תרגיל 4

פרטי הסטודנט :

\*רון זמיר:203340484

\*אייל מזרחי : 203564265

**הסבר כללי של** תרגיל 4 **:**

**מימשנו משחק דוקים:** עם תצוגה גרפית וסאונד הנבנים באמצעות ספריית SFML.

המשחק נפתח עם תפריט בו ניתן לבחור להתחיל משחק, לראות עזרה (כללים והסבר על המשחק) ויציאה. לאחר בחירת האפשרות למשחק חדש ניתן לבחור רמת קושי:

קל, בינוני, קשה.

כללי המשחק ואופן הניקוד מפורטים במשחק בלחצן העזרה.

כמו כן הוספנו מעבר לדרישות גם צג המציג את התוצאה הגובה ביותר שהשחקן ביצע במשחקים שלו מהרגע שפתחת את התוכנית וכאשר מגיע לשיא חדש אזי כמובן התוצאה מתחלפת.

**תיכון:**

התוכנית מורכבת ממחלקת התפריט המכילה את מחלקת כפתור אשר יורשים ממנה סוגי הכפתורים בעת לחיצה על כפתור מופעלת פונקציית הפעולה שלו כמו כן כפתור המשחק חדש מכיל את הכפתורים של רמת הקושי ואובייקט של הקונטרול אשר מנהל את המשחק.

אשר מחזיק אובייקט של דוקים שהאובייקט של דוקים מכיל "הרבה" אובייקטים של דוק בודד אשר מנהלת את החיתוכים ואם ניתן להרים דוק או לא.

כמו כן כדי לבדוק האם דוק נחתך אם אחרים השתמשנו ברשימת שכנים של דוק עבור דוקים שמתחתיו כך שברגע שמרימים את הדוק העליון ביותר מעדכנים את הדוקים שמתחתיו שהוא כבר לא במשחק.

**רשימת הקבצים שנוצרו ע"י הסטודנטים :**

Button.h/cpp: מוריש לכל הכפתורים בתפריט, יש לו פונקציות אם לחצו על הכפתור או לא.

שומרת את התמונה של הכפתור.

: start\_menu.h/cpp אחראי על הצגת התפריט מכיל וקטור פולימופריזם של כפתורים.

ButtonExit.h/cpp: יורש ממחלקת כפתור, כאשר לוחצים על הספרייט שלו בתפריט אזי נסגר החלון.

ButtonaNEWGAME.h/cpp: יורש ממחלקת כפתור, מאתחל אובייקט של buildlevel ומריץ משחק.

COntrolel.h/cpp: אחראי על הפעלת המשחק קליטה מהשתמש שומר בתוכו את הדוקים , בסוף מציג הודעה אם ניצח או הפסיד או שבר שיא! .

מאגד בתוכו את כל האובייקטים של המשחק חוץ מהתפריט והכפתורים.

ButtonHELP.h/cpp: יורש ממחלקת כפתור, מחזיק תמונה של הוראות המשחק, כאשר נלחץ יציג את התמונה.

GameTime.h/cpp: אחראי על זמן המשחק , הדפסתו

GameScore.h/cpp: אחראי על ניקוד המשחק , הדפסתו ובדקת שבירת שיא

Dok.h/cpp: מחלקת דוק מכילה את כל התכונות של דוק.

DokS.h/cpp: מכילה מבני נתונים של דוקים (מפה ממוינת לפי X , וגם טבלת גיבוב כדי לשימור על סדר יצירה והדפסתם)

אחראית בדיקת חיתוכים בין דוקים לפני ריצת המשחק ושמירה עבור כל דוק ודוק האם הוא מעל מתחת לדוק אחר

אחראית על כל מה שקשור לדוקים במחשק מחזירה משנה בוליאני לקונטרולר עבור כל בקשה שלו אם זה ניתן או לא

**מבני נתונים עיקריים ותפקידיהם :**

map : במחלקת דוקים שומרת את הדוקים במחשק ממוינים לפי כאורדינטת X, גישה יעילה עת לחיצה על דוק עוברים על הדוקים בטווח הקורב ביותר ללחיצה בנוסף עת בונים את רשימת השכנים בתחילת ריצת התוכנית עבור כל דוק בודקים האם יתכן כי יחתך אם דוק אחר לפי המיון של הX

Unored\_map: : כדי לשמור את הדוקים בסדר סדרתי לפי איך שהם נוצרו כולומר מי שנוצר ראשון יופדס ראשון כמו כן המפתח הינו האינדקס הראשוני שבו הם נבראו וכך מבנה זה משמש להדפסת הדוקים כאשר צריך למחוק דוק מבצעים זאת בפעולה אחת מהירה של FIND לפני האינקדס המקורי של הדוק עת הוא נוצר (משתנה השמור בכל דוק) ואז מוחקים אותו .

וקטור של הכפתורים אשר בעצם מצביעים חכמים העובד בעקרון הפולימורפיזם.

**אלגוריתמים הראויים לציון :**

**בניית רשימת שכנים עבוד כל דוק בתחילת בניית המשחק.**

**עת מסיימים לייצר את כל הדוקים יש קריאה לפונקציה אשר בונה רשימת שכנים לכל דוק כלומר כל דוק שהינו מעל דוק אחר : הדוק שלמעלה מחזיק את הדוק שלמטה והדוק שמעליו יש דוק מעלים את המונה שלו באחד כך שבודקים האם המונה של דוק שווה לאפס אזי ניתן להרים אותו אחרת לא ניתן.**

**כל בדיקת לחיצה על דוק מתבצעת במפה של דוקים ממוינת בזמן ריצה לוגריתמי**

**באגים ידועים : לא ידוע על באגים.**

**הערות אחרות :** אין.