**Oop 2**

**תרגיל 4 – "circle the cat"**

**מגישות: נועם מרג'ני & אפרת אילוז**

**נועם –**

**אפרת –**

**=====================================================================**

תיאור

**=================**

**משחק זה הוא מז'אנר משחקי הפאזל, בהם מוצג לשחקן לוח עם "חידה" שהוא צריך לפתור לפי כללי המשחק.**

**לשחקן מוצג לוח אריחים (-עיגולים) עם "אריחים חסומים" (-העיגולים הכהים) ו"אריחים ריקים" (-העיגולים**

**הבהירים). במשחק ישנו חתול, שיכול ללכת על העיגולים הריקים בלבד. השלב מתחיל עם מספר עיגולים שכבר**

**מלאים, כשהחתול עומד במרכז על אריח ריק.**

מטרת המשחק:

**על השחקן להקיף את החתול מכל צדדיו, בעזרת "מילוי האריחים", כך שלא יהיה לו לאן ללכת**

רשימת קבצים

**=================**

**main.cpp**

**Controller.cpp**

**ניהול המשחק.**

**Board.cpp**

**הצגת הלוח, טיפול בו ובאריחים ע"י מבני הנתונים של האריחים.**

**Resources.cpp**    
מחזיק את העזרים שנוספו למשחק ומנהל את המשאבים.

**GameScreen.cpp   
מחזיק את מסך** המשחק**.**

**Enemy.cpp**   **מנהל את החתול – נקרא enemy כדי לאפשר גנריות והחלפת החתול בדמות אחרת לצורך העניין.**

**levelHandler.cpp**   **מנהל את השלבים של המשחק.**

**Button.cpp**    
**אחראי על הכפתורים במשחק (undo, reset).**

**Stack.h**

מנהל את מבנה הנתונים מחסנית.

**Macros.h**

**מכיל בתוכו את כל ה- consts enum, שהשתמשנו בהם.**

לכל קובץ cpp קיים קובץ h ולכן סך הכל קיימים 8 קבצי cpp 9 קבצי h. = 17

תיכון

**=================**

**מחלקת controller:**

**מחלקה זו היא המחלקה הראשית אשר מנהלת את התנהלות המשחק.**

**מחלקת Board:**

**מחלקה זו מנהלת את לוח המשחק, המחלקה מחזיקה בתוכה עיגולים שמרכיבים את הלוח, העיגולים דייכים לספריית sfml והם מטיפוס circle shape.**

**מחלקת Resources:**

**מחלקה זו היא מחלקה ראשית שאותה מימשנו באמצעות Singelton (שהוא סוג של Design Pattern) ואליו ניתן לגשת בעזרת מצביע יחיד לנתונים משאר המחלקות. מחלקה זו היא נועדה עבור עזרים חיצוניים שקשורים למשחק: טעינת תמונות, סאונדים וגופנים.**

**מחלקת GameScreen:**

**מנהלת את המסך, למשל מדפיסה את מס' הצעדים ומדפיסה את מסכי המעבר שנדרשו במשחק.**

**מחלקת Enemy:**

**מחלקת האיוב, כלומר במקרה שלנו – החתול. נמצאים בה חישובים שקשורים לחתוך וכן האנימציה שלו.**

**מחלקת levelHandler:**

**מחלקה זו מנהלת את השלבים של המשחק על ידי יצירת שלב והגרלת אריחים לפיזור בלוח.**

**מחלקת Button:**

**מחלקה זו מחזיקה את הכפתורים במשחק (undo, reset) ומנהלת אותם במחלקות השונות.**

**מחלקת Stack:**

**מחלקה תבניתית שמנהלת את המחסניות במשחק (נשתמש בה במחלקת enemy וב- board כי שם אנו משתמשים במחסנית).**

**מבני נתונים**

=================

**השתמשנו במבנה הנתונים 'מחסנית' על מנת לנהל את כפתור ה- UNDO.   
השתמשנו בוקטורים.**

**אלגוריתמים לציון**

=================

**על מנת למצוא את המסילה הקצרה ביותר שתוביל את החתול מחוץ לגבולות הלוח מימשנו את אלגוריתם הbfs , על ידי זה ינצח החתול את המשתמש.**

**באגים לציון**

=================

אין כאלו.

**הערות נוספות**

=================

**-השתמשנו באנימציה עשירה של החתול על מנת שהקפיצות והתזוזות יראו דומות למקור ועל מנת שהמשחקיות תהיה עשירה ומעניינת יותר.**

**-מימשנו מבני נתונים מיוחדים כגון stack.**

**-יצרנו מחלקה תבניתית "stack.h" .**

**-מימשנו את אלגוריתם ה- bfs כדי שהחתול יברח בצורה היעילה ביותר.**

**- מימשנו גם כפתור reset שמחזיר אותנו אל תחילת השלב.**

**-במשחק יש שלושה שלבים (אפשרות להוסיף עוד בעתיד), השלב הראשון בעל 11-14 אריחים (הקל ביותר) ובשלבים הבאים דרגת הקושי עולה על ידי ירידה בכמות האריחים.**

**-מימשנו שכשהחתול מנצח ומגיע לאחד מקצוות הלוח הוא גם בורח ע"י האנימציה.**