

Design Pattern

חלק א':

1. משחק חלליות מתחיל במסך פתיחה אשר בו כתוב "לחץ כדי להתחיל משחק חדש". בחלון זה ניתן רק ללחוץ על כפתור 'התחל משחק חדש'. כאשר המשחק מתחיל יופיע כיתוב "בהצלחה". בתחילת המשחק יש לחללית השחקן (חללית טובה) 200 נקודות פגיעה. ועל השחקן לפוצץ 100 חלליות אויב בכדי לנצח.

ישנם מספר אירועים אפשריים:

- חללית השחקן נפגעה מטיל אויב. במקרה זה מספר נקודות הפגיעה יפחת ב-X אם מספר נקודות הפגיעה ירד מתחת ל- 50% המשחק עובר ל"סכנה" ותופיע כיתובית "זהירות!" אם מספר נקודות הפגיעה של חללית השחקן הגיעה ל- 0 המשחק עובר ל"הפסדת! נסה שוב! לחץ על כפתור 'התחל משחק חדש'"
- חללית השחקן פגעה בחללית אויב ולכן מספר חלליות האויב יפחת ב- 1 אם נשארה חללית אויב 1, המשחק עובר ל"מלחמה נגד בוס גדול"
- אם נשארו 0 חלליות אויב המשחק עובר ל"נצחון גדול! לחץ על כפתור 'התחל משחק חדש'"
- חללית השחקן התרפאה, במקרה כזה יגדל מספר נקודות הפגיעה של החללית ב-X נקודות אם כמות נקודות הפגיעה עלתה ממתחת 50% למעל 50% יש להציג כיתובית "חזרה לשגרה"
- לחיצה על כפתור 'התחל משחק חדש' במהלך המשחק. במקרה כזה, לא יקרה שום דבר
- בכל פעם שחללית השחקן נפגעת במצב 'סכנה' – יש להדפיס את מספר נקודות הפגיעה שנשארו לחללית

במצב מלחמה נגד בוס גדול, יש לפגוע בבוס 50 פעמים כדי להשמידו

2. ממש עץ מיספרים ופונקציות Sum אשר מחשבת את סכום המספרים מאותו ענף/עלה ועד לתחתית (לא ברקורסיה). גם לעלה וגם לענף יש מספר. ממש גם פונקציה הבודקת אם כל הרכיבים בעץ מכילים מספרים זוגיים בלבד – כן ברקורסיה.

3. ייצר מערכת שליטה/בקרה במכונית: התחל נסיעה, סיים נסיעה, הגבר מהירות, האט, פנה ימינה, פנה שמאלה, הצג מצב דלק, הצג מיקום. כעת כתוב ממשק אשר נותן גישה רק לפונקציות הבאות: הצג מצב דלק, והצג מיקום

4. בנה מבנה נתונים המכיל רשימה של עובדים. לכל עובד יש מספר ת.ז., שם, גיל, שכר. בנה פונקציה המקבלת את הרשימה ומדפיסה אותה, אך מכיוון שמקבל הרשימה עשוי "לפגוע" בה, בנה את הפונקציה כך שהמשתמש יקבל "עותק" ולא את המקור. ממש פונקציית מיון לרשימה. אם מספר האיברים ברשימה קטן מ- 100 השתמש במיון לפי ת.ז. בין 100-200 מיון לפי שם ומ- 300 ומעלה מיון לפי שכר

5. בנה מחלקה בשם תמונה (אשר תכיל מטריצה של 100X100 מספרים שלמים). ושדה פורמט. בנה מחלקות פילטרים שונים:

1. פילטר לפי צבע. למען הדוגמא פשוט הגדל את כל המספרים ב- 1
2. פילטר לפי קצוות. למען הדוגמא אפס את שורות 0-1 וגם את 98-99
3. פילטר לפי רזולוציה. למען הדוגמא מחק כל ערך הגדול מ 50,000
4. פילטר לפי דמויות. למען הדוגמא הגרל אזור בתמונה וחלק ב-2 את כל התאים באותו אזור

