

מטלת סיכום קורס

STAR WARS: EPISODE 0

תאריך הגשה: מוצ"ש, 04/05/2024, 23:59

עבודה ביחידים

תיאור התרגיל

תרגיל זה עוסק במשחק סימולציה תלוי-זמן של "מלחמת הכוכבים", ומיועד לעשות שימוש בפרדיגמת **Model-View-Controller**. אתם נדרשים לתת פתרון מונחה-עצמים בשפת C++ תחת אילוצים לשימוש בתבניות-תכן (*design patterns*). הסימולציה מתארת שיגרת תפעול של האימפריה הגלקטית, באמצעות שינוע של גבישים מתחנות לכוכבי מצודה, תחת אבטחה של חלליות האימפריה כנגד ניסיונות שוד של גורמים המזוהים עם המורדים.

מרחב החלל

חלל האימפריה מוגדר באמצעות מרחב אוקלידי דו-מימדי; יחידת המידה הינה אלף קילומטר ($1000km$), וההנחה היא כי המרחב אינו סופי (אין אכיפה של תנאי שפה, מעבר למגבלות האחסון של הטיפוס float). צפון מוגדר להיות הכיוון החיובי של ציר y (0 מעלות), מזרח הוא הכיוון החיובי של ציר x (90 מעלות), דרום הינו הכיוון השלילי של ציר y (180 מעלות), ומערב הכיוון השלילי של ציר x (270 מעלות). מיקום או יעד של כלי-טיס יוגדר באמצעות צמד קואורדינטות, אך כיוון טיסה יכול להיות מוגדר באמצעות זווית.

סוכן אימפריה (Imperial Agent)

אובייקט סוכן מייצג חייל בדרגה כלשהי הפועל במסגרת האימפריה הגלקטית. בסימולציה שלנו קיימים שלושה סוגי סוכנים: חייל אויב (Midshipman), מפקד (Commander), ואדמירל (Admiral). הסוכנים תמיד מתניידים בכלי-טיס התואמים את דרגתם – חייל מטיס "חללית שינוע" (shuttle), מפקד מטיס "חללית הפצצה" (TIE Bomber), ואדמירל מטיס "חללית השמדה" (Star Destroyer) – שיתוארו בהרחבה.

תחנה (Space Station)

תחנה היא אובייקט סימולציה בעל מיקום קבוע ותכולה של מיכלי גבישים (ללא מגבלת אחסון); לכל תחנה יש כושר ייצור של גבישים כתלות בזמן, והגבישים מועברים באמצעות מיכלים אל "כוכבי המצודה". קיומה נקבע בשלב איתחול הסימולציה.

כוכב מצודה (Fortress Star)

כוכב מצודה הינו אובייקט סימולציה בעל מיקום קבוע ותכולה של מיכלי גבישים (ללא מגבלת אחסון); הגבישים המאוחסנים מתקבלים רק באמצעות פריקה פרטנית של חלליות שינוע. קיום הכוכב נקבע בשלב איתחול הסימולציה.

חללית שינוע (shuttle)

כלי-טיס זה משנע מיכלי גבישים מתחנות לכוכבי מצודה ע"פ פקודות שינוע. בהגיעה לכוכב היעד, החללית תפרוק את מספר המיכלים שבפקודת השינוע ותחנה בכוכב היעד. יצירת החללית מתרחשת במהלך הסימולציה, שם גם ייקבעו המהלכים שלה.

חללית הפצצה (TIE Bomber)

כלי-טיס זה מפטרל בחלל האימפריה בין התחנות וכוכבי המצודה השונים, ובהשלמת מסלול חוזר לנקודת היציאה. יצירת החללית מתרחשת במהלך הסימולציה, שם גם ייקבעו המהלכים שלה.

חללית השמדה (Star Destroyer)

כלי-טיס זה משייט בחלל האימפריה, ויש לו יכולות ירי מדוייק מטווח רחוק. יצירת החללית מתרחשת במהלך הסימולציה, שם גם ייקבעו המהלכים שלה.

חללית מבריהים פלקון (Millennium Falcon)

כלי-טיס זה תוקף ושודד חלליות שינוע; הוא איננו רשאי לחנות בתחנות/כוכבי מצודה. יצירת הפלקון מתרחשת במהלך הסימולציה, שם גם ייקבעו המהלכים שלו.

