**README – Project**

**שם הקורס**: מבוא לתכנות מונחה עצמים והנדסת תוכנה.

**שמות הסטודנטים**:

* אליעד קרני | ת"ז 206329104 | login: eliadka
* שרון לוי | ת"ז 311593313 | login: sharonlevi

**הסבר כללי:**

בפרויקט זה, אנו ממשים את המשחק Lode Runner בשימוש הספרייה הגרפית SFML בשפת C++.

המשחק כולל 4 שלבים הנמצאים בקובץ “Levels.txt” הנמצא בתיקיית resources.

**תיכון :**

המחלקה GameObject היא מחלקת הבסיס של האובייקטים במשחק וממנה יורשות כל מחלקות האובייקטים במשחק- אובייקטים דינאמיים וסטטיים.

מחלקת Controller מנהלת את מהלך המשחק ומעבר בין שלב לשלב, מתקשרת עם כלל המחלקות וכך מנהלת את התורות והתנגשויות בין האובייקטים.

המחלקה GameState מנהלת את מצב המשחק ומשנה את כל הפרמטרים בהתאם להתקדמות ולאירועים שקורים במהלך המשחק.

מחלקת Board מחזיקה את כל האובייקטים של המשחק במצבם ההתחלתי, מספקת נתונים אודות אובייקטים אלה במידת הצורך (בתורות או התנגשויות).

האובייקטים הסטטיים מתחלקים לשלוש מחלקות יורשות בצורה ישירה- סולם,מוט ואובייקטים שניתן לאסוף אותם-מתנה ומטבע.

למחלקת Enemy יש שלוש מחלקות יורשות – מחלקה לכל סוג אויב.

למחלקת Gift יש 4 מחלקות יורשות – מחלקה כל סוג מתנה.

מחלקת Resources מחזיקה באמצעות מפות, את כל המשאבים של המשחק -טקסטורות,מוזיקה וסאונדים.

