ספר פרויקט

שם התלמיד: נדב כרמונה

ת.ז זהות: 324187301

שם בית ספר: הדרים הוד השרון

שם המורה: אלי גולדשטיין

תוכן עניינים

[מבוא 3](#_Toc104642788)

[מערכת השרתים ותקשורת עם השחקנים 3](#_Toc104642789)

[מטרת הפרויקט 4](#_Toc104642790)

[היעדים של הפרויקט 4](#_Toc104642791)

[מהות המערכת 4](#_Toc104642792)

[הפעלת המערכת 4](#_Toc104642793)

[מבנה נתונים שבשימוש 4](#_Toc104642794)

[תקשורת בין כל השרתים למשתמש 6](#_Toc104642795)

[התחברות למערכת 6](#_Toc104642796)

[מהלך משחק 6](#_Toc104642797)

[Client מתנתק מהמערכת 6](#_Toc104642798)

[מדריך למשתמש 7](#_Toc104642799)

[פתיחת וכניסה למערכת 7](#_Toc104642800)

[המשחק עצמו 10](#_Toc104642801)

[מבנה של הפרויקט 11](#_Toc104642802)

[Database\_server 11](#_Toc104642803)

[Connection\_server 11](#_Toc104642804)

[Game\_server 13](#_Toc104642805)

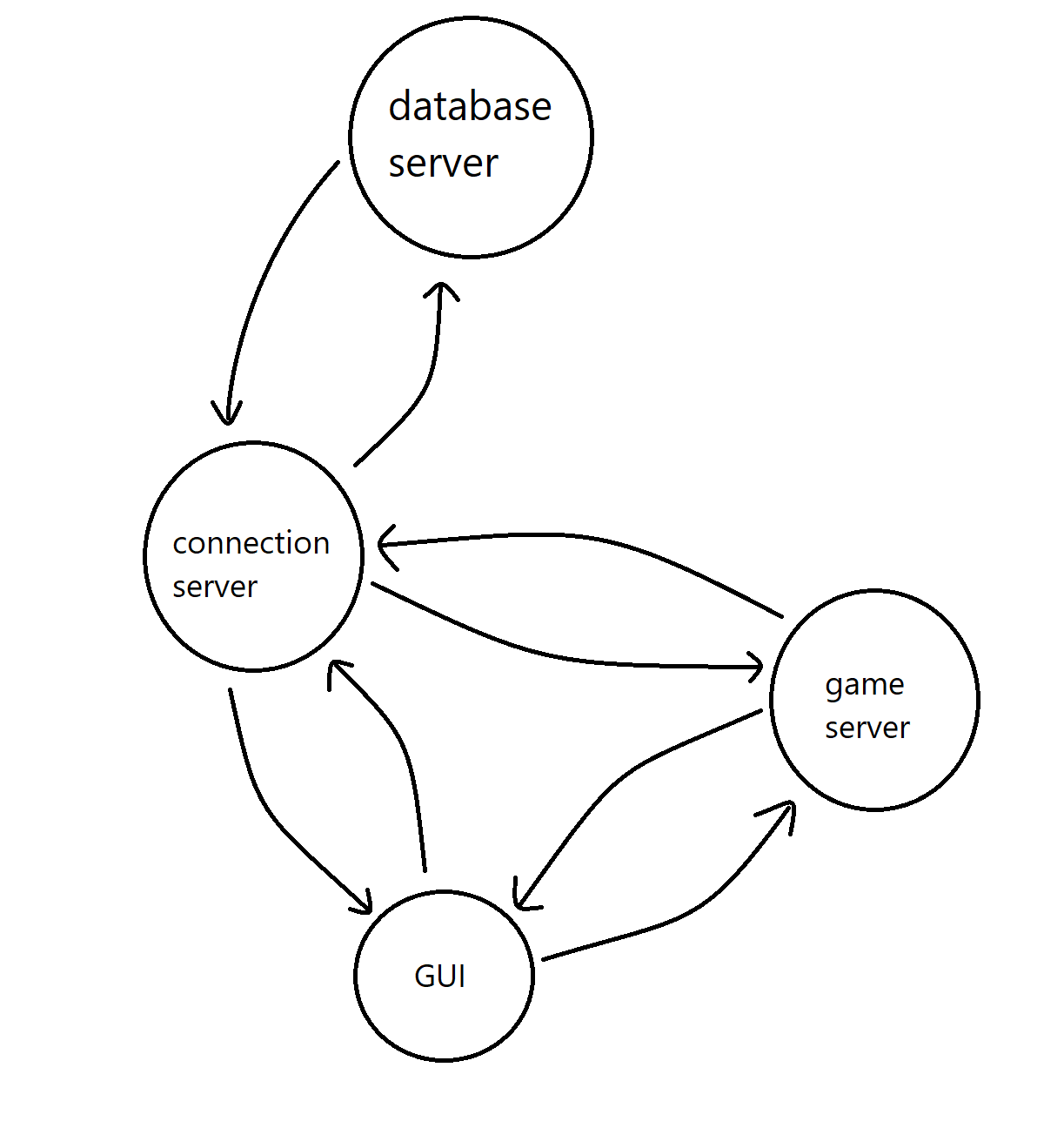
[GUI 14](#_Toc104642806)