נושאים מתקדמים בתכנות מונחה עצמים, חורף 2024

מודל (model)

זהו אובייקט יחיד אשר נדרש להיות מוגדר בתבנית Singleton; אחריותו לנהל מעקב אחר עולם הסימולציה על כל היבטיו, החל מניהול הזמן וכלה באחסון האובייקטים. בפרט, עליו להחזיק באמצעות מצביעים את כל כלי הטיס, הסוכנים והאתרים המשתתפים בסימולציה, ולספק שירותי גישה אליהם. בנוסף, האובייקט אחראי לספק שירותי עדכון לאובייקט התצוגה (view).
על בנאי המודל ליצור בכל מקרה כוכב מצודה בשם DS במיקום (40, 10), עם תכולת מיכלים 100000.

תצוגה (view)

אובייקט זה הינו בעל אחריות אחת והיא הצגת מפת חלל האימפריה באמצעות גרפיקה מבוססת-ascii. כל אובייקט מיוצג במפה באמצעות שני תווים הראשונים של שמו. **עדכון המידע מבוסס על אינטראקציה עם המודל.**

יחידת-בקרה (controller)

זהו אובייקט יחיד שאחריותו לנהל את האינטראקציה מול המשתמש, ולנתב את הקלטים המתקבלים עבור המודל. אחריות זו כוללת גם את ניהול השגיאות בקלטי המשתמש.

איפיון מפורט של מרכיבי הסימולציה

מדובר בסימולציה בה צעדי הזמן נשלטים באמצעות פקודות בדידות, כאשר העולם "קופא" בכל צעד זמן, עד למעבר הזמן הבא. מעבר של צעד זמן מתרחש ע"פ פקודה של המשתמש, מייצג שעה עגולה אחת, וכולל עדכון של כל העצמים בעולם הסימולציה.

כלי-טיס (כללי)

אובייקט כלי הטיס הכללי (spaceship) יכול להימצא במצבים הבאים:

- מצב עצירה ("Stopped"); הוא מצבו ההתחלתי של כל כלי-טיס.
- מצב חניה ("Docked") באתר נתון
- מצב סיום ("Dead") במיקום נתון (איפוס כח לאחר מתקפה)
- מצב תנועה אל מיקום / תנועה אל אתר / תנועה אל כיוון ("Moving to...")

כמו כן, הוא מצופה לתמוך בפונקציונליות הבאה:

- עצירה: פקודה זו תבטל את יעדיו של כלי-הטיס ותעדכן את מצבו ל-"Stopped".
- קביעת יעד טיסה: פקודה זו תקבע את יעדו של כלי-הטיס באמצעות צמד קואורדינטות ותעדכן את מצבו ל - "Moving to <destination position>"
- קביעת כיוון טיסה: פקודה זו תקבע את כיוון הטיסה של כלי-הטיס באמצעות זווית ותעדכן את מצבו ל- "Moving on course ...".
- הצגת סטטוס: הדפסת מצב כלי-הטיס הכוללת את שמו, מיקומו, ואת סטטוס השיט שלו (ראו דוגמאות מפורשות בהמשך).

חללית שינוע

חייל האימפריה מטיס את חללית השינוע, אשר איננה חמושה בנשק כלשהו וחשופה למתקפות של מורדים. החללית מקבלת פקודות שינוע הכוללות מקור (אתר תחנה) ויעד (אתר כוכב מצודה). בהינתן פקודת שינוע חוקית (start_supply), החללית תטוס אל התחנה, תעמיס מיכלים במשך שעה אחת, תטוס משם אל אתר היעד, ותפרוק את המיכלים הללו במשך שעה אחת. באם לא ניתנה פקודה נוספת, החללית תישאר ביעד בתום פקודת השינוע.

כלי-טיס זה מאופיין ע"י מצב ההגנה שלו (כל חלליות השינוע מאותחלות עם 10 יחידות כח; הערך המקסימלי הוא 20 יחידות כח). כל חללית שינוע משייטת במהירות קבועה של 500 km/h. החללית יכולה לשאת לכל היותר 5 מיכלים בפקודת שינוע; באם יש בתחנה פחות מ-5 מיכלים, החללית תיקח את כל יתרת המיכלים. השלמת פקודת שינוע באופן מלא תביא לתוספת של יחידת כח אחת למצבה של החללית.

