שפת ג'אווה ומבוא לפיתוח ווב

טקסס הולדם – חורף 2017

מרצה: **אביעד כהן** [aviadco@mta.ac.il](mailto:aviadco@mta.ac.il)

יועץ מקצועי: **אדון עידן ענטין**

בודק: **ג'ון דו**

תקציר

איך לבלות את החורף בנעימים  
(או: החורף מגיע...)

תוכן העניינים

[דרישות הקורס 4](#_Toc492722076)

[כללי 4](#_Toc492722077)

[איך להגיש תרגילים באיחור, ולהישאר בחיים 5](#_Toc492722078)

[הנחיות כלליות לכתיבת התרגיל 6](#_Toc492722079)

[תרגיל reflection – תרגיל עצמאי ביחידים (5%). מועד הגשה: 25.11.17 8](#_Toc492722080)

[כללי 8](#_Toc492722081)

[מבנה התרגיל 8](#_Toc492722082)

[ניקוד 8](#_Toc492722083)

[איך בודקים ? 8](#_Toc492722084)

[מה מגישים ? 9](#_Toc492722085)

[מקרה בדיקה לדוגמא 9](#_Toc492722086)

[טקסס הולדם – גרסת ה"פוקר" הנפוץ ביותר בימינו ! 11](#_Toc492722087)

[מטרת התרגיל(ים) בקורס 11](#_Toc492722088)

[הוראות הפעלה וחוקי המשחק הבסיסים 11](#_Toc492722089)

[הנחיות ספציפיות למימוש תרגיל "טקסס הולדם" 14](#_Toc492722090)

[קווים כללים ומנחים לעיצוב המימוש 14](#_Toc492722091)

[תרגיל 1 – מימוש טקסס הולדם כאפליקציית Console (15%) - הגשה: 3.12.17 15](#_Toc492722092)

[פרטים יבשים 15](#_Toc492722093)

[דרישות 15](#_Toc492722094)

[בונוסים 19](#_Toc492722095)

[סבבה, סיימתי. מה ואיך להגיש ? 19](#_Toc492722096)

[שאלות ותשובות 20](#_Toc492722097)

[תרגיל 2 – מימוש טקסס הולדם כאפליקציית JavaFX (35%) – הגשה: 7.1.18 21](#_Toc492722098)

[פרטים יבשים 21](#_Toc492722099)

[דרישות 21](#_Toc492722100)

[בונוסים 24](#_Toc492722101)

[סבבה, סיימתי. מה ואיך להגיש ? 24](#_Toc492722102)

[שאלות ותשובות 25](#_Toc492722103)

[תרגיל 3 – מימוש טקסס הולדם כאפליקציית WEB (45%) –הגשה: 25.2.18 26](#_Toc492722104)

[פרטים יבשים 26](#_Toc492722105)

[דרישות 26](#_Toc492722106)

[בונוסים 30](#_Toc492722107)

[סבבה, סיימתי. מה ואיך להגיש ? 30](#_Toc492722108)

[שאלות ותשובות 31](#_Toc492722109)

[נספח א' – דירוג ידיים בפוקר - מן החלש ביותר (10) לחזק ביותר (1) 32](#_Toc492722110)

[נספח ב' – שימוש בספרייה חיצונית לטובת חישוב דירוג ידיים 33](#_Toc492722111)

[נספח ג' – תיאור המשחק באמצעות קובץ XML 34](#_Toc492722112)

[הגדרת סוג משחק - GameType: 34](#_Toc492722113)

[: Structureהגדרת מבנה המשחק - 34](#_Toc492722114)

[מבני נתונים נוספים לסוגי המשחקים השונים 35](#_Toc492722115)

[נספח ד' – קובץ סכמה לוידוא קובץ ה XML 36](#_Toc492722116)

דרישות הקורס

## כללי

1. בקורס אין בחינה אך חובה להגיש תרגילים (סה"כ 4).
2. את מרבית התרגילים (3) ניתן להגיש בזוגות, אך לא בשלישיות / רביעיות / חמישיות או יותר

(כן, גם אם מדובר בשלישיה / רביעיה / חמישיה הצועדת יחדיו לאורך שנים מאז גיל הגן והגישה עד עכשיו את כל הפרוייקטים ביחד).