[קוד הפרויקט 16](#_Toc104642807)

[Database server code 16](#_Toc104642808)

# מבוא

## מערכת השרתים ותקשורת עם השחקנים

database server הינו השרת הראשי, בנוסף הוא מתקשר עם ה-MySQL .

\*החיצים מראים את התקשורת בין הצדדים השונים

connection server הוא השרת המגשר בין כל הפניות לשרת הdatabase, הוא בודק את תקינות ההודעות ומוודא שתהיה סביבת עבודה נקייה לשרת הdatabase.

game server הוא שרת המשחק עצמו, הוא מריץ את המשחק למשתמשים.

GUI הוא השחקן עצמו, כל משתמש מריץ GUI ובעזרתו מתחבר למערכת.

## מטרת הפרויקט

ניהול משתמשים במערכת. שחקנים יכולים להתחבר או ליצור משתמשים במערכת ולהיכנס למשחק. המשחק עצמו הוא משחק יריות דו ממדי שבו כל אחד משחק ריבוע שיכול לזוז ולירות, המטרה היא לפגוע בכמה שיותר שחקנים אחרים וכך לצבור נקודות. השחקנים יכולו בנוסף להתחבר לאתר אינטרנט מיוחד שבו בעזרת התחברות למערכת יראו את ממוצע הניקוד שלהם למשחק ומיקומם ביחס לשחקנים האחרים. במקרה של שכחת סיסמא יוכל השחקן לבקש שהמערכת תשלח לו מייל עם סיסמא חדשה להתחברות.

## היעדים של הפרויקט

על הפרויקט לנהל משתמשים, עד כ-20, ללא תקיעות והאטה בזמן הריצה, ללא לאבד מידע על השחקנים. על שרת הconnection לסננן את כל ההודעות הלא תקינות מהמערכת. הgame server יוכל להריץ את המשחק לעד 5 שחקנים בו זמנית ללא האטות ותקיעות.

## מהות המערכת

ניהול משתמשים במערכת. שחקנים יכולים להתחבר או ליצור משתמשים במערכת ולהיכנס למשחק. המשחק עצמו הוא משחק יריות דו ממדי שבו כל אחד משחק ריבוע שיכול לזוז ולירות, כל השחקנים נמצאים במסך שחור ובו קירות אפורים המשמשים כקירות ומחסה, המטרה היא לפגוע בכמה שיותר שחקנים אחרים וכך לצבור נקודות. כששחקן נכנס למערת מוצג בפניו home screen אשר בו יכול השחקן לראות את ממוצע הניקוד שלהם למשחק, ממתי המשתמש שלהם קיים ומהו מיקומם ביחס לשחקנים האחרים. מה-home screen השחקנים יכולים להתחבר למשחק או להתנתק מהמערכת.