בנוסף לפונקציונליות הנזכרת, החללית מצופה לתמוך בפונקציונליות הבאה:

- יצירת חללית חדשה באמצעות הפרמטרים הבאים:
 - שם החללית (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - שם הטייס (חייל אימפריה)
 - צמד קואורדינטות לציון המיקום ההתחלתי
- סטטוס קונקרטי: בנוסף לשמה, שם הטייס, מיקומה וסטטוס הפעולה שלה, חללית השינוע תציין את מספר המיכלים שהיא נושאת ואת מצב ההגנה שלה (ניתן להוסיף מידע טקסטואלי אחר).
- פקודה בשם start_supply לקביעת תכנית שינוע (ארגומנטים: שני אתרים – שמות תחנה וכוכב מצודה חוקיים).
- רצף של פקודות שינוע חוקיות יתורגם לביצוע הפקודות ברצף זו אחר זו (כלומר, מתן פקודה במהלך ביצוע פקודה אחרת).
- [חללית השינוע אינה תומכת בפקודות התנועה של כלי הטיס הכללי (course/position)].

נושאים מתקדמים בתכנות מונחה עצמים, חורף 2024

חללית הפצצה

מפקד באימפריה מטיס את חללית ההפצצה. החללית מפטרלת בין האתרים השונים באופן אוטומטי. בהינתן פקודת טיסה לאתר מסוים (destination), החללית תטוס לאתר זה, משם תמשיך לאתר הקרוב ביותר אליו, ממנו אל האתר הקרוב ביותר שטרם ביקרה בו, וכך הלאה. במקרה של שוויון מרחקים בין אתרים קרובים שהחללית טרם ביקרה בהם, ייבחר האתר הראשון לפי סדר אלפביתי. מטרתה לבקר בכל אתר פעם אחת בלבד, ולאחר שעשתה זאת תשלים את מסלולה בחזרה אוטומטית לאתר הראשון בו ביקרה ותסיים את הפטרול. חללית זו יכולה גם לקבל פקודות טיסה פרטניות (course/position). כל ביקור של חללית הפצצה באתר אינו אורך זמן. החללית מפטרלת במהירות קבועה של 1000 km/h, והיא מצופה לתמוך בפונקציונליות הבאה:

- יצירת חללית הפצצה חדשה באמצעות הפרמטרים הבאים:
 - שם החללית (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - שם הטייס (קצין אימפריה בדרגת commander)
 - שם אתר לציון המיקום ההתחלתי
- פקודה בשם destination לקביעת אתר היעד (ארגומנט: שם אתר חוקי)
- פקודות בשם course לקביעת כיוון השיוט (ארגומנט: זווית במעלות) וכן position לקביעת יעד הטיסה (ארגומנטים: צמד קואורדינטות).

חללית השמדה

אדמירל באימפריה מטיס את חללית ההשמדה. החללית משייטת בחלל האימפריה בהינתן פקודת טיסה – הן של יעד קונקרטי (destination) והן של כיוון (course/position). החללית משייטת במהירות קבועה של 2000 km/h, והיא נושאת טילים מדויקים שלאחר שיגורם משייטים במהירות של 3000 km/h. היא מצופה לתמוך בפונקציונליות הבאה:

- יצירת חללית השמדה חדשה באמצעות הפרמטרים הבאים:
 - שם החללית (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - שם הטייס (קצין אימפריה בדרגת admiral)
 - צמד קואורדינטות לציון המיקום ההתחלתי
- פקודה בשם destination לקביעת אתר היעד (ארגומנט: שם אתר חוקי)
- פקודות בשם course לקביעת כיוון השיוט (ארגומנט: זווית במעלות) וכן position לקביעת יעד הטיסה (ארגומנטים: צמד קואורדינטות).
- פקודה בשם shoot לשיגור טיל בצעד הזמן הבא (ארגומנט: צמד קואורדינטות); במידה והטיל יפגע במדויק בחללית פלקון, היא תעבור מיידית למצב "Dead".

