**README- project Dave-Game**

הסבר כללי על המשחק- dave:

לוח המשחק בנוי מפלטפורמות, עליהם רשאי השחקן לדרוך. על הלוח מסתובב השחקן אוסף יהלומים ומתנות ומתחמק מאש וממפלצות. מטרת השחקן היא לקחת את המפתח הנמצא על הלוח ועמו לפתוח את הדלת לשלב הבא תוך צבירת נקודות בכל שלב.    
השחקן מוגבל במספר חיים, כלומר בכל פעם שנכשל יורד לו חיים. במשחק זה ישנם 3 שלבים.

Design:

**מחלקות האובייקטים במשחק:**

Object**:** מחלקת בסיס המייצגת את כל האובייקטים הקיימים במשחק.

DynamicObject: מחלקה היורשת מ- ,Object היא לאובייקטים שיש להם תזוזה כמו: מפלצת ושחקן.

StaticObject: מחלקה היורשת מ- ,Object היא לאובייקטים הסטטיים במשחק כמו: יהלומים, מתנות, אש וכו'...

**מחלקות של Design Patterns**:

תבנית העיצוב singleton

אצלנו זה מחלקת grafix. אחראית לטפל בגרפיקה של המשחק ובטעינת התמונות של האובייקטים.

תבנית העיצוב State:

State: מחלקת הבסיס לכל המחלקות המייצגות מצב שבו השחקן יכול להיות, כמו קפיצה, הליכה, נפילה וכו'. יורשות ממנה המחלקות:

* Walk State: מצב הליכה, זהו מצב ברירית המחדל עבור השחקן.
* Jumping State: מצב קפיצה, אליו יכול להגיע ממצב הליכה ע"י לחיצה על החץ העליון.
* Fall State: מצב נפילה לאחר הקפיצה של השחקן.
* Standing State: מצב בו השחקן עוצר ולא זז לשום כיוון.

תבנית העיצוב Factory:

מחלקת Factory: מחזיקה פונקציות סטטיות שתפקידן ליצור את האובייקטים של המשחק. השתמשנו בה ליצירת האובייקטים הסטטיים במשחק.

תבנית העיצוב MultyMethods:

מחלקת Collision: מחלקה המטפלת בכל ההתנגשויות במשחק ע"י הגדרת פונקציות לכל האובייקטים המתנגשים עם אובייקטים אחרים במשחק. הטיפול בהתנגשות מתבצע במחלקה עצמה.

**מחלקות אחרות הקשורות להצגת המשחק וריצתו:**

Board: מחלקה האחראית ליצירת האובייקטים הסטטיים בכל מהלך המשחק. כמו כן היא אחראית להצגתם על הלוח.

Level: מחלקה האחראית לטיפול בשלבים וכן אחראית לתזוזתם וניהול ההתנגשויות שלהם. כלומר קריאה לפונקציות המתאימות במחלקות המתאימות, והטיפול עצמו לא נעשה ב Level.

Menu: זוהי מחלקה המייצגת ניהול של הוצאת פקודות לפועל כמו: התחלת המשחק, סיום ועזרה. היא מכילה ווקטור של .Buttons

Controller: קורא למחלקת תפריט ששם נעשה טיפול בלחיצה על כפתורים בתפריט המוצג בתחילת המשחק.

רשימה של הקבצים שנוצרו:

קבציh :

בקבצים אלו נמצאים ההצהרות על הפונקציות והבנאים שבכל מחלקה וכן השדות בכל מחלקה.

"Level.h"- ניהול שלב יחיד של המשחק  
"Dave. h "- מחלקת השחקן – דייב. מחזיקה נתונים על השחקן וכן פונקציית תזוזת השחקן  
"Monster. h " מחלקת מפלצת. מחזיקה פונקציית תזוזה של המפלצת.

Diamond. h "" מחלקת יהלום

Board. h ""- מחלקת לוח אחראית על עריכת הלוח וציור האובייקטים עליו

"Factory. h "- מחלקת ייצור אובייקטים

"StandingState.h "- מחלקה יורשת מstate המייצגת מצב עמידה של שחקן

"Collision.h "- מחלקת טיפול בהתנגשויות במשחק

"Help.h "- מחלקת כפתור עזרה

"Fire.h "- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת האש בfactory.

"Game.h "- ניהול כל השלבים במשחק יחד.

"JumpingState.h " מחלקה יורשת מstate המטפלת במצב קפיצה של שחקן

"Wall.h "- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת הקיר בfactory.

"WalkState.h " מחלקה יורשת מstate המטפלת במצב הליכה ימינה או שמאלה של השחקן.

"Gifts. h "- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת המתנות בfactory.

"FallState. "- מחלקה יורשת מstate המטפלת במצב נפילה כתוצאה מקפיצה של השחקן.

"Controller.h "-מחלקה שאחראית להפעיל את menu

"State.cpp"- מחלקת בסיס לכל מצבי התזוזה של השחקן

"InformationDisplay.cpp"- מחלקת הצגת מידע על המשחק כמו: כמות הניקוד, חיים וכו'

"Key.cpp" - מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת המפתח בfactory.

