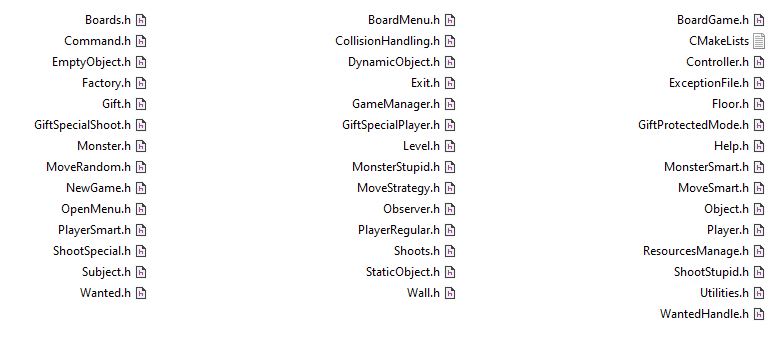
**README-PROJECT**

**שיראל בוסקילה 211668124**

**הסבר כללי של הפרויקט:**

אנחנו נמצאים בעולם של מארוול, כוחות הרשע שולטים, וכך קמו הנוקמים – מטרתם היא למצוא את כל החפצים שכוחות הרשע הסתירו. אבל כוחות הרשע עדיין נמצאים בעולם ולכן הנוקמים חייבים להיזהר. לנוקמים יש כוחות על טבעיים- הם יכולים לעוף לשבור קירות ולירות – גם לכוחות הרשע – הזהרו!!!

**תיכון design :**



**פירוט**

* **Controller**- במחלקה זו נממש את הפונקציה run() שאחראית על ניהול המשחק. המחלקה מכילה משתנים מסוג המחלקות של openMenu ו- GameManager
* **GameManager-** זו נממש את הפונקציה run() המנהלת את שלבי המשחקהיא מבצעת קריאה לבנאי של Level וקריאה לפונקציה run()של Level
* **openMenu**- מחלקה זו מציגה תפריט בסיסי של בחירת משחק חדש או יציאה מהמשחק. המחלקה מכילה משתנה מסוג השתמשנו בתבנית עיצוב command כדי לממש את הכפתורים
* **Level**- במחלקה זו נממש את הפונקציה run() שאחראית על ניהול שלב, המחלקה ממיכלה משתנה מסוג Boardsו וגם וקטור עם המיקומים של המפלצות vector<std::unique\_ptr<Monster>>m\_monsters
* **Boards-** מכיון שיש שני לוחות השתמשנו במחלקה שתכיל את שני הלוחות תפריט משחק ולוח המשחק
* **boardMenu**- מחלקה זו תציג את הנתונים של השחקן בשעת המשחק, כלומר תציג את החפץ שהוא צריך למצוא, את מספר הפסילות שנשארו לו, את מספר החיים שלו, את הזמן עד שהשלב נגמר וכן תציג את מספר השלב הנוכחי.
* **boardGame**- במחלקה זו נטפל בקריאה מהקובץ. המחלקה מכילה משתנה מסוג vector<vector<std::unique\_ptr<staticObject>>m\_gameBoard שזהו לוח המשחק שעליו יהיו האובייקטים הסטטיים.
* **Factory-** מחלקה היוצרת את האוביקטים הסטטים, המפלצות, המתנות והיריות.
* **Resources**- מחלקה זו מטפלת בטעינת התמונות מהקובץ וקבצי שמע ומכילה משתנים מסוג map
* **Utilities**- במחלקה זו ישנם קבועים ו- enum שנשתמש בהם במהלך התכנית.
* **command**- מחלקה זו מממשת תבנית עיצוב והיא מחלקת הבסיס לכפתורים NewGame Help Exit. מחלקה זו מכילה משתנה מסוג . ב- command ישנה פונקציה וירטואלית טהורה excute, ובכל אחת מהמחלקות שיורשות מ- command יש לה מימוש שונה ולכן כאשר יש לחיצת עכבר בתפריט הקומפיילר יודע באיזה לחיצת עכבר לטפל על ידי שימוש בפולימורפיזם.
* **NewGame**- מחלקה זו יורשת ממחלקת command, וכאשר נלחץ על כפתור 'New Game', נתחיל משחק חדש.
* **Exit**- מחלקה זו יורשת ממחלקת command, וכאשר נלחץ על כפתור 'Exit נצא מהמשחק.
* **Help**- מחלקה זו יורשת ממחלקת command, וכאשר נלחץ על כפתור help יופיעו שני מסכי הסבר על המשחק.
* **Object**- מחלקה זו מייצגת אובייקט משחק והיא מחלקת הבסיס למחלקות DynamicObject ו- StaticObject. מחלקה זו מכילה משתנה מסוג sf::Sprite .
* **DynamicObject**- מחלקה זו יורשת ממחלקת Object והיא מייצגת אובייקט דינמי, כלומר אובייקט שזז בלוח. ממחלקה זו יורשים המחלקות player ו- , shoots Monster
* **Player** - מחלקה זו יורשת ממחלקת DynamicObject,והיא אבסטרקטית –