פלקון מברחים

המורדים והמברחים משייטים בחלל האימפריה, באמצעות חלליות הפלקון, ושודדים שליחי אימפריה המשנעים גבישים. כלי-טיס זה מאופיין ע"י כח התקיפה שלו (איתחול כל הפלקונים הוא 5 יחידות כח; הערך המקסימלי האפשרי הינו 20 יחידות כח). מהירות החללית משתנה, עם ערך מקסימלי של 3000 km/h. אובייקט זה נדרש לתמוך בפעולות הבאות:

- יצירת חללית פלקון חדשה באמצעות הפרמטרים הבאים:
 - שם החללית (מחרוזת אלפביתית עם לכל היותר 16 תווים)
 - צמד קואורדינטות לציון המיקום ההתחלתי
- פקודות בשם course לקביעת כיוון (ארגומנטים: זווית במעלות ומהירות השיוט) וכן position לקביעת יעד הטיסה (ארגומנטים: צמד קואורדינטות ומהירות הטיסה).
- תקיפה: פעולת התקיפה אמורה להיות מכוונת כלפי חללית שינוע אשר נמצאת ברדיוס 100 ק"מ מהפלקון בתחילת זמן הפעולה (צעד הזמן הבא בסימולציה), אחרת הפעולה תסתיים כשילון. אם יעד התקיפה נמצא בטווח, הפעולה מתבצעת מיידית (כלומר בצעד הזמן הנוכחי, שהוא בהכרח בשעה עגולה), ובמהלכה נבדקת נוכחות של חלליות הפצצה בקרבת חללית השינוע המותקפת.

התקיפה תצליח רק אם מתקיימים בו-זמנית שלושה תנאים בזמן ובמיקום התקיפה:

- * חללית הפלקון נמצאת בתוך רדיוס של 100 ק"מ מחללית השינוע
- * כח הפלקון עולה על כח חללית השינוע (מספר יחידות כח הפלקון גדול ממספר יחידות מצב ההגנה של חללית השינוע)
- * לא קיימת חללית הפצצה ברדיוס של 250 ק"מ ממיקום התקיפה

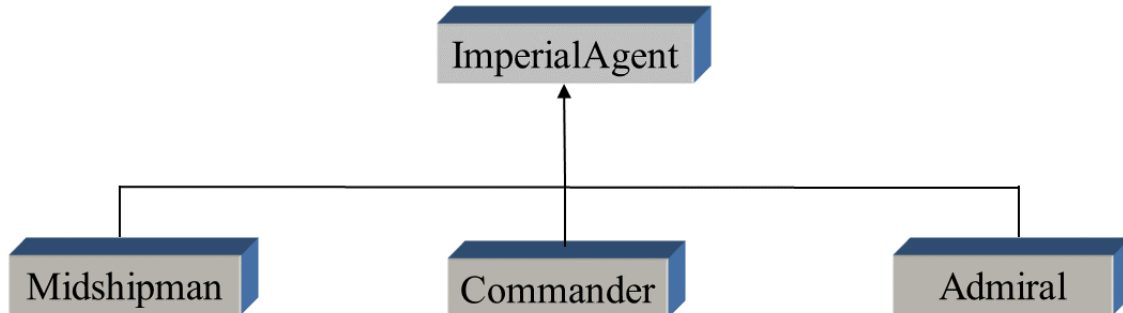
נושאים מתקדמים בתכנות מונחה עצמים, חורף 2024

- במקרה של חללית השינוע, כל מתקפה עליה תביא לאיבוד של יחידת כח אחת ממצב ההגנה שלה. אם התקיפה צלחה, כל הגבישים נשדדים (מספר המיכלים מתאפס), פקודת השינוע שלה תבוטל מיידית, והיא תעבור למצב "Stopped". אם התקיפה כשלה, חללית השינוע תמשיך בהשלמת הפקודה שלה כמתוכנן.
- במקרה של חללית הפלקון – בין אם התקיפה צלחה או לא, הפלקון תעבור למצב "Stopped" במיקום השוד. אם התקיפה צלחה, כוחה של הפלקון יגדל ביחידה אחת; אם התקיפה כשלה, כוחה יקטן ביחידה אחת.
- עבור שני הצדדים המעורבים בתקיפה, באם מצב כוחה של חללית מתאפס בעקבותיה, היא תעבור למצב "Dead".

בכל מקרה, חללית ההפצצה אינה משתתפת באינטראקציה עצמה במידה והיא מצוייה באיזור.

