב-SQL, אוטומטיות וניידות. בקיצור, SQLite3 הוא פתרון מוצלח עבור פרויקטים קטנים ואפליקציות ניידות שדורשות מסד נתונים קל משקל ונייד.

ביבליוגרפיה:

<https://chatgpt.com/>

https://www.w3schools.com/cpp/cpp_class_methods.asp

https://www.w3schools.com/cpp/cpp_output.asp

https://www.w3schools.com/cpp/cpp_classes.asp

https://www.w3schools.com/cpp/cpp_oop.asp

https://www.w3schools.com/cpp/cpp_inheritance_multiple.asp

https://www.w3schools.com/cpp/cpp_arrays.asp

https://www.w3schools.com/cpp/cpp_inheritance_multiple.asp

https://www.w3schools.com/cpp/cpp_strings.asp

https://www.w3schools.com/cpp/cpp_structs.asp

https://www.w3schools.com/cpp/cpp_constructors.asp

https://www.w3schools.com/cpp/cpp_booleans.asp

<https://www.educba.com/c-plus-plus-thread-sleep/>

<https://www.codementor.io/@hbendali/c-c-macro-bit-operations-ztrat0et6>

<https://stackoverflow.com/questions/18559028/undefined-reference-to-imp-wsacle>

[anup](#)

<https://www.geeksforgeeks.org/typedef-in-cpp/>

<https://stackoverflow.com/questions/612328/difference-between-struct-and-typedef-struct-in-c>

<https://www.scaler.com/topics/cpp-sleep/>

<https://www.geeksforgeeks.org/sql-using-c-c-and-sqlite/>

<https://www.geeksforgeeks.org/scope-resolution-operator-in-c/>

<https://www.javatpoint.com/cpp-date-and-time>

<https://www.quora.com/How-do-I-use-terminal-commands-in-C++>

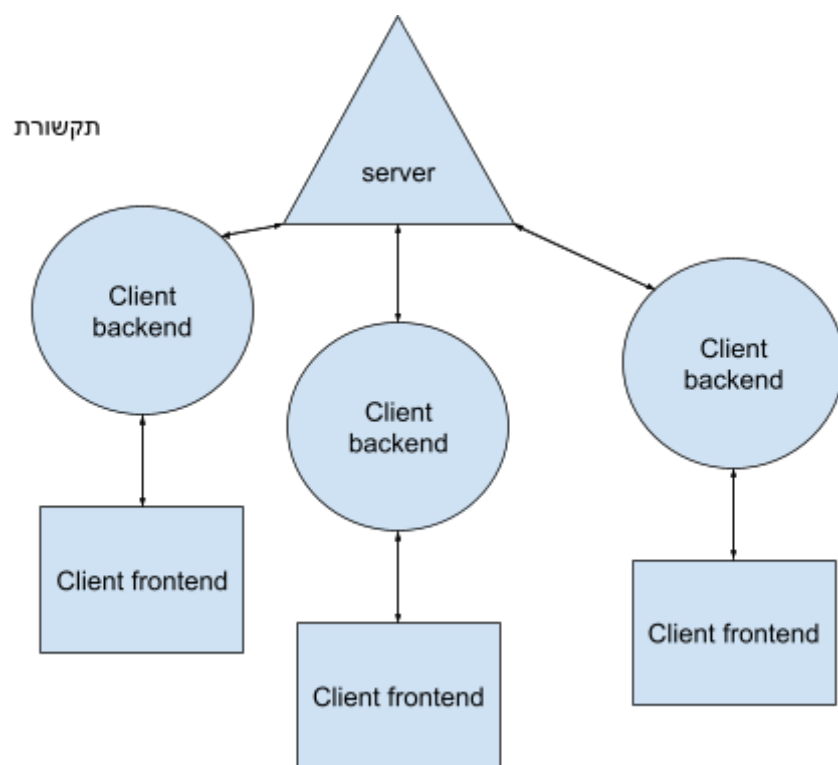
<https://www.geeksforgeeks.org/multithreading-in-cpp/>