## הפעלת המערכת

ה- database server וה- connection server נפתחים ידנית במחשב אחד. ה-GUI נפתח על ידי השחקנים מכל מחשב שהם נמצאים. ה- game server נפתח ונסגר על ידי המערכת לפי הצורך.

## מבנה נתונים שבשימוש

השתמשתי במנה הנותנים של MySQL. בחרתי דווקא ב MySQL כיוון שהוא פשוט ונוח לשימוש והיה לי רק את הצורך הבסיסי לפעולות פשוטות של קריאה כתיבה ומחיקה ממבנה נתונים. לא התעמקתי בחיפוש אחר מבנה נתונים בגלל הצורך הבסיסי שלי בו.

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

## תקשורת בין כל השרתים למשתמש

### התחברות למערכת

הclient(GUI) שולח הודעה לconnection server שבוקר את תקינות ההודעה ומעביר אותה לdatabase server שבודק ברשימה בMySQL אם המשתמש יכול להתחבר. אם הוא מנסה להיכנס למשתמש שכבר קיים שרת הdatabase בודק אם הסיסמא שהclient שלח תואמת את זה של המשתמש והאם המשתמש לא מחובר כרגע, ואם הclient מנסה ליצור משתמש חדש שרת הdatabase בודק אם לא קיים משתמש אחר עם שם דומה. אז שרת הdatabase שולח הודעה לconnection server שמעביר את ההודעה לclient שניסה להתחבר וה client. אם הclient יכול להתחבר הוא פותח home screen אשר ממנו יכול לראות את ממוצע הניקוד שלו ומיקומו ביחד למשתמשים אחרים, בנוסף יכול הclient להיכנס למשחק או להתנתק. אם הclient נכנס למשחק הוא מתנתק מהconnection\_server ומתחבר לgame\_server.

### מהלך משחק

במשך המחשק הclient שולח את הinput שלו לgame\_server והgame\_server מחזיר את מה על הclient להדפיס על המסך שלו. בסוף כל משחק או כאשר הgame\_server מתרוקן לגמרי משחקנים הוא מעביר את מצב הניקוד של כל שחקן לconnection\_server שבודק את ההודעה ומעביר אותה הלאה לdatabase\_server שיעדכן את היסטוריית הניקוד וממוצע הניקוד של כל משתמש.

### Client מתנתק מהמערכת

Client שולח הודעת התנתקות לgame\_server, מתנתק ממנו ומתחבר לconnection\_server בחזרה לhome screen. משם הclient יכול להתחבר למשחק בחזרה או להתנתק סופית. אם מתנתק סופית שולח הודעה על כך לconnection\_server שמעביר את ההודעה על כך לdatabase\_server שמעדכן את ההתנתקות של המשתמש מהמערכת.

## מדריך למשתמש

### פתיחת וכניסה למערכת

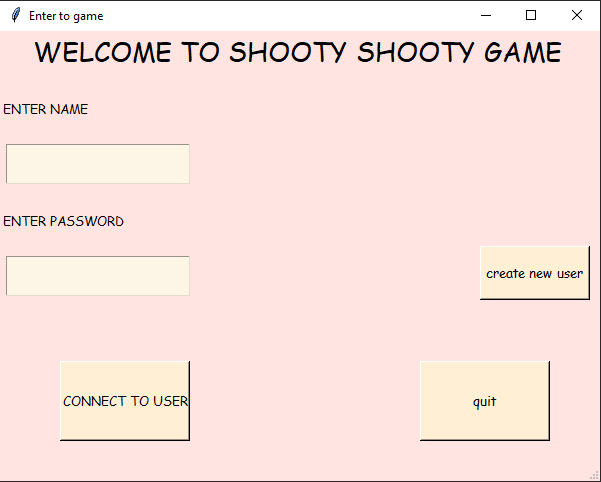
 -מתקינים את הקובץ GUI.exe ופותחים אותו.