"Door. h " - מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת הדלת בfactory.  
"MyExeption. h "- מחלקת חריגות כללית.

"loudingFillExeption.h"-  חריגה המייצגת שגיאה בעת הורדת קובץ.  
"Menu. h "- ניהול מסך תפריט  
"Button. h " -מחלקת כפתור  
"DynamicObject. h "- מחלקה של אובייקטים דינאמיים.  
"StaticObject. h "- מחלקת אובייקט סטאטי.  
"Object h " -מחלקת בסיס של אובייקט. מחזיק את המיקום של האובייקט, את התמונה הנוכחית שלו, גודל התמונה ופונקציות שונות.  
"ViewManage. h " -ניהול הview  
"Grafix. h " -מחלקה המייצגת גרפיקה.

"Const.h"- מחלקת הגדרת קבועים בתוכנית.

קבצי :cpp

בקבצים אלו נמצאים המימושים של הפונקציות שבכל מחלקה ואתחול הבנאים.

"Level.cpp" -ניהול שלב יחיד של המשחק  
"Dave.cpp" - מחלקת השחקן – דייב. מחזיקה נתונים על השחקן וכן פונקציית תזוזת השחקן.  
"Monster.cpp" -מחלקת מפלצת. מחזיקה פונקציית תזוזה של המפלצת.

Diamond.cpp"" - מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת היהלום בfactory.

Board.cpp""- מחלקת לוח אחראית על עריכת הלוח וציור האובייקטים עליו

"Factory.cpp"- מחלקת ייצור אובייקטים

"StandingState.cpp"- מחלקה יורשת מstate המייצגת מצב עמידה של שחקן

"Collision.cpp"- מחלקת טיפול בהתנגשויות במשחק

"Help.cpp"- מחלקת כפתור עזרה

"Fire.cpp"- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת האש בfactory.

"Game.cpp"- ניהול כל השלבים במשחק יחד.

"JumpingState.cpp" מחלקה יורשת מstate המטפלת במצב קפיצה של שחקן

"Wall.cpp"- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת הקיר בfactory.

"WalkState.cpp" מחלקה יורשת מstate המטפלת במצב הליכה ימינה או שמאלה של השחקן.

"Gifts.cpp"- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת המתנות בfactory.

"FallState.cpp"- מחלקה יורשת מstate המטפלת במצב נפילה כתוצאה מקפיצה של השחקן.

"Controller.cpp"-מחלקה שאחראית להפעיל את menu

"State.cpp"- מחלקת בסיס לכל מצבי התזוזה של השחקן

"InformationDisplay.cpp"- מחלקת הצגת מידע על המשחק כמו: כמות הניקוד, חיים וכו'

"Key.cpp"- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת המפתח בfactory.

"Door.cpp"- מכילה בנאי המקבל תכונות של staticObject וכן שדה בוליאני registerit בשביל יצירת הדלת בfactory.  
"MyExeption.cpp" מחלקת חריגות כללית.

"loudingFillExeption.cpp" - חריגה המייצגת שגיאה בעת הורדת קובץ.  
"Menu.cpp"- ניהול מסך תפריט  
"Button.cpp" -מחלקת כפתור  
"DynamicObject.cpp"-  מחלקה של אובייקטים דינאמיים.  
"StaticObject.cpp"-  מחלקת אובייקט סטאטי.  
"Object.cpp" - מחלקת בסיס של אובייקט. מחזיק את המיקום של האובייקט, את התמונה הנוכחית שלו, גודל התמונה ופונקציות שונות  
"ViewManage.cpp" -ניהול הview  
"Grafix.cpp" -מחלקה המייצגת גרפיקה.

Main.cpp- אחראי על ריצת המשחק.

מבני נתונים חשובים ותפקידיהם:

במחלקת Factory:

typedef std::map<char, std::unique\_ptr<StaticObject>(\*)(const sf::Texture&,sf::Vector2f, const sf::Vector2i,int )> mymap;. משמש ליצירת האובייקטים הסטטיים.

במחלקת grafix:

std::map<std::string, std::vector<sf::Texture>>, להחזקת וקטורים של textures עבור כל האובייקטים.

במחלקת Collision:

using HitMap = std::map<Kay, HitFunctionPtr>;

using **HitFunctionPtr** = void(\*)(Object&, Object&, Level& l);

using **Key** = std::pair<std::type\_index, std::type\_index>;

תפקידה הינו לקבל שני שמות שמות של אובייקטים ולהפנות אותם לפונקציה המתאימה לטיפול בהתנגשות של שניהם.

: Levelבמחלקת

המפלצות. להחזקת std::vector< Monster>m\_monster

: Menu במחלקת

להחזקת כפתורי התפריט. std::vector<Button>m\_buttonOfMenu;

: Board במחלקת

vector <vector<std::unique\_ptr<StaticObject>>> m\_editBoard;

אלגוריתמים הראויים לציון:

הקפיצה של השחקן חושבה לפי שעון. מוקצב לשחקן זמן לעמוד באוויר ואז נופל

באגים ידועים:

הערות אחרות: