בס"ד

**README**

**כותרת :**

תכנות מונחה עצמים ופיתוח משחקים – פרוייקט משחק xonix

**פרטי הסטודנטיות:**

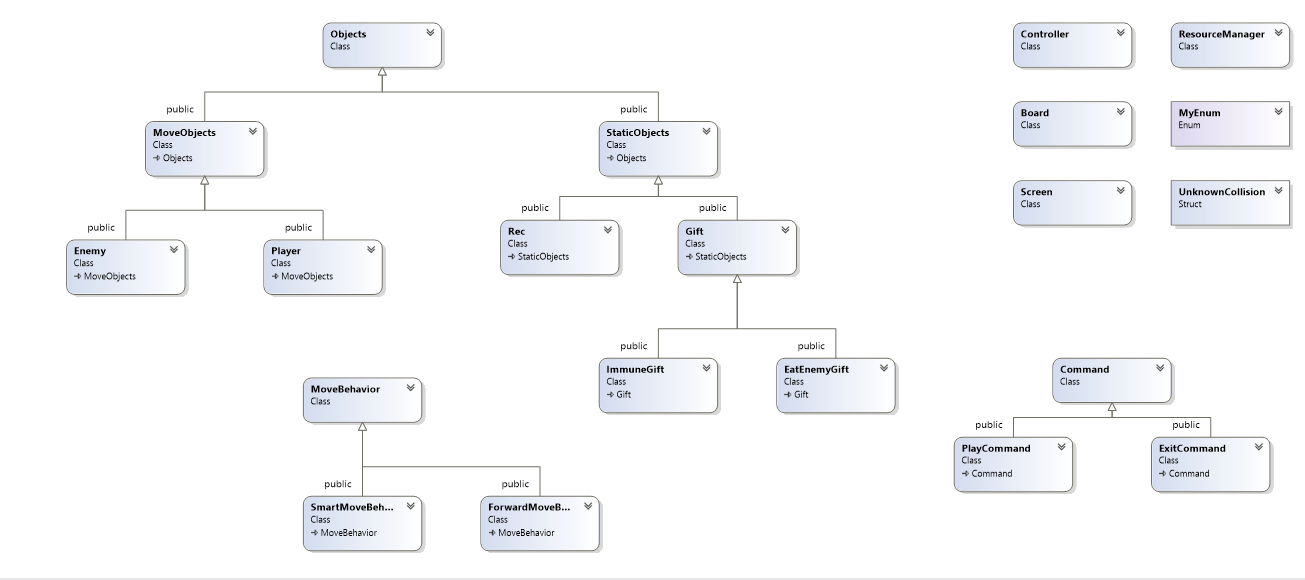
אפרת כהן 206733966

**הסבר כללי של התרגיל:**

משחק Xonix שבו על השחקן לתחום שטחים, כאשר עליו להיזהר מהתנגשות באויבים.

במהלך המשחק מופיעות מתנות אותן הוא יכול לאכול והן נותנות לו יכולות-על. השלב מסתיים כאשר השחקן תחם את כל הלוח.

**תיכון:**

****

**Objects** - מחלקה המייצגת כל אובייקט במשחק, ובעיקר החזקת .sprite

יורשות ממנה שתי מחלקות: StaticObjects ו- MoveObjects .

**MoveObjects** - מייצגת את האובייקטים הזזים שהם: השחקן והאויבים. זו מחלקה אבסטרקטית.

**StaticObjects** - מייצגת את האובייקטים הפסיביים שהם המטבעות והמתנות וכן המשבצות שמיוצר הלוח. (אלו אובייקטים שמודפסים על הלוח ונשמרים בווקטור בBoard )

מתוך מחלקת MoveObjects יורשים שתי מחלקות: Player ו-,Enemy אשר יש להן פונקציונאליות שונה בתזוזה וביכולות של כל אחד.

**Player**– מייצרת את השחקן (הכובש) ותזוזתו.

**Enemy**– מייצרת את האוייבים, ישנם 2 סוגים של אוייבים: רנדומלי וחכם מיוצרים ע"י תבנית strategy.

מתוך מחלקת StaticObjects יורשות שתי מחלקות: Gift ו-. Rec

**Rec**– מייצגת כל משבצת במטריצה שיש לה מצב (שטח אויב, שטח כבוש ומעבר)

**Gift**– מייצגת את המתנות שהכובש לוקח.

ממחלקת Gift יורשות שתי מחלקות: ImmuneGift ו- EatEnemyGift

**ImmuneGift**–הכובש מוגן מהאויבים פעם אחת.

**EatEnemyGift** **–** הכובש רשאי לאכול אויב אחד.

**Board** – מייצג את כל האובייקטים הסטטיים (המתנות והמטבעות) על ידי ווקטור באופן פולימורפי, ואת משבצות הלוח מייצגת מטריצה אשר לכל משבצת קיים מצב שמתעדכן במהלך המשחק וכן גם לאויבים יש ווקטור גם באופן פולימורפי.

**ResourceManager** – זוהי מחלקה המתנהלת בעזרת .singletonהטעינות של התמונות, המוזיקה וטקסט מתבצעות פעם אחת בלבד.

**Controller** - מנהלת את הלולאה העיקרית של המשחק, מחזיקה את האובייקט הכובש ,הלוח והעיצוב.

**Command**- אחראית על התפריט ומתחלקת לשתי מחלקות שיורשות מימנה:

ExiCommand, PlayCommand

**ExiCommand -** יציאה מהמשחק לאחר הקשה על exit.

- **PlayCommand** התחלת המשחק.

**MoveBehavior-** האסטרטגיה של תזוזת האויבים, הרנדומלי והחכם. ממנו יורשות המחלקות SmartMoveBehavior ו- ForwardMoveBehavior

**SmartMoveBehavior**- האויב החכם זז ע"פ תזוזת הכובש. כלומר-לפי תזוזת הכובש הוא בודק באיזה כיוון הכובש לעומתו וזז בהתאם (לפי ציר ה-x וציר הy).

**ForwardMoveBehavior**- האוייב הרנדומלי מגריל כיוון רנדומלי.

**CollisionHandling**- אחראית על ההתנגשויות בין האובייקטים בשיטת multi method.

**Screen**- אחראי על המסך פתיחה , רקע , סיום, תפריט והצגת החיים והשלבים.

**Colide**- אחראי על בדיקה בין שני אובייקטים אם הייתה התנגשות.

**קבצים:**

main.cpp

מחלקה ראשית היוצרת מופע של Menu ומתחילה להריצו.

Controller.h Controller.cpp

ניהול הלולאה הראשית של המשחק.

Board.h Board.cpp

ייצוג הלוח, האובייקטים הסטטיים, האויבים.

Object.h Object.cpp

מחלקת הבסיסית, אשר מתארת את כל האוביקטים על הלוח ע"י השדה sprite שלו תמונה ומיקום.

MoveObject.h MoveObject.cpp

אחראית על כל האוביקטים הדינמיים במשחק, זוהי מחלקה אבסטרקטית.

StaticObject.h StaticObject.cpp

מחלקה אבסטרקטית המייצגת את האובייקטים הסטטיים.

Player.h Player.cpp

אחראית על השחקן (הכובש). מכילה שדה של חיים

Enemy.h Enemy.cpp

אחראית על האויבים.

SmartMoveBehavior.h SmartMoveBehavior.cpp

אחראית על האויבים החכמים, מיישמת את פונקציית move באופן שהאויבים ירדפו אחרי הכובש.

ForwardMoveBehavior. cpp ForwardMoveBehavior.h

אחראית על האויבים הרנדומלים, מיישמת את פונקצי moveבאופן שהאויבים יזוזו בצורה אקראית.

Gift.h Gift.cpp

מייצגת את המתנות אותן לוקח הכובש. ממחלקה זו יורשות שתי מחלקות: ImmuneGift ו- EatEnemyGift

ImmuneGift.h ImmuneGift.cpp

אחראית להגן על הכובש מאויב פעם אחת.

IncreaseSpeed. cpp IncreaseSpeed.h

אחראית לאפשר לכובש לאכול אויב פעם אחת.

ResourceManager.h ResourceManager.cpp

אחראית על טעינת כל התמונות, המוזיקה ,וטקסט. מיושמת בעזרת .singleton

Command.h Command.cpp

אחראית על כפתורי התפריט, ממומשת בעזרת פולימורפיזם.

ExitCommand.h ExitCommand.cpp

כפתור יציאה מהמשחק.

PlayCommand.h PlayCommand.cpp

כפתור התחלת משחק.

Colide.h Colide.cpp

בדיקה בין שני אובייקטים אם הייתה התנגשות.

**מבני נתונים עיקריים ותפקידיהם:**

- **ווקטור** מסוג StaticObjects מייצג את המתנות והמטבעות. שומר את כל האובייקטים הסטטיים וקורא להם באופן פולימורפיזי. ווקטור מסוג אויבים, ומטריצה (ווקטור על ווקטור) של הלוח.

-**ווקטור** מסוג 2f שמכיל את המיקומים שעבר הכובש שכשמגיע לשטח שלו הווקטור הופך לשטח של הכובש.

- **MultiMethod**-אחראית על התנגשות בין האובייקטים.

- **תבנית singlton**: תבנית זו נועדה להגביל את יצירת המופעים של מחלקה  מסוימת למופע יחיד, שימושית כאשר נדרש רק מופע אחד על מנת לתאם פעולות במערכת.  בתוכנית זו הסינגלטון במחלקת ResourceManager שאחראית על טעינת התמונות, המוזיקה והטקסט, טעינה של כל תמונה תתבצע פעם אחת בלבד.

- **map**: מילון הממומש על ידי עץ חיפוש מאוזן, כל צומת בו שומר מפתח וערך במחלקת ResourceManager בהם התמונות נטענו.

- **command**: לתפריט.

- **strategy**: אסטרטגיה למימוש אויבים, אויב חכם ורגיל.

-**exception**

**אלוגריתמים הראויים לציון:**

- ה- constructor וה- distractor פועלים על ידי פולימורפיזם.

- אסטרטגיית תזוזת אוייבים **move** שנמצאת ב MoveBehavior פונקציה וירטואלית טהורה, אשר ממומשת במחלקות היורשות. נקראת כל פעם בהתאם לאויב הנוכחי.

- תזוזת האויב החכם שזז לפי המיקום של השחקן.

-collision בשיטת MultiMethod.

**באגים ידועים:**

**הערות נוספות:**

\*מהצגת הפרויקט שיניתי:

-אסטרטגיה של המיקום , שלחתי את הלוח לאויב ובדקתי אם הייתה התנגשות, אם כן בדקתי את מצב המשבצת אם האויב יכול להתקדם או לא, מיקום האויב החכם על שטח האויב בלבד והאויב הרנדומלי על שטח השחקן בלבד.

-ל Controller הורדתי פונקציות.

\*המרצה וויתרה לי על סגירת השטח ומה שנלווה לזה (שלבים וכד').