**תרגיל 4 סמסטר ב' נושא התרגיל – SixColors**

שם המגישים: מיכאל אורייב ישעיה וינסטוק

ת.ז. - michaelur 317458123

ת.ז. ishaiawi – 304939234

***הסבר כללי:***

נתבקשנו לבנות את המשחק SixColors. המשתתפים הם שחקן יחיד מול מחשב כאשר המשחק מתנהל לפי תורות .מדובר על לוח שמורכב ממשולשים ומעויינים שיכולים להיות אחד מתוך 6 צבעים שונים, כל אחד בתורו יכול לבחור אחד מתוך ששת הצבעים במטרה לכבוש כמה שיותר צבעים שמסביבו, אי אפשר  לבחור צבע שנבחר על ידי השחקן היריב בתור הקודם. השחקן מתחיל מ-פינה שמאלית תחתונה והמחשב יתחיל מפינה ימנית עליונה, השחקן הראשון שיצליח לכבוש יותר מחצי מכמות המשולשים/מעוינים על הלוח ינצח.

***תיכון:***

game.h/.cpp

אחראי על יצירת הצורות ומילוי רשימת השכנים, ציור הלוח, תזמון של מי התור השחקן או המחשב, מחזיק בתוכו וקטור של צורות לצורך יצוג הלוח(הצורות יכולות להיות מ-2 סוגים שונים כמפורט בהמשך)

Buttons.h/cpp

אחראי אל לחיצת הכפתורים ובחירת הצבע הרצוי, מצייר את הכפתורים, הכפתור יקבל מיקום שנלחץ מהgame ויודיע לו על איזה צבע לחצו.

Shape.h/cpp

מייצג אחת מ-2 צורות, מכיל בתוכו וקטור של רשימת שכנים וסימני זיהוי של צבע, האם הצורה כבר נכבשה על ידי  המשתמש או המחשב, יכולת צביעה, ביצוע רקורסיה ועוד דברים הכרחיים.

יורש מ - std::enable\_shared\_from\_this<Shape>

Triangle.h/cpp

מייצג אחת מתוך שתי הצורות שצויינו לעיל מייצג משולש. יורש מshape ויש לו את כל התכונות שצוינו למעלה

TwoTriangles.h/cpp

מייצג אובייקט המכיל בתוכו שני משולשים. יורש מ- Triangleומתאים כל פונקצייה של Shape כך שתפעל על כל משולש בצורה נכונה

Rectangle.h/cpp

מייצג אחת מתוך שתי הצורות שצויינו לעיל מייצג מעויין. יורש מshape ויש לו את כל התכונות שצוינו למעלה

***התקשורת בין האובייקטים כדלקמן:***

המשחק מתחיל כאשר הלוח נוצר תחילה על ידי בניה של מטריצה מסוג אובייקט shape, ולאחר מכן מהמטריצה בונים וקטור אחד ארוך שבנוי על פי רשימת שכנים(הצורות שנוצרות יכולות להיו1 מ-2 סוגים והיצירה שלהם קוראת בצורה וירטואלית) בתחילת המשחק המחשב והשחקן נמצאים כל אחד בפינה משלו עם משולש בודד ברשותם, כאשר השחקן לוחץ על כפתור צבע מסויים הgame- מזהה את מיקום לחיצת העכבר ושולח את המיקום שנלחץ אל הbutton שמחזיר לgame תשובה איזה צבע נלחץ, לאחר שהשחקן מסיים את התור שלו המחשב בוחר צבע ועושה את התור שלו וכך ממשיכים. בעת צביעת הלוח מתבצעת רקורסיה בשביל לצבוע את כל השכנים שמסביב, הרקורסיה מופעלת מהgame אבל יש גם חלק שקורה בתוך האובייקט shape. כאשר המשחק מסתיים תוכרז הודעת ניצחון/כישלון בהתאם.

***רשימת הקבצים שנוצרו:***

**h file**

Buttons.h

Game.h

Rectangle.h

resource.h – נוצר ע"י הקומפיילר

Shape.h

Triangle.h

TwoTriangles.h

**cpp files**

Buttons.cpp

Game.cpp

Main.cpp

Rectangle.cpp

Shape.cpp

Triangle.cpp

TwoTriangles.cpp

***מבני נתונים:***

*מבני נתונים זמניים*

std::vector<std::vector<std::shared\_ptr<Shape>>> tempMatrix; //The shape matrix

std::vector<std::shared\_ptr<Shape>> tempVector; //Each line in the shape matrix

*מבני נתונים בשימוש שותף במהלך המשחק*

std::vector<std::shared\_ptr<Shape>> m\_shapeVector;

std::vector<sf::RectangleShape> m\_buttons;

std::vector<std::shared\_ptr<Shape>> m\_neighbors;

*אלגוריתמים מתוחכמים:*

1. אלגוריתם רקורסיבי של צביעת הלוח לשחקן ולמחשב(משתמשים באותו האלגוריתם) – מעבר רקורסיבי על השכנים של כל צורה השייכת לאחד השחקנים וצביעה בהתאם.
2. אלגוריתם רקורסיבי למחשב לבחירת הצבע הבא – המחשב סופר כמה צורות מכל צבע מסויים הוא יכול "לתפוס" בתור הבא שלו ובוחר את האפשרות בעלת מספר הצורות הגבוה ביותר (אלא אם השחקן כבר בחר את הצבע הזה, ואז תיבחר האפשרות הבאה)

*באגים:*

/

*הערות:*

/