כעת צור תמונה והעבר תמונה בין כל הפילטרים (מ- 1 ועד 4). אם באחד השלבים גילית שכל תאי המטריצה שווים ל- 0 אז צא. כעת צור אפשרות להעביר מטריצה לפי פילטר 3,4 והעבר שם תמונה

6. לפניך מערכת ציור והדפסות תלת-מימדית. קיימות במערכת הצורות הבאות: משולש, עיגול ומלבן. הוחלט שלכל צורה במחלקת האבא (SHAPE) יהיו הפונקציות האבסטרקטיות של הדפסה תלת-מימדית, הדפסה דו-מימדית ועדכון רזולוציה. מכיוון שהמימוש בכל מחלקה אינו פשוט וכמו כן עומדים להוסיף מחלקות רבות בפרוייקט- הוחלט לאגד את הפונקציונליות הזאת במחלקות נפרדות- מחלקה של הדפסה תלת-מימדית, מחלקה של- הדפסה דו-מימדית ומחלקה של- עדכון רזולוציה. בנה את המערכת
7. בנה באמצעות pattern מתאים אפליקציית קניה ומכירה של רכבים. כל משתמש שנרשם יכול להיות קונה או מוכר. כל קונה, כשנרשם, נותן את שמו + שם הרכב שאותו רוצה לקנות + מספר טלפון. כל מוכר שנרשם כותב את שמו + שם הרכב שהוא מוכר + מספר טלפון. בעת רישום מוכר או קונה נבדק אם יש במאגר הצעה רלוונטית, אם כן נשלחת הודעה גם למוכר וגם לקונה של שמו + מספר הטלפון של הספק/קונה. אם אין הצעה רלוונטית המשתמש נרשם במערכת, עד לרגע שבו תתקבל הצעה רלוונטית
8. בנה באמצעות pattern מתאים מערכת אימונים בחדר כושר. כל אימון מורכב מ- חימום (יכול להיות לדוגמה הליכה מהירה או ריצה קלה וכו'), אימון ארוכי (כגון משחק עם כדור וכו'), אימון משקולות (ידיים או רגליים וכו') ומתיחות (מתיחות גמישות או מתיחות לחץ וכו'). בנה מחלקות מתאימות לצורך האימונים
9. לפניך מפעל מכוניות המייצר מכוניות משני סוגים : מכונית היברידיות ומכוניות רגילות. חלקי הרכב הזהים בכל סוגי המכוניות הם: פח, גלגלים, מושבים. חלקי הרכב השונים בין הרכב ההיברידי לבין הרכב הרגיל הם: מנוע, מערכת קירור, מערכת תאורה ובטריה. ייצר פונקציה המייצרת את כל חלקי הרכב בהתאם לסוג הרכב המבוקש. המשתמש ב- pattern מתאים [אזהרה: הרבה מחלקות...]
10. לפניך תוכנת חלונות. כל חלון יכול להכיל את הרכיבים הבאים: תאורה תלת-מימדית, מסגרת ציבעונית, שיקוף-חלקי, הצללה, ורקע מהבהב. החלון הבסיסי הוא ללא אף אחד מהרכיבים האלה. יש לאפשר בנייה של החלון בצורה גמישה ויעילה, כאשר מספר הקומבינציות יכול להיות מורכב משילוב של כל אחד מהרכיבים. כמו כן יהיה אפשר להוסיף רכיב לחלון במהלך התוכנית (ללא צורך ביצירה של מופע חדש של החלון)
11. בנה סימולציה של תנועה של אובייקט במרחב. המרחב בנוי ממטריצה של 5×5 . האובייקט הוא מספר (כאשר בהתחלה הוא "1") אשר יוצב במרכז המטריצה. בתחילה: תוצג לפניך המטריצה- כאשר כל משבצת תצויר ע"י נקודה (".") מלבד המשבצת בה נמצא האובייקט ששם יודפס "1".
- כעת המשתמש יזין קלט מהאפשרויות הבאות:
- "1" יזיז את האובייקט כלפי מעלה במטריצה
- "2" יזיז את האובייקט כלפי מטה במטריצה
- "3" יזיז את האובייקט כלפי ימינה במטריצה
- "4" יזיז את האובייקט כלפי שמאלה במטריצה
- "+" יגדיל את המספר (כלומר לדוגמה, ה- "1" יהפוך ל- "2". ניתן להגדיל עד 9)
- "-" יקטין את המספר (כלומר לדוגמה, ה- "2" יהפוך ל- "1". ניתן להקטין עד "0")
- "5" ישמור מצב (את מיקום האובייקט ואת מספרו)
- "L" ישחזר את המצב (יחזיר את האובייקט למיקום שנשמר ולמספר שנשמר)
- "X" יציאה מהסימולציה
- בכל פעם שהאובייקט יצא מגבולות המטריצה בדוק אם יש מצב שמור וחזור אליו. אם אין מצב שמור- צא מהסימולציה

