

תרגיל בית מספר 4

נושא: רשימה מקושרת גנרית ושחקני הוקי-קרח פולימורפיים

דדליין: יום א', 04/06/2023, 23:59

הגשה ביחידים

בהצלחה רבה!

תיאור התרגיל

תרגיל זה מהווה הרחבה של תרגיל בית 3 בו מימשתם עולם מונחה עצמים של זירת הוקי-קרח. בתרגיל הנוכחי יתווספו שני מרכיבים לסימולציה הקיימת – ראשית, מבנה נתונים גנרי שיושמש עבור השחקנים, ושנית, טיפוס שחקנים נוספים והתנהגות פולימורפית בתוך הקבוצה. לפיכך, מטרת התרגיל היא כפולה – ראשית, תרגול תכנותי של **תבניות** במימוש מבנה נתונים גנרי (הנושא יילמד בקרוב), ושנית, המשך תכנון ומימוש של עץ הורשה נתון.

חלק א': רשימה מקושרת ממוינת גנרית באמצעות קוד תבניות

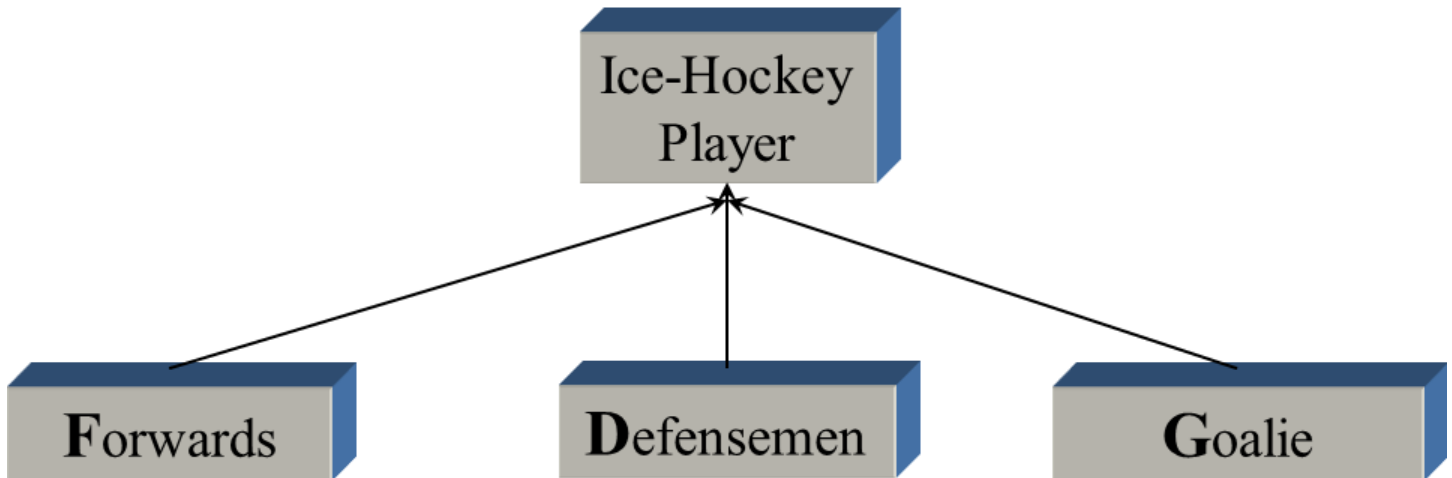
בחלק זה עליכם לממש רשימה מקושרת ממוינת גנרית באמצעות קוד תבניות, תחת דרישה למימוש של המרכיבים הבאים:

