**פרוייקט 2 – ירושה ופולימורפיזם**הגשה עד יום **ד' 19/6/2019 ב 12:00 בצהריים**.

עליכם לממש משחק מונופול. עליכם לייצר מערכת התומכת בדרישות הבאות:

**לוח המשחק:**

בלוח (**board**) המונופול 18 משבצות (ממוספרות מ-1-18) כל משבצת (**slot**) יכולה להיות נכס ( **Assetבציור מסומנת בכתום**) , משבצת דרך צלחה (**Go- בציור מסומנת בכחול**) , משבצת כלא (**Jail** – בציון מסומנת ירוק) או משבצת מפעל הפיס (**Chance** – בציור מסומנת סגול).

כל נכס שייך לעיר מסוימת (city1,city2,city3,city4…)

הלוח- יכיל את מערך המשבצות :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 10) Chance | 11) City 3  Asset 1 | 12) Chance | 13) City 3  Asset 2 | 14) City 3  Asset 3 |
| 9) City 2   Asset 3 |  |  |  | 15) City 4  Asset 1 |
| 8) City 2   Asset 2 |  | Monopoly board | | 16) City 4  Asset 2 |
| 7)Chance |  |  |  | 17) City 4  Asset 3 |
| 6) City 2   Asset 1 |  |  |  | 18) City 4  Asset 4 |
| 5) Jail | 4) City 1 Asset 3 | 3) City 1 Asset 2 | 2) City 1 Asset 1 | 1) Go : get 350 Nis |

**המחלקה Slot**

**מחלקה זו היא מחלקה מופשטת.**

**המחלקות שיורשות ממנה הן: Asset, Go, Jail, Chance.**

**מתודות:**

* בנאי – יש ליצר בנאי המאתחל את מספר המשבצת. קחו בחשבון שבאופן עקרוני התוכנית עלולה לייצר יותר מלוח משחק אחד.
* יש להצהיר על המתודה virtual bool play(Player\* p) = 0
* יש להצהיר על המתודה virtual string get\_name() const = 0
* **יש לממש מחסל, אופרטור השמה ובנאי העתקה במידת הצורך.**
* **Operator overload** לפי הצורך.

**המחלקה Asset**

**שדות:**

* עיר – מסוג string.
* שם קבוצת נכסים – מסוג string.
* מחיר קניית בית – משתנה מסוג int – מאותחל למספר שלם המוגרל בין 50-150
* דמי שכירות - משתנה מסוג int - מספר שלם המוגרל בין 5-50.
* נכס יכול להיות חופשי או משויך לשחקן מסוים.

**מתודות:**

* **בנאים, מתודות get,set ומחסל על פי הצורך.**
* **Operator overload** לפי הצורך.
* **מימוש המתודה play** – מתודה זו תקרא לאחר קידום השחקן בלוח.

אם הנכס שייך לשחקן אחר יש לבצע גביה של תשלום דמי שכירות ע"י קריאה למתודה

bool pay\_rent(int amount);

הנמצאת במחלקת player.

במידה ופעולה זו הסתיימה בכישלון (לשחקן אין מספיק כסף) המתודה תחזיר false – ויש להפסיק את המשחק.

אם הנכס לא שייך לאף שחקן המתודה תשאל את השחקן אם רוצה לקנות את הנכס ואם כן תקרא למתודה

bool add\_asset(Asset\* a);

הנמצאת במחלקת player.

* מימוש המתודה get\_name – מחזירה את המחרוזת שתודפס בלוח.

**המחלקה Go –**

**שדות:**

* טקסט ההנחיה – מחרוזת מסוג string.

**מתודות:**

* **בנאים, מחסל, אופרטור השמה ובנאי העתקה לפי הצורך**
* **Operator overload** לפי הצורך.
* יש לממש את המתודה play כך שלא תעשה כלום.   
  המתודה תחזיר true מאחר והמשחק אינו נגמר.
* מימוש המתודה get\_name – מחזירה את המחרוזת שתודפס בלוח.

**המחלקה Jail –**

**שדות:**

* טקסט ההנחיה – מחרוזת מסוג string.