ממנה יורשים שני מחלקות regularPlayer ן- smartPlayer

השחקן הופך לשחקן חכם כשהוא מתנגש במתנה שהופכת אותו לשחקן חכם

המחלקה מייצגת שחקן בלוח, כלומר מחלקה זו מנהלת את צעד השחקן . תכונות השחקן הם מספר חיים, בנוסף לתכונות שיורש ממחלקות הבסיס vector<std::unique\_ptr<shoots>>m\_shoot;- עבור סוגי היריות השונים של השחקן.

* **playerRegular**- יורש ממחלקת הבסיס player וזה השחקן הרגיל שעף יורה ושובר קירות אם הוא מתנגש בששחקן יורדים לו החיים
* **playerSmart**- כביכול כמו השחקן הרגיל אבל אם הוא מתנגש במפלצות הוא אוכל אותם
* **Shoots**- מחלקת אב עבור סוגי היריות השונות ממנה יורשות 2 מחלקות של יריות חכמה וטיפשה
* **Monster**- מחלקה זו יורשת ממחלקת DynamicObject, והיא מייצגת סוג מפלצת. ממחלקה זו יורשים המחלקות RegularMonster ו- SmartMonster, בה יש את המצביע לסוג התזוזה עם strategy ו- ובכל אחת מהמחלקות שיורשות ממנה, יש להם מימוש שונה ולכן כאשר נקרא לפונקציות האלה הקומפיילר יודע איזה פונקציה נקראה על ידי שימוש בפולימורפיזם.
* **RegularMonster**- מחלקה זו יורשת ממחלקת Monster, והיא מייצגת מפלצת רגילה שבה ימומשו הפונקציות שירשה.
* **SmartMonster**- מחלקה זו יורשת ממחלקת Monster, והיא מייצגת מפלצת מיוחדת שבה ימומשו הפונקציות שירשה. יש מימוש ל moveSTRATEGY וממנה יורשות שני מחלקות של תזוזה שונה
* **StaticObject**- מחלקה זו יורשת ממחלקת GameObject והיא מייצגת אובייקט סטטי, כלומר אובייקט שמונח על הלוח.
* **Gift**- מחלקה זו יורשת ממחלקת StaticObject, והיא מייצגת סוג מתנה, כלומר במשחק קיימים כמה סוגי מתנות שהשחקן יכול לאסוף

1. יריה חכמה

2. מצב מוגן מפני יריות

3. הפיכת השחקן לשחקן חכם

* **wantedHandle**- מחלקה שמטפלת באוביקטים שצריך למצוא, היא מציגה אותם גם בתפריט וגם במשך, בנוסף יש שימוש בתבנית עיצוב של Obsrever שמעדכנת אותנו אם החפץ נמצא וכשכל החפצים נמצאים מסתיים השלב –(קוראים את השמות של החפצים מקובץ WANTED

**רשימת קבצים חדשים שנוצרו:**

עבור כל מחלקה נוצרו 2 קבצים, קובץ הh - וקובץ ה-cpp קובץ הh מכיל את התכונות של המחלקה ואת ההצהרות על הפונקציות. ובקובץ הcpp בוצעו המימושים של הפונקציות. בנוסף.יש את קובץ main.cppשבו ניצור משתנה מסוג Controller ועליו נפעיל את פונקצית הrun. יש גם את קובץ utilities.

**.מבני נתונים עיקריים ותפקידיהם:**

* Vector<vector<std::unique\_ptr<StaticObjet>>m\_gameBoard;- עבור האוביקטים הסטטים.
* Vector<std::unique\_ptr<MonsterType>>m\_monsters מחזיק את המפלצות
* vector<std::unique\_ptr<Shoot>>m\_shoot;
* -std::map<std::string, sf::Texture>m\_images; יש כמה פעמים בשביל הסוגים השונים

**תבניות עיצוב שהשתמשתי:**

1. Command
2. Observer
3. Factory
4. multiple dispatch
5. strategy

**באגים ידועים:**

* לא ידוע, אם אתם מגלים תספרו לי😉