את תרגיל ה - [reflection](#_כללי) חובה להגיש ביחידים.

1. בעבודה משותפת על תרגיל יש להקפיד על מעורבות אקטיבית של כלל המגישים בכל חלקי התרגיל.
2. במידה והוגדר בונוס לתרגיל מסוים, ציון הבונוס יתווסף לציון התרגיל בלבד (ולא לציון הסופי של הקורס כולו).
3. התרגילים יוגשו דרך מערכת Mama. ייפתח רכיב 'מטלה' נפרד לכל תרגיל.
4. לפני שליחת התרגיל יש לבדוק שהוא עובד ומכיל את הקבצים המעודכנים ביותר, על מערכת "נקייה". בצעו את סט הפעולות שאתם מצפים מן הבודק לבצע וודאו כי הכל מתנהל כראוי וכסדרו.
5. ניתן להחליף את השותפ/ה בכל תרגיל, ללא צורך באישור או הודעה למרצה.

הניקוד על כל תרגיל נזקף לזכות הסטודנטים שבצעו אותו בלבד.

1. יש להעלות את התרגיל רק עבור אחד מבני הזוג ולהוסיף את שם בת/בן הזוג ומספר תעודת הזהות שלה/ו גם באתר וגם בקובץ ה readme המצורף (פרטים בהמשך).

## איך להגיש תרגילים באיחור, ולהישאר בחיים

1. ניתן להגיש תרגיל עד שבוע איחור, כאשר עבור כל 24 שעות איחור - יורדו 2 נקודות מציון התרגיל; תרגיל שיוגש באיחור של יותר משבוע (ללא סיבה מוצדקת) – פשוט לא יבדק.
2. בתרגילים השונים ניתן לממש בונוסים (פרטים בהמשך).

המטרה של הבונוס היא לעזור לכם להעלות את הציון ולא להורידו !

רוצה לאמר: אל תגישו באיחור בשביל להספיק לפתח בונוס.

בונוס מפתחים **אם ורק אם** סיימתם את כל דרישות הבסיס להגשה, יש לכם עוד מספר ימים, וברצונכם לנסות ולהגדיל את הציון ע"י בונוס.

היות וכך ולמען הסר כל ספק: **לא ייבדקו** הבונוסים עבור תרגילים שהוגשו באיחור (שאינו מוצדק).

בהתאם לכך, אני שומר לעצמי חירות רבה יותר בשינוי כזה או אחר של סעיפי הבונוס, גם במהלך התרגיל עצמו.

1. עומס בלימודים, בעבודה, בחיים, בגלל הילדים או ההורים, שכנים וחברים (או בכל תחום אחר) אינו נחשב כסיבה לגיטימית להארכה.
2. במקרה של מחלה או מילואים יש להוסיף סריקה של אישור המחלה או המילואים להגשת התרגיל; בהגשת תרגיל באיחור בשל שירות מילואים יש לצרף להגשת התרגיל קובץ סרוק של טופס השחרור מהמילואים (ולא את צו הקריאה).
3. במקרה של בקשה להארכה (מכל סיבה שהיא, לרבות מילואים ומחלה) יש לפנות למרצה מראש על מנת לקבל אישור (במייל או פרונטלית בכיתה)
4. לאנשי הקבע – הישארות של שבת בבסיס אינה נחשבת כסיבה לגיטימית להארכה (מכיוון שזה חלק מהסדר העבודה בצה"ל); יציאה לאבט"ש כן נחשבת כמילואים ויש להגיש אישור ממפקד הבסיס.
5. תלמיד המגיש תרגיל באיחור של עד שבוע עם סיבה מוצדקת ישלח את הנימוקים לאיחור ביחד עם התרגיל.