- פונקציית הכנסת איבר בשם `insert`; פונקציה זו תכניס איבר באופן ממין, תחת הנחה שאופרטור `>` ממומש עבור טיפוס הנתונים; על המיון להיות בסדר יורד. הכנסת איבר שכבר קיים ברשימה הינה אפשרית, כלומר, תיתכנה כפילויות.
 - פונקציה בשם `first` המחזירה מצביע לאיבר הראשון ברשימה (הטיפוס המוחזר הוא מצביע לפרמטר התבנית)
 - פונקציה בשם `last` המחזירה מצביע לאיבר האחרון ברשימה (הטיפוס המוחזר הוא מצביע לפרמטר התבנית)
 - פונקציית מציאת איבר בשם `findForward`; פונקציה זו תאתר את האיבר המבוקש בחיפוש המתחיל בראש הרשימה, בהסתמך על-כך שאופרטור `==` מומש עבור טיפוס הנתונים, ותחזיר מצביע אליו (אם קיימים מספר עותקים של האיבר, יש להחזיר מצביע לאחד מהם).
 - פונקציית מציאת איבר בשם `findBackward`; פונקציה זו תאתר את האיבר המבוקש בחיפוש המתחיל מזנב הרשימה (הנחות זהות לפונקציית החיפוש הקדמית).
 - פונקציית מחיקת איבר בשם `erase`; בהינתן איבר, פונקציה זו תמחק אותו מן הרשומות בהנחה שהינו קיים ברשימה; אם קיימים מספר עותקים, על כולם להימחק.
 - פונקציית ניקוי הרשימה (מחיקת כל האיברים, כלומר, מעבר לרשימה ריקה) בשם `clear`; **נדרש מימוש ברקורסיה**.
 - פונקציית הדפסה לערוץ הפלט הסטנדרטי, אשר מדפיסה את כל איברי הרשימה בסדר המיון, בשם `print`; סדר הדפסת האיברים הינו סדר המיון היורד. הפונקציה תסתמך על אופרטור הדפסה לפלט הסטנדרטי `<<` של טיפוס הנתונים.
 - שאלתה על מספר האיברים המצויים ברשימה בשם `size`.
- מעבר לאפיון ולדרישות הללו (כולל הדרישה לממש את פונקציית הניקוי באופן רקורסיבי), אתם חופשיים לתכנן ולממש כרצונכם מבנה נתונים זה, כל עוד נשמרת הפונקציונליות של רשימה ממוינת.
- שימו לב, כי אתם רשאים לדרוש תכונות מסוימות שעל האיברים לקיים כדי שיאוחסנו ברשימה הגנרית שלכם. עליכם לפרט בתיעוד את כל הדרישות וההנחות להן אתם נזקקים.

הנושא "מחלקות תבנית" בשפת C++ יילמד בהרצאה הבאה, ביום רביעי 24/05.

בחלק זה של התרגיל אתם תשדרגו את תכנית קבוצת ההוקי אותה מימשתם בתרגיל הקודם לכדי קבוצה בעלת שחקנים פולימורפיים הממלאים תפקידים שונים.

מעתה, שחקני הקבוצה בהן הסימולציה תתחשב הינם שלושה אפשריים: Goalie ('G'), Defensemen ('D'), ו-Forwards ('F'). מבחינת תיאור אובייקטים, שלושתם שייכים לשחקני ההוקי (יורשים מאובייקט כללי "שחקן הוקי-קרח"). עץ הירושה המתאר משפחה זו נתון בסכימה הבאה:



עדכון השחקנים ימשיך להתרחש בכל פעימת זמן בדידה (דור), וישמור על אופיו הסתברותי, אך העדכון עצמו יהיה תלוי בסוג השחקן – כפי שיוסבר כאן בפירוט. תזוזה של כל שחקן בזירה תתרחש לפי משוואת התנועה המקורית, כלומר, עבור שחקן בעל אינדקס i , מיקומו בנקודת זמן $t+1$ ייקבע ע"פ המשוואה הבאה (זהה לתרגיל בית 3):

$$\vec{\ell}_i(t+1) = \vec{\ell}_i(t) + \vec{v}_i(t) \quad (1)$$

ההבדל בין השחקנים השונים יבוא לידי ביטוי בעדכון רכיב המהירות. מהירותו של כל שחקן תמשיך להיות תלויה כמו בתרגיל הקודם בשלושה מרכיבים – בערך הקודם, במוביל הקבוצה ($global\ best$), $\hat{y}(t)$, ובשיא האישי ($personal\ best$), $\vec{y}_i(t)$:

$$\vec{v}_i(t+1) = \alpha \cdot \vec{v}_i(t) + \beta \cdot r_{1i}(t) \cdot (\vec{y}_i(t) - \vec{\ell}_i(t)) + \gamma \cdot r_{2i}(t) \cdot (\hat{y}(t) - \vec{\ell}_i(t)) \quad (2)$$

הערכים של הפרמטרים $\{\alpha, \beta, \gamma\}$ ייקבעו ע"פ סוג השחקן:

- לשוער ("G") יהיו פרמטרים בעלי ערכים $\{\alpha_G = 0.01, \beta_G = -0.1, \gamma_G = -0.25\}$
- לשחקן ההגנה ("D") יהיו פרמטרים בעלי ערכים $\{\alpha_D = 0.05, \beta_D = 0.1, \gamma_D = 0\}$
- לשחקן ההתקפה ("F") יהיו פרמטרים בעלי ערכים $\{\alpha_F = 0.25, \beta_F = 1.0, \gamma_F = 1.0\}$

