

עבודת גמר Java

מגיש : עומר דהן



**עבודת גמר Java**

בפרויקט הגמר בחרתי לאתגר את עצמי ולפתח משחק Snake בסיסי בשפת Java, ובנוסף לכך גם להוסיף שימוש במחלקה גרפית על מנת ליצור משחק המבוסס GUI.

כללי המשחק הם להצליח ״לאכול״ כמה שיותר פריטים על הלוח המרכזי המוצג לעיני המשתמש, ועל מנת לא להיפסל אסור שהנחש יפגע בעצמו, ואסור לצאת מגבולות הלוח, בעת יציאה מגבולות הלוח המשחק מסתיים.

בפרויקט זה אין לי Tester הבודק את התוכנה, אלא רק נדרשת הפעלה של המשתמש, על מנת להפעיל את המשחק ניתן לפתוח את כל הקבצים ב IDE ולהריץ את המחלקה GameRun.java .

פרויקט זה היה מאתגר וכיפי ללמידה והכנה.

הפרויקט מחולק ל – 7 קבצי Java.   
  
1 ) Interface Drawable - ממשק המכיל פונקציה אחת בשם draw המקבלת כפרמטר אובייקט מסוג Graphic, ממשק זה משרת מטרה פשוטה מאוד בפרויקט, ממשק זה מגדיר חוזה שכל אובייקט במשחק שרוצה לצייר על הלוח בעצם צריך לבצע. שיטה זו צריכה להיות ממומשת בכל מחלקה התרצה לבצע ציור באמצעות המנועים הגרפיים של java.

2) GameObject Abstract Class – מחלקה זו משמשת כמחלקה בסיסית לכל האובייקטים במשחק שיש להם מיקום על לוח המשחק, קואורדינטות של x,y.  
מחלקה זו מתאימה לפרויקט מכיוון שמחלקות נוספות זוכות לתכונות והתנהגות משותפת על מנת להבטיח שכפול של קוד

3) enum Direction – מייצג את הכיוונים האפשרים שהנחש במשחק יכול לנוע ( מעלה, מטה, שמאלה, ימינה ). בעזרת כך נוכל לשלוט בתנועות הנחש, מה שהופך את הקוד לקריא יותר ומונע שימוש במספרים שלמים שרירותיים או מחרוזות על מנת לייצג כיוונים לדוגמא אם נרצה להשתמש במקש חץ עליון נצטרך להשתמש במספר 40, ואת זה רציתי לחסוך על מנת לפשט את קוד התוכנה.

4) GameBoard Class – מחלקה זו פועלת כפאנל של המשחק הרשאי שבו מצוירים כל האלמנטים החזותיים של המשחק ובנוסף גם הלוגיה של המשחק נמצאת במחלקה זו.   
המחלקה הזו מחברת את מרכיבי המשחק ביחד, היא מנהלת את חפצי המשחק, היא מאזינה למקשי מקלדת הנלחצים על מנת לשנות את כיוון הנחש, היא בודקת האם נאספו אלמנטים מהלוח על מנת להגדיל את הנחש, וגם מבצעת את הציור של חפצי המשחק על הלוח. מחלקה זו היא המרכז בפרוייקט שלי.

5) GameRun Class – מחלקה המאתחלת את המשחק ומריצה את החלון הראשי שמוצג למשתמש על מנת לשחק במשחק, מחלקה זו מאתחלת את מחלקת GameBoard ויוצרת אובייקט על מנת להגדיר את חלון המשחק מבחינת גודל, נראות וכו׳.

6) Item Class – מחלקה זו מייצגת את הפרטים שהנחש אוסף מהלוח על מנת לגדול, פריטים במשחק יופיעו בצבע אדום על גבי הלוח, אותם פריטים מנוהלים על ידי המחלקה GameBoard שבודקת האם הנחש הצליח ״לאכול״ את אותם פריטים כדי להפעיל פונקציה של גדילת הנחש.

7) Snake Class – מחלקה זו מייצגת את הנחש עצמו אשר נשלט במשחק, מחלקה זו היא גם מחלקה מאוד מרכזית במכניקה של המשחק, היא מנהלת את הגדילה של הנחש, התזוזה שלו , והציור שלו על גבי הלוח פאנל הראשי.

**מבני הנתונים שבהם השתמשתי בפרויקט ומה הסיבה:**

1. LinkedList עבור גוף הנחש – המחלקה Snake משתמשת ברשימה מקושרת בכדי לייצג את גוף הנחש, כל מקטע של גוף הנחש הוא GameObject ואותם קטעים מאוחסנים ברשימה מקושרת, בחירה זו אידאלית לפרוייקט שלי מכיוון שהיא מאפשרת הוספה והסרה יעילה של מקטעים בשתי הקצוות של גוף הנחש.

כשהנחש זז מתווסף ראש חדש לחזית הנחש, והזנב מוסר מהקצה. שימוש במחסנית עוזרת לי בעצם לייצר צמיחה ותנועה דינמית של הנחש בעת משחק.

1. ArrayList לאחסון נתונים לציור ( Drawble) – המחלקה GameBoard שומרת על רשימה של אובייקטים מסוג Drawble שאותם ניתן לצייר, בחרתי להשתמש ב ArrayList בשל הגמישות היעילה לאלמנטים. מכיוון שהמשחק חוזר על אובייקטים מסוג Drawble לעיתים מאוד קרובות בכדי להציג אותם. ומכך לאחר בדיקה שביצעתי הבנתי ש ArrayList מבחינת ביצועים מאוד יעיל. ובנוסף זה מתאים להוספה והסרה דינמית של פרטים בקלות, למשל כאשר הנחש ״אוכל״ את הפריט מהלוח וצריך לבצע הסרה של הפריט מהלוח המשתקף למשתמש במסך.
2. List בשביל Items - על מנת לנהל את הפריטים שנאספים על ידי הנחש במחלקת GameBoard יש שימוש בממשק List.   
   בחרתי להשתמש בממשק זה מכיוון שהוא מאפשר איטרציה יעילה על פריטים , הוספה והסרה ישירה של אותם פריטים כפי שהם מופיעים במשחק.