## הנחיות כלליות לכתיבת התרגיל

* במהלך הקורס יוצגו דוגמאות והסברים מבוססים על כתיבה בסביבת הפיתוח (IDE) – Intelij IDEA.

אתם מוזמנים (ומעודדים בזאת) לפתח גם כן את התרגיל בסביבת העבודה intelij.

ניתן לקבל רישיון חינם לשימוש בגרסת ultimate, רק בשל היותכם סטודנטים במכללה (כבר שווה !)

יחד עם זאת, כל אחד רשאי לבחור לעבוד בסביבת העבודה הנוחה והמוכרת לו. כך או אחרת הגשת התרגיל אינה כוללת את סביבת הפיתוח אלא אך ורק הרצה ידנית מ cmd (כמו פעם...).

שימו לב: מבחינתכם, לבודק פשוט אין intelij (או כל ide אחר) ולכן זו אפילו לא אופציה. חיסכו ממני (ומכם) את כתיבת המייל המבקש זאת.

* יש להגיש את התרגיל בתור קובץ zip/rar.

הקובץ יכיל:

1. כל הקבצים הרלבנטים להפעלת התרגיל (jar/war – פרטים בגוף התרגיל).
2. קובץ אצווה ( == batch) שיכיל את הפקודה שמריצה את התרגיל.
3. קובץ readme שיכיל את פרטי המגיש/ים, כמו גם הנחיות כלליות להרצה התרגיל וכל הנחות שלקחתם במהלך התרגיל ואתם סבורים שחשוב כי הבודק יכיר. (דמיינו כי בכל שאלה/תקלה שיתקל בהן הבודק, יעמוד לרשותו רק קובץ ה readme שלכם. דאגו להבהיר ולהסביר את כל הדברים שיכולים להשתבש ו/או שבעטיים ייתכנו בעיות/שאלות/תהיות וכיוצב').

כמו כן, כל הנחה שאתם מניחים בעצמכם לגבי אופן מימוש התרגיל (בין אם בלוגיקת התרגיל ובין אם בהנחה טכנולוגית) צריכה להיות רשומה בקובץ.

על קובץ הreadme להיות בפורמט word או pdf (**לא notepad !**).

* דווקא בגלל שאין זהו קורס שבו יכנסו לנבכי הקוד ויבדקו כל שורה ושורה, יש להקפיד ביתר שאת על קוד נקי ומסודר, קריא ויעיל. בפרט:
* הימנעו משכפול קוד
* פונקציות ארוכות מדי (יותר מגודל עמוד)
* בחירת שמות גרועים למחלקות, לפונקציות ולמשתנים
* אינדנטציה נכונה
* imports מיותרים
* יש להקפיד להשתמש ב-modifiers בצורה נבונה:
* מחלקה שלא אמורים לבנות אובייקטים שלה אמורה להיות מוגדרת כ-abstract
* קבועים יש לסמן כ- final
* משתנים של המחלקה רצוי (כמעט תמיד) להגדיר כ-private וכו'.
* יש להקפיד על מוסכמות בסגנון הכתיבה – שמות מחלקות יתחילו באות גדולה, שמות חבילות, משתנים ופונקציות באות קטנה, שמות קבועים יהיו מורכבים רק מאותיות גדולות וכו'.
* התמודדות עם קלט שאינו תקין (במקומות הרלבנטים) הוא חלק בלתי נפרד מחווית המפתח.

יש לוודא קלט תקין מהמשתמש בכל שלב ולהחזיר הודאות שגיאה קריאות, אניפורמטיביות במידה והקלט אינו תקין. (למשל: לא להגיד שהקובץ לא תקין – אלא מה לא תקין בקובץ בצורה מפורטת...)