רפלקציה

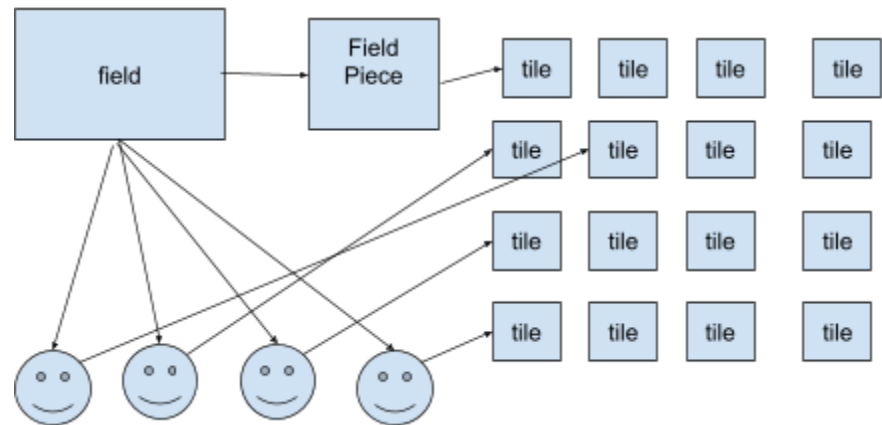
הפרויקט שלי עוסק בפיתוח משחק רב-משתתפים בזמן אמת, שבו השחקנים צריכים לשתף פעולה כדי לגלות מפת קניון, לגנוב מחנויות ולברוח לפני תום הזמן. בניית המשחק ותהליכי הפיתוח שלו הסתיימו בהצלחה, אך הם היו מלווים באתגרים מסוימים. תהליך העבודה היה משובח, כאשר נתקלת בקשיים במציאת דרך יעילה לתקשר בין הלקוח לשרת. כדי לפתור את הבעיה, החלטת להשתמש ב-memory share IPC, אשר התברר כאמצעי תקשורת נוח ויעיל. בנוסף, בתהליך זה למדת בעומק תכנות בשפת ++c ופיתוח אפליקציות בעזרת pygame, המהווים כלי חשובים שיעזרו לך גם בעתיד. למרות האתגרים, תהליך הלמידה היה מרתק ומועיל.

נספח

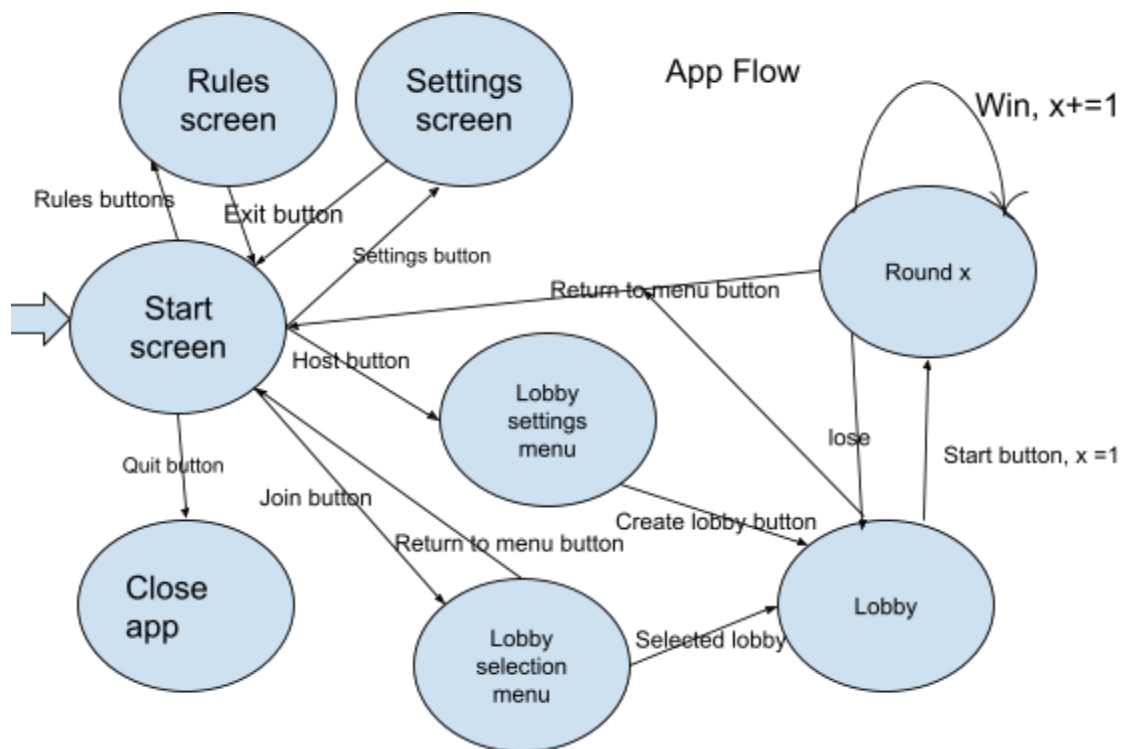
מודל תקשורת:



backend



מודל המשחק



רצף משחק