-נפתח חלון להכנת IP תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

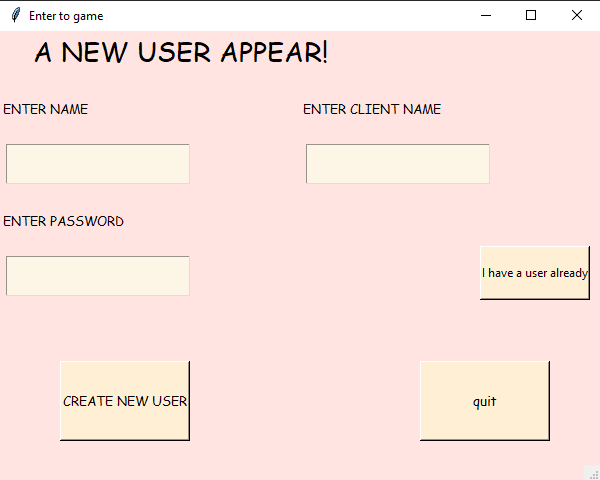
כותבים שם את הIP של הconnection server ואז לוחצים על כפתור ה"ENTER" עם העכבר.

<-הconnection server כותב את הIP שלו כשנפתח.

 -נפתח חלון התחברות :

מכניסים שם משתמש וסיסמא ולוחצים על הכפתור "ENTER GAME". בלחיצה על הכפתור "quit" המסך ייסגר וריצת הקוד תסתיים.

במקרה ורוצים ליצור משתמש חדש לוחצים על הכפתור "create new user".

 -כשלוחצים על הכפתור "create new user" נפתח החלון הבא:

מכניסים שם משתמש(שם שלא קיים כבר במערכת), סיסמא ואת שם השחקן ולוחצים על הכפתור "CREATE NEW USER". בלחיצה על הכפתור "quit" המסך ייסגר וריצת הקוד תסתיים.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי -אם מכניסים את כל הפרטים נכון והמערכת מאשר את החיבור נפתח מסך בית:

מסך הבית מראה את שם השחקן שהוכנס, זמן יצירת השחקן, ממוצע הנקודות שלו למשחק ואת מיקומו בין שאר השחקנים מבחינת ממוצע הנקודות.

כדי להיכנס למשחק צריך ללחוץ עם העכבר על הכפתור "ENTER GAME", בלחיצה על הכפתור "quit" המסך ייסגר וריצת הקוד תסתיים.

### המשחק עצמו

כשנכנסים למשחק ממסך הבית נפתח המסך של המשחק:

הריבוע הכחולה היא הדמות של השחקן, שאר השחקנים הם ריבועים בצבע אדום.

כאשר במשחק הכפתורים שמשתמשים בהם הם:

W- להזיז את הדמות למעלה.

A- להזיז את הדמות שמאלה.

S- להזיז את הדמות למטה.

D- להזיז את הדמות ימינה.

עם העכבר ניתן לכוון את הדמות, ועם לחיצה אחת הלחצן השמאלי של העכבר ניתן לירות עם הדמות.

Esc- יציאה מהמשחק בחזרה למסך הבית.



# מבנה של הפרויקט

## Database\_server

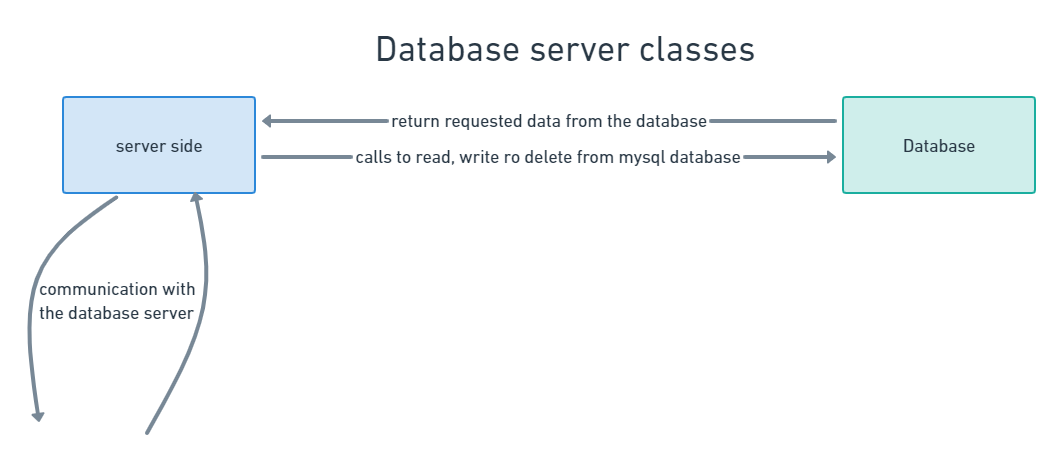
השרת הראשי האחראי על תקשורת עם MySQL וניהול ובקרה אחר כל השרתי משחק שקיימים.