* יש לדאוג שבאף מקרה לא ייזרק Exception שלא טופל ושיגרום לתעופה של התוכנית; יש לטפל  
  ב-Exceptions בנקודה שבה ניתן לעשות כן. אין להציג למשתמש stack trace !
* הוראות שגויות שייגרמו לאפליקציה שלא לרוץ יורידו נקודות, ולכן רצוי מאוד שתנסו להתקין את האפליקציה בעצמכם לפי ההוראות שתכתבו.
* **זהו תרגיל מתגלגל. המטרה היא לבנות בסיס ראשוני בתרגיל הראשון, ולהמשיך ולהשתמש בו, ככל האפשר (ואפשר !) במהלך התרגילים הבאים. השקיעו חשיבה ותכנון בעיצוב הפתרון תוך מחשבה על איך מה שתעשו היום ישרת אתכם מחר. (זה כלל נכון לחיים, לא רק לתרגיל זה).**
* חלק מהעבודה בתרגילים היא קבלת החלטות בנושאים שאינם מפורטים במדויק. המטרה היא לתרגל את הנושאים המרכזיים הנלמדים בקורס, ולא לתפור מוצר לפי דרישות של לקוח. על כן, בכל מקום שלא מופיעה דרישה מדויקת – מוטל עליכם לבחור בדרך ההגיונית ביותר שנראית לכם ולציין את בחירתכם בקובץ ה Readme אשר מוגש עם התרגיל. אם יש ספק לגבי אופן פעולתכם אתם מעודדים לשאול האם הפתרון שאתם חושבים לתת לסוגיה מסוימת הוא קביל ולגיטימי (שאלות בפורום מייל למרצה וכו)
* **ווידוא הגשת התרגיל טרם הגשתו:**
* **יש לוודא כי ההגשה שלכם רצה היטיב על מע' נקייה, באופן שבו גם הבודק יריץ אותה,** על מערכת נקייה וללא תוצרי לוואי אחרים של הפעלות קודמות שלכם.
  + **הבודק יבצע את הבדיקה על מע' windows 10. כל מי שמפתח על גבי mac/linux – זכותכם – אבל גם חובתכם לוודא כי אתם רצים היטיב על windows 10. למען הסר ספק, לא תתבצע בדיקה על מע' הפעלה אחרת. כמו כן לא תהיה התחשבות בתקלות שמקורם רק בשל עבודה על מע' הפעלה שונות (ולא שאמורות להיות הרבה תקלות כאלה..)**
  + **יש לוודא כי כל קבצי הבדיקה השונים שהועלו ל mama נטענים בהצלחה ע"י המע' שלכם טרם ההגשה. בדיקת הבודק תתחיל מבדיקה בסיסית המבוססת בצורה גסה על קבצים אלה. חבל ליפול Level 0 על שטות שיכולתם לעלות עליה בשנייה עוד בשלב הפיתוח.**
* **בחלק מהתרגילים ניתנת אפשרות למימוש דרישות בונוס.**
  + **ישנם 2 סוגי בונוסים:**
    1. **בונוס בתוך טווח התרגיל - יכול להביא אתכם לכל היותר לציון 100, ולחפות במקרה והורדו לכם נקודות בשל תקלות.**
    2. **בונוס מחוץ לטווח התרגיל – יכול להעלות את ציונכם אף מעבר ל 100 (וכן, יש כפל מבצעים לטובת הלקוח).**
  + **בכל מקרה יש לבצע את הבונוס אם ורק אם סיימתם את כל דרישות הבסיס ההכרחיות לתרגיל.**
  + **חלק מהבונוסים בתרגילים השונים הם כאלה שנועדו "להקדים תרופה למכה" – מימוש דרישה בתרגיל n אשר בכל מקרה תגיע כדרישה חובה בתרגיל n+1.**