תכנות בשפת ++C, אביב 2023

שינויים בהשוואה לתרגיל הבית הקודם

- עליכם לעשות שימוש במבנה הנתונים הגנרי אותו מימשתם בחלק א' של התרגיל בכדי לאחסן את שחקני הקבוצה באופן פולימורפי; כלומר, אובייקט הקבוצה מצופה להחזיק את כל השחקנים באמצעות מבנה נתונים יחיד זה.
- על הרשימה להחזיר את השחקנים באופן ממין בהתאם ליחס סדר המוגדר על מרחקם לנקודת המטרה.
- התכנית תקבל גם בתרגיל זה שני קבצי קלט – הראשון יתאר את זירת ההוקי ויהיה בעל פורמט זהה לתרגיל הקודם, והשני יכלול הן את פריסת השחקנים ההתחלתית והן את המהירויות ההתחלתיות שלהם, אך יכלול שינוי בפורמט כדי לאפיין את סוג השחקן. הקובץ יכיל בשורה הראשונה את מספר השחקנים בקבוצה, ובשורות לאחר מכן את סוגם ומצבם ההתחלתי. כל שורה תתחיל בתו המגדיר את טיפוס השחקן (התווים האפשריים הינם '{G','D','F'}), ולאחר מכן תכיל את מיקומו ההתחלתי של השחקן המתאים (באמצעות 2 קואורדינטות) ואת מהירותו ההתחלתית של אותו השחקן (באמצעות 2 רכיבי מהירות), מופרדים ברווחים. להלן דוגמה לקובץ קלט חוקי של איתחול השחקנים:

```
6
G 3 15 0.01 0.01
D 12 7.5 0.35 0.25
D 12 22.5 0.2 -0.2
F 21 7.5 0.3 0.1
F 21 22.5 0.1 -0.1
F 30 15 0.6 0.05
```

בסיום מעברי הזמן, על התכנית להחזיר כפלט את מספר צעדי הזמן של הסימולציה וכן את מיקומי השחקנים של צעד הזמן האחרון באמצעות ערוץ הפלט הסטנדרטי. על נתונים אלו להיות מודפסים בסדר המיון של הרשימה (השחקן הראשון יהיה הקרוב ביותר לנקודת המטרה). למשל, כך עשוי להיראות הפלט עבור הדוגמה שלעיל –

```
96717
D 10.45 7.51
G 3.01 13.96
D 13.87 20.9
F 58.18 14.84
F 58.32 14.44
F 57.99 17.1
```

הנחות עבודה

הנחות העבודה מהתרגיל הקודם לגבי מקרי קצה של מספר הדורות, טיפול בתנאי השפה, וגודל הקבוצה תקפות גם בתרגיל זה. כמו כן, הטיפול בקלט בלתי-חוקי נדרש להיות זהה לתרגיל הקודם.

דגשים

- אין אפשרות להשתמש בספריית STL בתרגיל זה.
- עליכם לוודא כי התכנית עוברת קומפילציית ++g התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המחלקות והפונקציות השונות.
- עליכם לוודא כי התכנית עוברת קומפילציית ++g התואמת את הקומפיילר שעל שרת החוג ללא כל שגיאות או אזהרות כלשהן – לשם כך יש להריץ את הבודק האוטומטי על שרת החוג בטרם ההגשה בכדי לוודא תאימות ונכונות: **hwcheck**.

הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('48'), שם התרגיל ('ex4') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי בפורמט הבא: **48_ex4_studID.zip**
- על ארכיב zip זה להכיל את כל קבצי המקור (ממשק/מימוש) הנדרשים לקומפילציה, והוא רשאי להכיל תיעוד טקסטואלי; מבחינת טיפוס קבצים, עליו לכלול רק קבצים עם סיומות ***.cpp *.h *.txt**
- לדוגמה: על סטודנט/ית בעל/ת מספר זהווי 012345678 להגיש ארכיב בשם **48_ex4_012345678.zip** הכולל את כל קבצי המקור של הפרוייקט, ללא תיקויות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

אי-הקפדה על ההנחיות, כולל פורמט ההגשה הדיגיטלי, תגרור הורדה בציון התרגיל.
לא תתקבלנה הגשות באיחור!