השרת מכיל שתי מחלקות:

ה server side אחראי על כל התקשורת עם הconnection server והעברת המידע.

מחלקת ה Database אחראית על מבנה הנתונים של MySQL.

קורא בקשות של clients, מעביר מידע מהמבנה נתונים ומעדכן אותו לפי הצורך. אם לא קיים מבנה נתונים או שאינו תקין הוא יוצא מבנה נתונים חדש. בעל פעולות קריאה כתיבה ומחיקה ממבנה הנתונים של ה MySQL. השרת עוקב אחרי שרתי המשחק (game server) ויכול לשלוח שחקנים לפי השרת הכי פנוי, ואם אין שרת משחק פנוי או שאין שרת משחק בכלל שולח הודעה לפתוח אחד חדש.



## Connection\_server

שרת הכניסות משמש כגשר תקשורת בין שרתי המשחק לבין שרת הdatabase.

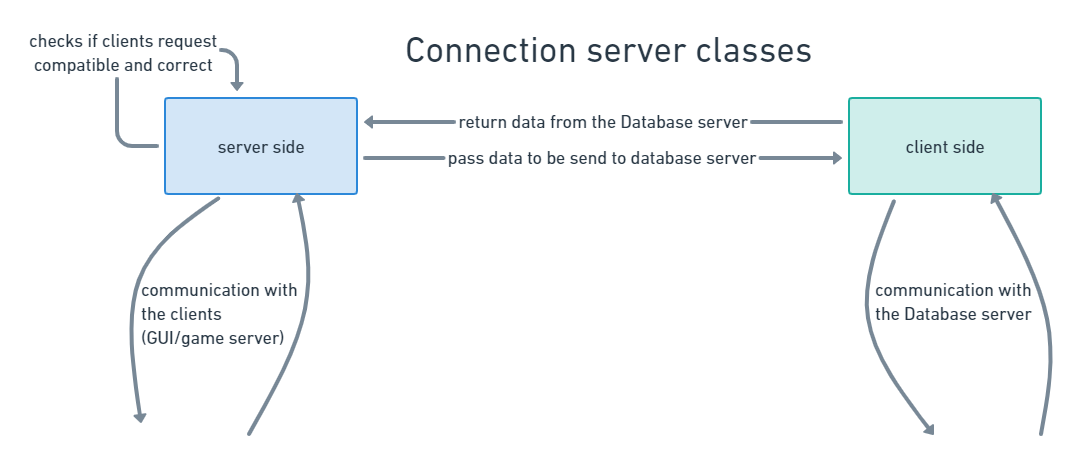
מכיל שני מחלקות:

server side אחראי על בדיקת ההודעות והתקשורת עם הGUI והgame server .

client side אחראי עם התקשורת עם ה database server.

מקבל הודעות מGUI ומ game server, בודק אם ההודעות תקינות ואם כן מעביר אותן לdatabase server לפי הצורך. ואת ההודעות של הdatabase server מחזיר ל GUI או game server המתאימים.

בנוסף השרת שומר רשימה של כל שרתי המשחק הפתוחים וכאשר שחקן חדש מתחבר הוא מפנה אותו לשרת הפנוי. ואם שרת מתנתק הוא מעדכן את הרשימה.