סוכני האימפריה

כאמור, כלי הטיס של האימפריה המשתתפים בסימולציה משוייכים לחיילים בדרגות מסויימות. אמנם סימולציה זו נוסבת בעיקר סביב כלי הטיס, אך יש משמעות גם לקיומם של סוכנים ולכן יש למדל אותם בהתאם לעץ ההורשה הבא:



היות ואובייקטים אלו צפויים לחזור על עצמם בשינויים קלים בתסריטים שונים (למשל, יצירת שלבי משחק שונים, או סימולציות אחרות של חברת התוכנה בהן ישתתפו סוכני האימפריה), הדרישה היא לתמוך ביצירת משפחה כזו של סוכנים באמצעות תבנית התכנות Abstract Factory.

סוכן אימפריה מתאפיין בשמו (מחרוזת אלפביתית עם לכל היותר 16 תווים) ודרגתו הצבאית (ראו את תיאור פקודת create בהמשך).

איפיון המודל

המודל נדרש לאחסן את כל העצמים של עולם הסימולציה.

המודל נושא באחריות לעדכון מצבם של כל האלמנטים בעולם הסימולציה במעברי זמן בידידים של שעה אחת. כלומר, על המודל לבצע את כל החישובים והעדכונים המתרחשים בכל מעבר זמן, אשר קבוע להיות יחידת זמן של שעה אחת.

איפיון יחידת הבקרה

יחידת הבקרה מנהלת את האינטראקציה מול המשתמש באמצעות קריאת פקודות, ובהתאם, מאפשרת שליטה במודל ובתצוגה. עיבוד טקסטואלי של פקודה מתבצע ע"פ הסדר הבא:

המילה הראשונה חייבת להיות המחרוזת exit, שם של כלי-טיס קיים, או פקודת מודל או תצוגה.

- באם המחרוזת הינה exit, על התכנית לסיים את הסימולציה ואת ריצתה בשלב זה.
- באם המחרוזת זהה לשם המלא של כלי-טיס, המילה שלאחריה נדרשת להיות פקודה חוקית שתעובד בהתאם.
- אחרת, המחרוזת נדרשת בהכרח להיות פקודה חוקית עבור המודל או התצוגה.

בכל הפקודות, תנו ההפרדה הסטנדרטי בין הארגומנטים הינו רווח בודד.

פקודות יחידת הבקרה עבור התצוגה:

פקודה בשם default לשחזור פרמטרי ברירת המחדל של המפה (ללא ארגומנטים)

פקודה בשם size לקביעת גודל המפה (ארגומנט יחיד, שלם חיובי s , המגדיר את מספר העמודות והשורות: $30 \geq s > 6$)

פקודה בשם zoom לקביעת קנה-המידה של המפה (ארגומנט יחיד, ממשי חיובי, לקביעת היחס של 1000 ק"מ לתא שטח במפה)

פקודה בשם pan לקיבוע ראשית הצירים של המפה (צמד ארגומנטים ממשיים)

פקודה בשם show לבקשה מיחידת התצוגה לצייר את המפה המעודכנת

נושאים מתקדמים בתכנות מונחה עצמים, חורף 2024

פקודות יחידת הבקרה עבור המודל:

פקודה בשם status בעקבותיה על כל האובייקטים בסימולציה לתאר את מצבם העדכני (ללא ארגומנטים)

פקודה בשם go לעדכון כל האובייקטים ביחידת זמן בדידה של שעה (ללא ארגומנטים)

פקודה בשם create (בעלת שתי הוראות):

1. ליצירת כלי-טיס (חללית שינוע / חללית הפצצה / חללית השמדה / חללית פלקון – עם מספר מותאם של ארגומנטים):

- סוג כלי-הטיס: shuttle / bomber / destroyer / falcon
- שם כלי-הטיס
- שם הטייס עבור חלליות של האימפריה הגלקטית
- ציון המיקום ההתחלתי: שם אתר או צמד קואורדינטות (בסוגריים, מופרדים ביניהם בפסיק)

2. ליצירת סוכן אימפריה:

- דרגה צבאית: midshipman / commander / admiral
- שם הטייס

באם כבר קיים כלי-טיס או סוכן בשם זה, הדבר נחשב לשגיאה.