**מתודות:**

* **בנאים, מחסל, אופרטור השמה ובנאי העתקה לפי הצורך**
* **Operator overload** לפי הצורך.
* יש לממש את המתודה play כך שתדפיס את ההנחיה לשחקן ותעדכן שהוא בכלא. המתודה תחזיר true מאחר והמשחק אינו נגמר.
* מימוש המתודה get\_name – מחזירה את המחרוזת שתודפס בלוח.

**המחלקה Chance –**

**שדות:**

* טקסט ההנחיה – מחרוזת מסוג string.

**מתודות:**

* **בנאים, מחסל, אופרטור השמה ובנאי העתקה לפי הצורך.** הבנאי מקבל כפרמטר את הסכום שזוכה בו השחקן כאשר הוא דורך על משבצת זו (אם הסכום שלילי הוא ירד מהמאזן של השחקן)**.**
* יש לממש את המתודה play כך שתדפיס את ההנחיה לשחקן. ותקרא למתודה:

void set\_balance(int balance);

שנמצאת במחלקה Player.  
במידה ופעולה זו הסתיימה בכישלון (לשחקן אין מספיק כסף) המתודה תחזיר false – ויש להפסיק את המשחק.אחרת המתודה תחזיר true.

* + מימוש המתודה get\_name – מחזירה את המחרוזת שתודפס בלוח.

**המחלקה Player:**

**שדות:**

* מספר סידורי של השחקן
* שם השחקן
* מאזן
* אינדיקטור אם נמצא בכלא
* מיקום בלוח ברגע נתון (אינדקס המשבצת)
* מצביע ללוח -שחקן יוכל לשחק במשחק אחד בלבד ברגע נתון.
* רשימת הנכסים שבבעלותו
* גודל רשימה זו

**מתודות:**

* **בנאי** המקבל את שם השחקן, מצביע ללוח בו משחק ומאזן התחלתי – ייווצר שחקן חדש בעל מספר סידורי, יתרה =350, אינדיקטור=false , מיקום התחלתי =1, רשימת נכסים ריקה.
* **בנאים, מחסל, אופרטור השמה ובנאי העתקה לפי הצורך**
* **Operator overload** לפי הצורך.
* עדכון אינדיקטור – כלא – מקבל סטטוס (true\false)
* מתודות get ו-set לכל השדות
* הוספת נכסים לשחקן –bool add\_asset(Asset\* a)

המתודה תקבל מצביע לנכס שהשחקן יכול לקנות , במידה ויש לו יתרה מתאימה יש להוסיף את הנכס לרשימת הנכסים שבבעלותו ולעדכן את הבעלים של הנכס להיות השחקן המפעיל. יש לעדכן את יתרת הלקוח במחיר הנכס. המתודה תחזיר חיווי אם הקניה הצליחה.

* חיוב שחקן – bool pay\_rent(int amount)   
  מיועד עבור שחקנים שנפלו בנכס שאינו בבעלותם עליהם לשלם דמי מעבר.
  + במידה ויתרת הלקוח מספקת לתשלום – יחויב הלקוח ויוחזר true.
  + אחרת – יש להחזיר את נכסי השחקן – יתבצע על בסיס רשימת הנכסים של השחקן כך שהנכס הראשון שנכנס הוא הראשון שיוחזר – ההחזרה תתבצע על הנכסים עד שיתרת הלקוח תאפשר את תשלום דמי השכירות (amount). במידה ולא הושג הסכום – המתודה תדפיס game over ואת שם השחקן ותחזיר false.
* **זריקת קוביות -** bool draw\_dice()

המתודה תבצע בדיקה האם השחקן בכלא – אם כן יש לעדכן את הסטטוס שלו שלא יהיה בכלא לטובת התור הבא ולהחזיר true.

במידה והשחקן אינו בכלא יש להטיל קובייה (מגרילים מספר 1-6) ולהתקדם בלוח על פי מה שיצא בקובייה. שימו לב שהלוח מעגלי.

בכל פעם שמסיימים סיבוב בלוח יש להוסיף 350 ₪ ליתרת הלקוח. יש להדפיס את מיקום השחקן בלוח לפני ואחרי זריקת הקוביה ולעדכן את מיקומו בלוח בהתאמה.