## Game\_server

השרת שמריץ את המשחק עצמו. מתחברים אליו שחקנים , עד 5. ישנה לולאה שחוזרת על עצמה של קבלת input מהשחקנים, חישוב מה שקורה במשחק עצמו, ושליחת המסך להדפסה לכל שחקן. במקרה של התנתקות גם אם אינה רצויה.

השרת בנוי מארבע מחלקות:

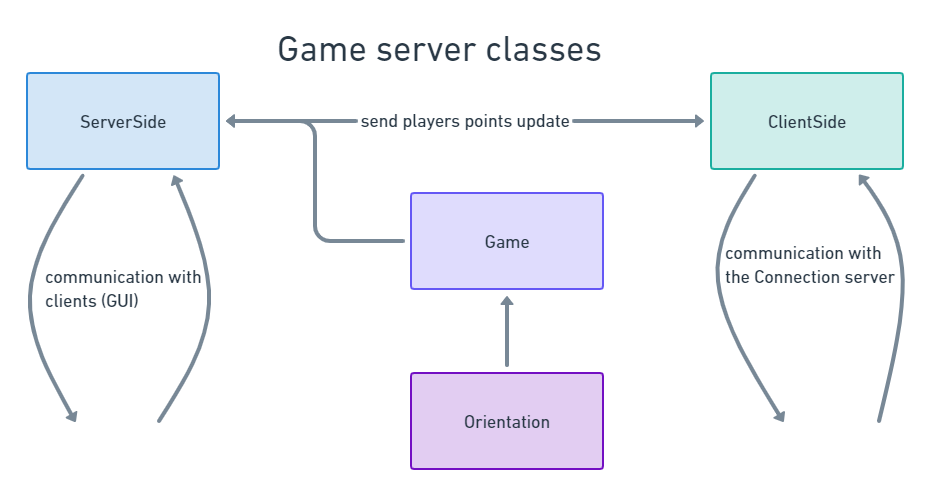
ServerSide הוא השרת עצמו, מנהל את כל הלקוחות במשחק, מקבל מהם את הקלטים שלהם ומחזיר להם מה להדפיס על המסך.

ClientSide אחראי על התקשורת עם הconnection server .

Game הינה המחלקה אשר מחשבת את כל מה שקורה במשחק (כל הקירות, היריות, השחקנים), היא מקבל את הקלטים של השחקנים ומחשבת ומזיזה את הדמויות.

המחלקה Orientation משמשת אך ורק לשליחת המידע להדפסת המשחק ללקוחות.

השרת משמש גם כ client לשרת הconnection, אשר אליו הוא שולח עדכון מצב הניקוד של כל שחקן בסוף משחק או כאשר כל השחקנים מתנתקים מהמשחק והשרת נסגר. שליחת המידע להדפסה ל GUI נעשית בעזרת ביצוע פעולת pickle שעושה serialize לאובייקט Orientation אשר נמצא בשניהם. אובייקט ה Orientation מכיל את כל התכונות של צורה שצריך להדפיס.



## GUI

השחקן עצמו, כקובץ executable אשר הלקוח מריץ, דרכו הוא מתחבר לשרת הconnection להתחברות וממנו לשרת המשחק. הקוד של ה GUI מכיל חלונות tkinter אשר מוצגים בכניסה למערכת וחלון pygame למסך של המשחק עצמו.

ה GUI מורכב משמונה מחלקות:

TopLevelMother משמש כמחלקת האם של TopLevel1 ו- TopLevel2.

TopLevel1 אחראי על מסך ה tkinter שנפתח כשמתחבר למשתמש, מתקשר כלקוח עם הconnection server .

TopLevel2 אחראי על מסך ה tkinter שנפתח כאשר יוצר משתמש חדש מתקשר כלקוח עם הconnection server .

IpCatcher אחראי על מסך ה tkinter שנפתח בתחילת ריצת הקוד כשמתבקש להכניס את ה IP של הconnection server .