(הפקודות הבאות יינתנו לאחר שם קיים של כלי-טיס):

פקודה בשם course לקביעת כיוון השיט עבור כלי-טיס נתון (ארגומנט: זווית במעלות; במקרה של falcon, לאחריה תימסר מהירות הטיסה)

פקודה בשם position לקביעת יעד הטיסה עבור כלי נתון (ארגומנטים: צמד קואורדינטות; במקרה של falcon, לאחריהם תימסר מהירות הטיסה)

פקודה בשם destination לקביעת האתר (ארגומנט: שם אתר חוקי)

פקודה בשם attack לקביעת חללית לתקיפה בצעד הזמן הבא (ארגומנט יחיד: שם חללית חוקי)

פקודה בשם shoot לקביעת מטרה לשיגור טיל של חללית השמדה בצעד הזמן הבא (ארגומנט: צמד קואורדינטות)

פקודה בשם stop לעצירת הטיסה (ללא ארגומנטים); **פקודה זו תגרום לביטול כל פקודות הטיסה העתידיות**, באם קיימות

איפיון התצוגה

אובייקט התצוגה מספק פלט מרחבי בתצורת מפה עליו פרושים איברי הסימולציה. התצוגה מורכבת ממטריצה ריבועית, אשר בברירת המחדל ובתחילת הסימולציה מאותחלת להיות בגודל 25X25; (גודלה לפחות 6, ולכל היותר 30).

תא מייצג טווח ערכים של קואורדינטות מרחביות (x,y), כאשר גודלו של כל תא מייצג את קנה-המידה; ברירת המחדל היא 2.0. ראשית הצירים, אשר יכולה להיות כל נקודה מרחבית, מוגדרת להיות הקואורדינטה בפניה השמאלית התחתונה של המפה. מבחינת ייצוג, כל תא במפה מיוצג ע"י שני תווים, כאשר תא ריק מיוצג ע"י נקודה ' '. ולאחר מכן רווח ' ' :

שימו לב כי יחידת המידה המרחבית היא 1000 ק"מ. בעוד שיחידת מידת המהירות של כלי-הטיס היא קמ"ש.

פורמט קלט

התכנית תקבל קלט קובץ אחד ויחיד המתאר את האתרים בחלל האימפריה. לאחר קומפילציה של התכנית לכדי קובץ הרצה בשם simST, כך ניתן יהיה להריץ אותה במקרה של עבודה דרך טרמינל:

```
$ ./simST sites.dat
```

על כל שורה לתאר אתר באמצעות טיפוסו (station/fortress), שמו, מיקומו בחלל (באמצעות צמד קואורדינטות בסוגריים), וכמות מיכלי הגבישים ההתחלית שבו, ויכולת הייצור השעתית (במקרה של תחנה בלבד). להלן דוגמא לתוכן של קובץ אתרים תקין:

נושאים מתקדמים בתכנות מונחה עצמים, חורף 2024

```
station, Yavin, (20.00, 10.00), 100, 5
station, Rakatan, (10.00, 20.00), 20, 3
station, Arcanum, (30.00, 0.00), 50, 4
station, Claw, (20.00, 20.00), 10, 2
station, Facility729D, (10.00, 10.00), 2, 3
station, Drydock4, (0.00, 30.00), 6, 3
```

1. במקרה של קלט לא חוקי (למשל, שגיאת קבצים, הזנת תווים/שדות לא חוקיים, וכיוצא באלה), התכנית תסיים את ריצתה בשלב זה ותדפיס הודעת שגיאה מתאימה לערוץ השגיאות הסטנדרטי.
2. בהנחה שהקלט של המשתמש חוקי, על התכנית לפתוח ערוץ קלט סטנדרטי מול המשתמש לשם קליטת פקודות.

קונסול לקליטת פקודות

היות ומדובר בסימולציה עם מעברי זמן בדידים ונשלטים, על התכנית להדפיס בכל שלב את צעד הזמן הנוכחי. הפניה אל המשתמש בזמן X לקבלת פקודה תתרחש בפורמט הבא:

Time X: Enter command:

ניתן להזין מספר רב של פקודות controller בכל צעד זמן, ורק הפקודה go תביא להתקדמות הזמן (סיום צעד הזמן הנוכחי) תוך חישוב כל העדכונים הנדרשים בשעת המעבר.

