**1**.**כותרת**: מבוא לתכנות מונחה עצמים והנדסת תוכנה / פרויקט

**2.פרטי הסטודנטיות:**

* הודיה שכרוב. ת.ז.:211634035
* חני ישראל . ת.ז.: 211429568

**3.הסבר כללי של התרגיל:**

בפרויקט זה בנינו את המשחק Digger הוא משחק מחשב שמטרתו לאסוף יהלומים תוך התחמקות

ממפלצות. השחקן יכול "לאכול" אבנים כדי להגיע אל היהלומים. לאחר ש"אוכל" את כל היהלומים הוא עובר לשלב הבא..

**4.תיכון design :**

בתכניתנו יצרנו 25 מחלקות:

* Controller
* Board
* ButtonType
* Character
* Consts
* Diamond
* Digger
* DinamicObject
* ExitGame
* GameManager
* GameObject
* GiftType
* InformationDisplay
* Level
* Menu
* MonsterType
* NewGame
* RegularMonster
* Resources
* ScoreGift
* SpecialMonster
* StaticObject
* Stone
* StoneGift
* TimeGift
* Wall

**פירוט**

* **Controller**- במחלקה זו נממש את הפונקציה run() שאחראית על ניהול המשחק. המחלקה מכילה משתנים מסוג המחלקות של Menul ו- GameManager
* **GameManager-** זו נממש את הפונקציה run() המנהלת את שלבי המשחקהיא מבצעת קריאה לבנאי של Level וקריאה לפונקציה run()של Level
* **Menu**- מחלקה זו מציגה תפריט בסיסי של בחירת משחק חדש או יציאה מהמשחק. המחלקה מכילה משתנה מסוג Vector<std::unique\_ptr<ButtonType>>m\_menu שבו נשמור את הכפתורים הנ"ל.
* **Level**- במחלקה זו נממש את הפונקציה run() שאחראית על ניהול שלב, המחלקה ממיכלה משתנה מסוג Boardו וגם וקטור עם המיקומים של המפלצות ctor<std::unique\_ptr<MonsterType>>m\_monsters וגם משתנה מסוג InformationDisplay
* **InformationDisplay**- מחלקה זו תציג את הנתונים של השחקן בשעת המשחק, כלומר תציג את מספר הנקודות שצבר, את מספר הפסילות שנשארו לו, את מספר האבנים שמותר לו לאכול ואת הזמן שנותר לו, וכן תציג את מספר השלב הנוכחי.
* **Board**- במחלקה זו נטפל בקריאה מהקובץ. המחלקה מכילה משתנה מסוג <vector<std::unique\_ptr<staticObject>>m\_gameBoard שזהו לוח המשחק שעליו יהיו האובייקטים הסטטיים.
* **Resources**- מחלקה זו מטפלת בטעינת התמונות מהקובץ וקבצעי שמעת ומכילה משתנה מסוג .Vector<Texture sf::font m\_font,st::music m\_music
* **Const**- במחלקה זו ישנם קבועים ו- enum שנשתמש בהם במהלך התכנית.
* **ButtonType**- מחלקה זו מייצגת כפתור והיא מחלקת הבסיס לכפתורים NewGame ו- ExitGame. מחלקה זו מכילה משתנה מסוג .texture ב- Button ישנה פונקציה וירטואלית טהורה handleClick, ובכל אחת מהמחלקות שיורשות מ- Button 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 Button, וכאשר נלחץ על כפתור 'New Game', נתחיל משחק חדש.
* **ExitGame**- מחלקה זו יורשת ממחלקת Button, וכאשר נלחץ על כפתור 'Exit Game' נצא מהמשחק.
* **GameObject**- מחלקה זו מייצגת אובייקט משחק והיא מחלקת הבסיס למחלקות DynamicObject ו- StaticObject. מחלקה זו מכילה משתנה מסוג sf::Sprite .
* **DynamicObject**- מחלקה זו יורשת ממחלקת GameObject והיא מייצגת אובייקט דינמי, כלומר אובייקט שזז בלוח. ממחלקה זו יורשים המחלקות Digger ו- , MonsterType
* **Digger** - מחלקה זו יורשת ממחלקת DynamicObject, והיא מייצגת שחקן בלוח, כלומר מחלקה זו מנהלת את צעד השחקן . תכונות השחקן הם מספר חיים, ניקוד ומספר אבנים שמותרות לאכילה בנוסף לתכונות שיורש ממחלקות הבסיס.
* **MonsterType**- מחלקה זו יורשת ממחלקת DynamicObject, והיא מייצגת סוג מפלצת. ממחלקה זו יורשים המחלקות RegularMonster ו- SpecialMonster, בה יש את הפונקציה הווירטואליות טהורות move() 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 MonsterType, והיא מייצגת מפלצת רגילה שבה ימומשו הפונקציות שירשה.
* **SpecialMonster**- מחלקה זו יורשת ממחלקת MonsterType, והיא מייצגת מפלצת מיוחדת שבה ימומשו הפונקציות שירשה.
* **StaticObject**- מחלקה זו יורשת ממחלקת GameObject והיא מייצגת אובייקט סטטי, כלומר אובייקט שמונח על הלוח. ממחלקה זו יורשים המחלקות Diamond, Stone, Wall ו- , GiftType ולכל אחת מהמחלקות שיורשות ממחלקה זו יהיה מימוש שונה לפונקציה הווירטואלית collide () ולכן כאשר נקרא לפונקציות האלה הקומפיילר ידע איזה פונקציה נקראה על ידי שימוש בפולימורפיזם.
* **GiftType**- מחלקה זו יורשת ממחלקת StaticObject, והיא מייצגת סוג מתנה, כלומר במשחק קיימים כמה סוגי מתנות שהשחקן יכול לאסוף, כגון: תוספת למספר האבנים שמותר "לאכול", הגדלת הזמן המוקצב, הוספת ניקוד. כל מתנה תוצג במחלקה נפרדת.

**5.רשימת קבצים חדשים שנוצרו:**

יצרנו 25 מחלקות, ועבור 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 consts.

**6.פורמט קובץ שלב:**

ניתן להוסיף שלב בקובץ ניפרד הקובץ צריך להיות בשם Leveli.txt ש i- הוא מספר השלב.

**7.מבני נתונים עיקריים ותפקידיהם:**

* Vector<vector<std::unique\_ptr<staticObject>>m\_gameBoardמחזיק את האוביקטים הסטטים
* Vector<std::unique\_ptr<MonsterType>>m\_monsters מחזיק את המפלצות
* Vector<std::unique\_ptr<ButtonType>>m\_menu מחזיק את הכפתורים
* Enum

**.8אלגוריתמים הראויים לציון:**

* אלגוריתם מנהטן בתזוזה של המפלצות החכמות

**9.באגים ידועים:**

* תעופת תכנית בגלל גישה למקום שלא קיים בזיכרון.
* תעופת תכנית בגלל דברים שהיו חסרים ב CMacke.
* בעיות בקריאה מקובץ.
* בעיות בחריגה מגבולות הלוח בתזוזת השחקן והמפלצות.

**10.** **הערות אחרות:**

* גודל הלוח הוא מספרים המתחלקים ב1200 ו-800.