**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 :**

**רשימת קבצים :**

* Controller.cpp/ Controller.h – מחלקה שתפקידה להריץ את המשחק ולנהל אותו. בין היתר מזמנת שיטות של המפה, מדפיסה את מצב המשחק, טוענת שלבים חדשים ועוד.
* Board.cpp\ Board.h – מחלקה המחזיקה בכל זמן נתון את השלב הנוכחי בו השחקן נמצא. מחזיקה מידע אודות מיקומיהם ההתחלתיים של השחקן, האויבים והמטבעות ובודקת האם התזוזות של האובייקטים הדינאמיים אפשריות.
* Player.cpp\ Player.h – מחלקת שחקן, שומרת את מיקומו העדכני של השחקן במשחק, ומשחקת את תורו של השחקן.
* Enemy.cpp\ Enemy.h – מחלקת אויב, מחזיקה את מיקומו העדכני של אותו אויב ומשחקת את תורו של האויב.
* 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 ומספר השלב בו הוא נמצא.
* Utilities.cpp\ Utilities.h – פונקציות עזר לשימוש כללי.

**מבני נתונים עיקריים:**

* וקטור דו ממדי.
* מפה.

**אלגוריתמים הראויים לציון:**

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

1. השיטה מקבלת את מפת השלב ואת מיקומו הנוכחי של השחקן.
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).

**הערות אחרות:**