להלן רצף חוקי של פקודות קונסול לאחר איתחול התכנית באמצעות קובץ הדוגמא sites.dat :

```
Time 0: Enter command : create admiral Tarkin
Time 0: Enter command : create commander Ozzel
Time 0: Enter command : create midshipman Peter
Time 0: Enter command : create destroyer T0 Tarkin (24.00, 24.00)
Time 0: Enter command : create bomber B2 Ozzel (40.00, 10.00)
Time 0: Enter command : create shuttle GX Peter (0.00, 0.00)
Time 0: Enter command : create falcon Millennium (42.00, 42.00)
Time 0: Enter command : B2 destination Yavin
Time 0: Enter command : GX start_supply Yavin DS
Time 0: Enter command: T0 course 180 20
Time 0: Enter command: Millenniumcourse 270 20
Time 0: Enter command: go
Time 1: Enter command: status

fortress DS at position (40.00, 10.00), Inventory: 100000
station Yavin at position (20.00, 10.00), Inventory: 105
station Rakatan at position (10.00, 20.00), Inventory: 23
station Arcanum at position (30.00, 0.00), Inventory: 54
station Claw at position (20.00, 20.00), Inventory: 12
station Facility729D at position (10.00, 10.00), Inventory: 5
station Drydock4 at position (0.00, 30.00), Inventory: 9
```

נושאים מתקדמים בתכנות מונחה עצמים, חורף 2024

destroyer T0 at (24.00, 22.00), Heading on course 180.00 deg, speed 2000.00km/h

bomber B2 at (39.00, 10.00), Heading to Yavin, speed 1000.00km/h

shuttle GX at (0.45, 0.22), Heading to Yavin, speed 500.00km/h

falcon Millennium at (40.00, 42.00), Heading on course 270.00 deg, speed 2000.00km/h

קובץ טקסטואלי של פקודות אלו, הכולל הצגת מפה באמצעות `show`, נתון לכם כנספח במוודל בשם `.console.txt`. בנוסף, מצורף קובץ טקסטואלי המדגים את פקודות התצוגה עבור אותה סימולציה בזמן 0 - כנספח במוודל בשם `.view_console.txt`.

פונקציית main

התכנית שלכם נדרשת לרוץ בעזרת פונקציית ה-main הבאה (העדכונים המותרים לקובץ הינם רק ברמת ה-include):

main.cpp

```
int main(int argc, char *argv[]) {
    std::cout.precision(2);
    std::cout << std::fixed;

    Controller controller();
    controller.run(argc, argv);
    return 0;
}
```

הנחות

- תנועת כלי-טיס תמיד תתרחש בקו ישר בין נקודות מקור ליעד.
- האלמנטים בסימולציה הינם בעלי גודל אפס, ולפיכך אין להתחשב בתסריט של התנגשות בין עצמים סמוכים במרחב.

אילוצים ודגשים

- תבניות-תכן: אובייקט המודל נדרש להיות סינגלטון בעל גישה גלובלית; יצירת סוכני האימפריה נדרשת להסתמך על תבנית *Abstract Factory*.
- המימוש נדרש להיות בשפת C++ בתקן החדש.
- בתרגיל זה עליכם להשתמש במצביעים החכמים של C++0x בכל שימוש במצביעים.
- יש לבדוק תקינות קלטים; במקרה של אי-תקינות, יש להפעיל מנגנון חריגות ולהציג הודעות שגיאה מתאימות.
- עליכם לשאוף ליעילות ביצועים, ובפרט, לממש *move semantics* עבור מחלקות שעשויות להפיק מכך תועלת.
- עליכם לוודא כי התכנית עוברת קומפילציית `g++ -std=c++11` התואמת את הקומפיילר שעל שרת המכללה ללא כל שגיאות או אזהרות כלשהן, ורצה בהצלחה. לשם כך, ניתן לבדוק קומפילציה עם הבודק האוטומטי בקוד 54.
- עליכם לתעד את הקוד באמצעות הערות המתארות בקצרה את המחלקות והפונקציות השונות.

הגשה

- עליכם להגיש במערכת Moodle קובץ ארכיב מטיפוס zip בלבד, ששמו כולל את קוד הקורס ('54'), שם התרגיל ('ex3') ותעודת הזהות של הסטודנט/ית המגיש/ה, מופרדים בקו תחתי בפורמט הבא: `54_ex3_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 `*.h`, `*.cpp` ו-`*.txt`.
- לדוגמה: על סטודנט שמספר הזיהוי שלו הינו 012345678 להגיש ארכיב בשם `54_ex3_012345678.zip` הכולל את כל קבצי המקור של הפרוייקט, ללא תיקיות כלשהן, ורשאי להכיל קובץ טקסטואלי לתיעוד.

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

Bibliography

1. E. Gamma, R. Helm, R. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995
2. B. Stroustrup, The C++ Programming Language, 4th ed., Addison-Wesley, 2013