**רשימת קבצים :**

* Controller.cpp/ Controller.h – מחלקה שתפקידה להריץ את המשחק ולנהל אותו. בין היתר מזמנת שיטות של הלוח, מדפיסה את מצב המשחק, טוענת שלבים חדשים ועוד.
* Board.cpp\ Board.h – מחלקה המחזיקה בכל זמן נתון את השלב הנוכחי בו השחקן נמצא, טוענת שלב חדש מהקובץ בעת הצורך. מחזיקה מידע אודות האובייקטים במשחק. בין היתר בודקת האם תזוזה כלשהי היא אפשרית.
* DataReader.cpp\ DataReader.h – מחלקה שתפקידה לטעון שלב חדש מקובץ השלבים, כוללת את הנתיב לקובץ, פותחת אותו וטוענת כל פעם את השלב החדש.
* GameState.cpp \GameState.h – מחלקה המחזיקה מידע אודות מצב השחקן: כמה חיים יש לו, ניקוד, הזמן שנותר לשחק ומספר השלב בו הוא נמצא.
* GameObject.cpp\GameObject.h – מחלקה אבסטרקטית, מחלקת האם של כל האובייקטים במשחק.
* MovingObject.cpp\MovingObject.h – מחלקה אבסטרקטית. מחלקת האם של כל האובייקטים הנעים במשחק.
* StaticObject.cpp\StaticObject.h – מחלקה אבסטרקטית, מחלקת האם של האובייקטים הסטטיים במשחק.
* CollectableObject.cpp\CollectableObject.h – מחלקה אבסטרקטית, מחלקת הבסיס של כל האובייקטים שניתן לאסוף אותם במשחק (מטבע\מתנה)
* Player.cpp\ Player.h – מחלקת שחקן, מנהלת את התזוזות של השחקן, האנימציות והתנגשויות שלו עם האובייקטים הסטטיים במשחק. בנוסף גם מבצעת את פעולת החפירה.
* Enemy.cpp\ Enemy.h – מחלקת הבסיס של האויבים במשחק, כל שלושת סוגי האויבים יורשים ממחלקה זו.
* Coin.cpp\ Coin.h –מחלקה המייצגת את המטבע. יש לו ממבר סטטי שמנהל את מספר המטבעות על הלוח ומשנה אותו כשהשחקן אוסף מטבעות\השלב משתנה\מתחיל מחדש. כמו כן, הוא מנהל התנגשויות עם השחקן.
* Gift.cpp\ Gift.h –מחלקת הבסיס של המתנות, כל מחלקה המייצגת סוג מתנה יורשת ממחלקה זו.
* Ladder.cpp\ Ladder.h –מחלקה המייצגת את האובייקט הסטטי סולם.
* Wall.cpp\ Wall.h – מחלקה המייצגת את האובייקט הסטטי קיר\רצפה.
* Rod.cpp\ Rod.h – מחלקה המייצגת את האובייקט הסטטי מוט.
* TimeGift.cpp \TimeGift.h –מחלקה המייצגת את המתנה שמוסיפה בונוס של זמן.
* ScoreGift.cpp\ ScoreGift.h –מחלקה המייצגת את המתנה שמוסיפה ניקוד לשחקן.
* LifeGift.cpp\ LifeGift.h –מחלקה המייצגת את המתנה שמוסיפה עוד חיים לשחקן.
* BadGift.cpp\ BadGift.h –מחלקה המייצגת מתנה "רעה" שמוסיפה עוד אויב למשחק.
* SmartEnemy.cpp\ SmartEnemy.h – מחלקה המייצגת אויב חכם, יורשת ממחלקת אויב, נע לעבר השחקן.
* FoolEnemy.cpp\ FoolEnemy.h – מחלקה המייצגת אויב טיפש, יורשת ממחלקת אויב, נע רק ימינה שמאלה.
* RandEnemy.cpp\ RandEnemy.h – מחלקה המייצגת אויב אקראי, יורשת ממחלקת אויב, נע בתנועה רנדומלית בלוח.
* Resources.h\Resources.cpp – מחלקת סינגלטון בדומה לנלמד בהרצאה. מחזיקה את כל המשאבים של המשחק- טקטורות, סאונדים ומוזיקה בתוך מבנה נתונים של מפה. מכילה פונקציה סטטית שאותה ניתן לזמן מכל מקום בתוכנית ולקבל את המשאב הרצוי.
* Door.h \ Door.cpp – מחלקה המייצגת דלת הנמצאת בכל שלב וממנה יוצאים האויבים שנוצרים ע"י המתנות הרעות.
* Utilities.cpp\ Utilities.h – פונקציות עזר לשימוש כללי.
* Macros.h – קובץ המכיל את כלל הקבועים בפרויקט.

**מבני נתונים עיקריים:**

* וקטור דו ממדי.
* מפה.

**אלגוריתמים הראויים לציון:**

תזוזת האויב – במחלקת SmartEnemy השיטה PlayTurn מחשבת את הצעד הבא של האויב עפ"י האלגוריתם הבא:

1. השיטה מקבלת את מפת השלב וזמן התזוזה.
2. האם האויב והשחקן נמצאים על עמודה שונה במפה?
3. אם כן – האם השחקן נמצא מימינו של האויב? אם כן תזיז את האויב ימינה, אחרת תזיז אותו שמאלה.
4. האם המיקום החדש תואם את המיקום הקודם?
5. אם כן- האם השחקן נמצא מעל האויב?
6. אם כן- תזיז את האויב צעד למעלה.
7. אחרת-תזיז את האויב צעד למטה.
8. תעדכן את מיקומו החדש של האויב במפה עפ"י המיקום החדש שהתקבל.

**באגים ידועים:**

The enum type 'sf::PrimitiveType' is unscoped. Prefer 'enum class' over 'enum' (Enum.3).

(שורה 32 בController.cpp). – באג של SFML

**הערות אחרות:**

הקובץ Levels.txt המכיל את השלבים נמצא בתיקייה resources.