OOP1-פרויקט

מגישים: אוראל אביעד 305575003 וכרם ג'אבר 207251141

**הסבר כללי של התרגיל:**

בתרגיל זה התבקשנו לבנות גרסה למשחק דיגר, הדיגר צריך לאכול יהלומים, ולהיזהר לא להיאכל ע"י מפלצות, ומותר לו לאכול אבנים בכמות מוגבלת שניתנת לו בתחילת כל שלב. אם הדיגר נאכל ע"י מפלצת אנו מאתחלים את הדיגר והמפלצות למיקום ההתחלתי שלהם באותו שלב.

גם המשחק רץ לפי זמן, לכל שלב יש זמן אחר , אם הזמן נגמר אנחנו מאתחלים את השלב מחדש ומורידם לדיג את החיים.

גם הדיגר יוכל לאכול מתנות וזה יוסיף לו חיים, זמן, אבנים שמותר לאכול.

אם הדיגר אוכל יותר אבנים ממה שמותר לו השלב מאותחל מחדש ומחזיר את הדיגר, המפלצות והיהלומים לתחילת השלב.

כל עוד נשאר חיים לשחקן הוא ממשיך ומתקדם בשלבים שהוכנסו לו בקובץ ""Board.txt

**תכנון (Design):**

בנינו משחק שמנוהל ע"י , Controller הוא משתמש בזיכרון של ה BORAD

ובנינו מחלקת Board, שעוזרות לקונטרולר לנהל את המשחק בצורת תכנות "מונחה עצמים" ע"י חלוקה לאובייקטים, ולכל אובייקט יש את הפונקצייות שלו.

גם יש לנו מחלקת ResourceManager שהיא פועלת בסגנון Singleton שהיא מאותחלת רק פעם אחת וקוראת את כל הקבצים שצריכים למשחק כמו תמונות,FONT,מוזיקה,SPRITESHEET.

ויש לנו מחלקות לכל אובייקט במשחק כמו מפלצות, דיגר ,מתנות, קיר, אבנים

גם השתמשנו בשיטת ה Double-Dispatch כדי לבדוק התנגשויות בין האובייקטים שיש במשחק.

**פורמט קובץ השלב:**

אנחנו קוראים את השלבים מתוך קובץ שיש ב resources שם אנחנו מציירם את השלבים שאנחנו

רוצים למשחק, אז אם רוצים להוסיף שלב פשוט הולכים לקובץ הזה ומציירים את השלב שבא לנו.

**רשימת קבצים הנוצרו על ידינו:**

Controller.h/cpp – המחלקה שמנהלת את המשחק: היא משתמשת בזיכרון של ה BOARD

מנהלת את השלבים ואת התור של השחקן הוא המפלצות ומעדכנת את המסך בהתאם.

Board.h/cpp – המחלקה שתפקידה לבנות ולהחזיק את לוח המשחק.

מעדכנת את מיקום ה Digger ואת מיקומי הMonsters שהקונטרולר יוכל לעדכן אותם אצלו.

בנוסף 1)מאתחלת שלב חדש 2)משחזרת את השלב מחדש

GraphicObject.h/cpp-

זה מחלקת הבסיס שלנו שרוב המחלקות יורשות ממנה את הפונקציות הבסיסיות כמו draw,setpos,setsize..

GameObject.h/cpp

זה המחלקה של האובייקטים שיש פה הפונקציות של ההתנגשויות בין האובייקטים, היא יורשת מ GraphicObject

MovingObject.h/cpp

זה המחלקה של האובייקטים שזזים כמו מפלצות דיגר שהיא מטפלת בכל התזוזה שלהם וכל ההתנגשויות, והשעון שלהם.

Digger.h/cpp – המחלקה שמייצגת את השחקן.

מכילה פונקציות set get לקבל מיקום של השחקן ולאתחל את המיקום שלו.

בנוסף פונקציית move שבודקת על איזה מקש השחקן לחץ והאם ניתן לזוז לשם.

במידה שכן מזיזה את השחקן למיקום החדש.

Monster.h/cpp – המחלקה שמייצגת מפלצת.

מכילה פונקציות מיקום. ופונקציית move שמגרילה מספר ולפיו מחליטה לאן לזוז במידה והיא אכן יכולה לזוז לשם.

StupidMonster.h/cpp

מחלקה של המפלצת הלא חכמה שהיא יורשת מ MONSTER

SmartMonster.h/cpp

מחלקה של המפלצת החכמה שהיא יורשת מ MONSTER

StaticObject.h/cpp

המחלקה של האובייקטים הסטטים(שלא זזים) כמו אבן קיר יהלום

Diamond.h/cpp

מחלקה של יהלום שהיא יורשת מ StaticObject

Wall.h/cpp

מחלקה של קיר שהיא יורשת מ StaticObject

Present.h/cpp

מחלקה של מתנה שהיא יורשת מ StaticObject

Menu.h/cpp

מחלקה של התפריט של התחלת המשחק

Toolbar.h/cpp

מחלקה של סרגל המשחק שמכיל את החיים זמן מספר אבנים מספר שלב

Text\_object.h/cpp

מחלקה לטעינת ה font וכל דבר שקשור לכתב(TEXT)

Stone.h/cpp

מחלקה של אבנים שהיא יורשת מ StaticObject

**מבני נתונים עיקריים ותפקידם:**

vector of levels in the game control

vector of monsters in the board

vector of unmovable objects in the board

**אלגוריתמים הראויים לציון:**

Each smart monster when is its turn to move, calculates the distance from every available spot that.

it could move to, to the location where the digger is, and then chooses the place where the distance is the minimal.

Note: the monsters will only stay in their place if they have no possible way to move, we first designed

the algorithm to let the monster stay in its place if that was a closer spot to the digger but it seemed

boring that the monsters wouldnt move sometimes**.**

**באגים ידועים:**

אין.

**הערות אחרות:**

לפעמים שמגדילים את המסך גודל האובייקטים לא גודל יש בעיה ב Spritesheet באנמציה של הדיגר

**טעיות שתקנו אחרי ההצגה:**

תיקנו את המהירות של המפלצות

עשינו לכל מתנה מחלקה

עשינו לכפתורים גם מחלקה

שנינו קצת ב Members