12. במעבדה לבניית מחשבים יש לבנות מחשבים בסדר פעולות קבוע. הסדר הוא: בניית מארז, הוספת לוח אם, הוספת מעבד, הוספת כרטיס גרפי, הוספת זיכרון והרצת בדיקה. השתמש ב design pattern מתאים כדי לייצר מחלקות מחשבים מהסוגים הבאים: מחשב משחקים (מעבד חזק, הרבה זיכרון וכרטיס גרפי חזק), מחשב משרדי (חומרה בינונית), מחשב סלון (לוח אם דק במיוחד ומארז דק, כרטיס גרפי חזק ומעבד בינוני)

13. במערכת לניהול טיסות ישנו interface IFlightDao המכיל 3 פונקציות – הוספת טיסה, עדכון טיסה ומחיקת טיסה. פונקציות אלה מקבלים כפרמטר טיסה- מחלקת Flight (שבה יש מספר טיסה, כמות נוסעים, שם חברת העופה, מדינת מקור ומדינת יעד). ישנו ה- FlightFacade אשר מממש את הפונקציות המדוברות. כמו כן יש פונקציות המקבלות כפרמטר IFlightDao וקוראות לשלושת הפונקציות המדוברות עם אובייקט מסוג Flight. עד כה נשלח FlightFacade כפרמטר. כעת הגיעה ערכת חבילה של פונקציות (צד ג') המממשות הוספת טיסה, עדכון טיסה ומחיקת טיסה בצורה טובה יותר, הבעיה שהן מקבלות Encrypted_Flight שהוא אובייקט דומה ל Flight רק שיש לו שדה נוסף של חישוב סך כל תוי המחרוזות + כמו כן, כל המחרוזות שבתוכן הן הפוכות. כיצד ניתן לגרום לתוכנית לעבוד מבלי לשנות את חתימות הפונקציות המקוריות?

14.

חלק ב- שאלות מחשבה:

1. לפניך רשימת מילים. עליך לגלות כמה פעמים כל מילה חוזרת על עצמה (גם בסדר אותיות שונה). לדוגמא:

“java”, “jjava” “vaj”, “aavj”, “j”, “vja”, “dan”, “and”, “ddan”

התשובה תהיה:

“java” - 3 פעמים (“java”, “aavj”, “vja”)

“jjava” - 1 פעמים

“vaj” - 1 פעמים

“j” - 1 פעמים

“dan” - 2 פעמים (“dan”, “and”)

“ddan” - 1 פעמים

מותר לעבור על הרשימה רק פעם אחת

2. בנה תוכנית החושפת את האפשרויות הבאות:

- להכניס מספר
- לשלוף את המספר האחרון שהוכנס
- להציץ במספר האחרון שהוכנס (מבלי להוציאו)
- [רמז: מחסנית]
- הדפס את המספר הגדול ביותר שקיים כרגע במחסנית
- הדפס את המספר הקטן ביותר הקיים כרגע במחסנית

שים לב: בכל פעם שמכניסים מספר ו/או מוציאים מספר- המספר הגדול ביותר והקטן ביותר עשויים להשתנות...

אסור להשתמש במערך או רשימת עזר כלשהי

אסור לטובת מציאת המספר הגדול ביותר/הקטן ביותר להוציא את כל איברי המחסנית ולהכניסם
שוב

בהצלחה