לאחר מכן יש לקרוא למתודה play במחלקה המתאימה מתוך המיקום החדש בלוח. ולהחזיר את הreturn שלה.

**המחלקה Board – :**

באתר הקורס נמצאים הקבצים Board.h ו Board.cpp לשימושכם.

**Main**

בניית המשחק:

* יש לייצר לוח לפי המתואר בתרשים.
* יש לייצר מערך של שני שחקנים.

**מהלך המשחק:**

* בתחילת המשחק כל משתתף מקבל 350 ₪.
* המשחק מתקיים בתורות.
* כל משתתף משחק בתורו על פי מספרו הסידורי.
* בתחילת כל תור יש לשאול את השחקן אם להמשיך במשחק, להדפיס את הלוח או לפרוש.
* במידה וממשיך לשחק יש לבדוק אם הוא בכלא-
  + אם כן – יש לדלג על תורו ולעדכן את סטטוס יציאתו מהכלא עבור התור הבא.
  + אחרת -המשתתף בתורו מטיל קובייה (יש להגריל מספר בין 1-6) ומתקדם בהתאם על מערך לוח המשחק (מערך מעגלי)
  + אם המשתתף סיים סיבוב – יש להוסיף 350 שח ליתרת השחקן
  + יש להפעיל את המתודה play בהתאם לסוג המשבצת במיקום החדש בו נחת.
* לאחר כל סיבוב יודפס המאזן של השחקן ורשימת הנכסים שלו.

המשחק יסתיים כאשר משתתף ביקש לפרוש או אם משתתף אינו יכול לעמוד בתשלום.

* באתר הקורס נמצא קובץ main.cpp לשימושכם.
* באתר הקורס נמצא קובץ monopoly.exe לשימושכם.

**אופן ההגשה**

1. הגשה בזוגות.
2. העבודה תוגש מודפסת ומשודכת באופן נוח לקריאה.  
   1. בעמוד הראשון כותרת העבודה בפורמט הבא:

עבודה ב OOP **31631** סמסטר א' תשע"ט, עבודה 2

תאריך ההגשה: *dd/mm/yyyy*

שם הסטודנט, תעודת זהות, מייל: שם משפחה, שם פרטי, Email

שם הסטודנט, תעודת זהות, מייל: שם משפחה, שם פרטי, Email

ציון:

---------------------------------------------------------------------------------------------

....

....

* 1. תחת הכותרת "מקורות" רשימה של כל המקורות בהם נעזרתם בכתיבת התוכנית
  2. תחת הכותרת "בעיות ידועות" רשימה של הבעיות הידועות בתוכנית (אם קיימות)
  3. בעמוד חדש תדפיס של כל קבצי התוכנית. לכל מחלקה קבצי .h ו- .cpp נפרדים.  
     יש להקפיד על שמות קבצים כפי שמצוין לעיל.  
     על התוכנית להיות כתובה באופן מסודר ומדורג (הזזה ימינה עם פתיחה של כל סוגריים מסולסלים).

יש לתת שמות משתנים משמעותיים.

יש להוסיף הערות הסבר באנגלית בתוכנית.  
המנעו משכפול קוד ומפונקציות ארוכות.  
העברת אוביקטים תעשה by reference ובמידת האפשר const.  
מתודות שיכולות להיות const יש להכריז עליהן ככאלו.

* 1. בעמוד חדש תדפיס את פלט התוכנית עם קובץ הקלט הנתון, כולל בדיקת גלישת זכרון.

1. בנוסף להדפסות יש להגיש את כל קבצי הקוד שכתבתם למטלת ההגשה במודל:
   1. על הקבצים להיות מכווצים בתוכנת zip ולא בשום תוכנת כיווץ אחרת.
   2. שם הקובץ zip יהיה ה ת.ז. של הסטודנטים כך: id1\_id2\_hw2.zip.
   3. שמות הקבצים בהגשה:  
      Board.h  
      Board.cpp  
      Player.h  
      Player.cpp  
      Slot.h  
      Slot.cpp

צוות הקורס שומר לעצמו את הזכות לזמן את הסטודנט/ים לבחינה בעל פה על העבודה שהוגשה.

בהצלחה!