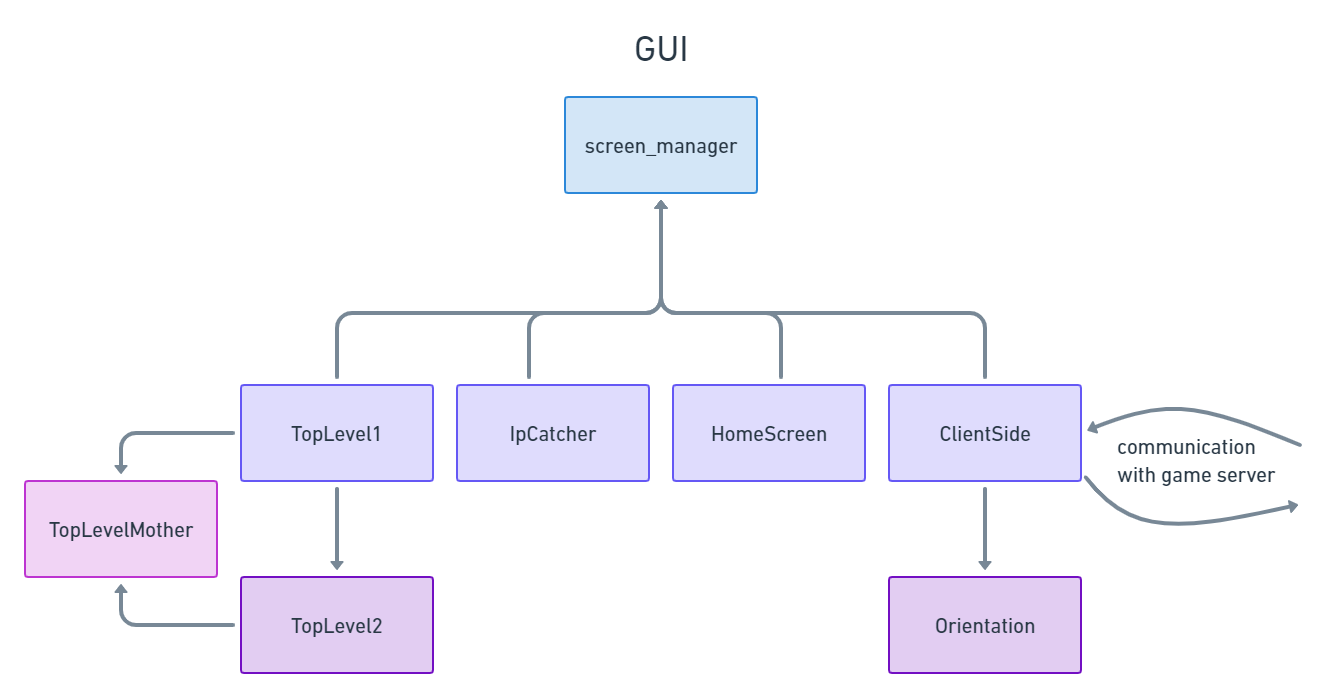
HomeScreen אחראי על מסך הבית (tkinter) שנפתח כשמתחבר למשתמש ולמערכת בהצלחה מתקשר כלקוח עם הconnection server .

ClientSide משמש כלקוח שמתקשר עם ה game server, שולח לו את הקלטים של השחקן, ומדפיס את לוח המשחק בעזרת pygame.

Orientation משמש אך ורק לקבלת המידע להדפסת המשחק ללקוחות מgame server .

screen\_manager הוא האחראי על כל המסכים שיכולים להיפתח לשחקן, הוא גם דואג שכאשר השחקן מתנתק מהמשחק הוא חוזר למסך הבית.

קבלת המידע להדפסה מה game server נעשית בעזרת ביצוע פעולת pickle שעושה deserialize לאובייקט Orientation אשר נמצא בשניהם. אובייקט ה Orientation מכיל את כל התכונות של צורה שצריך להדפיס, כך הוא עובר על כל האובייקטים ומדפיס את כל הצורות על המסך.



# קוד הפרויקט

## Database server code