**הדבר נועד לעודד אתכם להוריד את העומס הצפוי בתרגיל n+1, מתוך הנחת יסוד שתרגיל n הוא קל יותר ומרווח יותר.**

**תכננו את עבודתכם בהתאם ושאפו "להקדים תרופה למכה", במידת האפשר. (וגם זה כלל חשוב לחיים, בלי קשר לתרגיל ולקורס).**

* + **פירוט הבונוסים, משקלם ונקודותיהם מפורט בגוף התרגיל הספציפי.**
  + **אם כבר מממשים בונוס, יש לממש את כולו, עפ"י דרישתו כדי לזכות במלוא הניקוד שהוא מקנה. בכל מקרה ההחלטה על ניקוד הבונוס היא בידי הבודק/מרצה בלבד (אל דאגה, המגמה היא להיות נדיבים ככל האפשר...)**
  + **כאמור, ולמען הסר כל ספק – ניקוד הבונוס מתווסף לניקוד התרגיל הספציפי שבו הוא מומש ולא לניקוד הסופי של הקורס. לא ניתן לקבל ציון סופי בקורס שהוא מעל ל 100 (גם אם בזכות הבונוסים הגעתם לציון כזה).**

תרגיל reflection – תרגיל עצמאי ביחידים (5%). מועד הגשה: 25.11.17

## כללי

**מועד הגשה: 25.11.17 צורת הגשה: ביחידים בלבד**

בתרגיל זה תזכו להכיר ולתרגל את מנגנון ה reflection בג'אווה.

המצגת מכילה ידע בסיסי המהווה נק' פתיחה בלבד לנושא זה, ולכן כחלק מהתרגיל תדרשו גם ללמוד לבד נושאים נוספים הקשורים לעולם ה reflection שייתכן ואינם מכוסים (או מכוסים כהלכה) במצגת.

שימו לב כי במצגת ישנם 3 נושאים. התרגיל הוא על הנושא הראשון בלבד, של reflection.

מהות התרגיל היא לכתוב investigator שיודע לקבל מופע (instance) של איזהשהוא אובייקט, ואז יודע "לחקור" אותו ולענות על מספר שאלות בהקשרו.

במהלך התרגיל תצטרכו להשתמש אך ורק ביכולת ה reflection של ג'אווה כפי שמוסברות במצגת.

אין להשתמש בשום ספריית צד שלישי כדי לבצע את המטלות השונות !

צפי העבודה על התרגיל, בהינתן שקראתם והבנתם את המצגת, הוא 3-4 שעות.

צפי אורך הקוד שעליכם לכתוב הוא לא יותר מ 250 שורות (אני עשיתי זאת ב 160 שורות מרווחות היטיב..)

הבדיקה לתרגיל תבוצע בצורה אוטומטית, ע"י קוד שיטען את הקובץ שלכם ויריץ את סט הבדיקות שנגזרות ממנו, כלומר יקרא לכל השיטות המוגדרות ב interface שבחרתם, תוך השוואת הערך המוחזר מהקריאה אל ערך מצופה.

## מבנה התרגיל

התרגיל מכיל ממשק בשם Investigator, אותו עליכם לממש. מהות השיטות בממשק הוא לחקור instance של class אחר.

כל שיטה בממשק מתועדת היטיב מבחינת מה היא צריכה לעשות, מה הפרמטרים שהיא מקבלת ומה היא צריכה להחזיר.

חלק מהתרגיל כולל גם התמודדות עם התיעוד והבנה בעזרתו בלבד מה עליכם לעשות בכל שיטה ושיטה.

**הערות חשובות**:

1. שימו לב כי ה class שאתם מממשים חייב להכיל default public constructor !
2. שימו לב שבפרויקט שלכם עליכם למקם את הממשק שקיבלתם (Investigator.java). עליכם למקם אותו בדיוק תחת ה package: reflection.api (ובהתאם לכך גם היררכיית הספריות כמובן).
3. אין לבלוע exceptions או להדפיס אותם. במידה ומתרחשת תקלה או שתזרקו הלאה את ה exception ותוכנית הבדיקה תתמודד איתו בדרכה (הורדת ניקוד על הסעיף המדובר) או שתחזירו ערך כלשהוא כתוצאה המטודה (גם אם הוא לא נכון).

## ניקוד

התרגיל שווה עד 5 נקודות **מהציון הסופי** (!!)

(לא רע ל 3 - 4 שעות עבודה ו 200 שורות קוד...)

## 

## איך בודקים ?

קיבלתם את תוכנית הבדיקה עצמה, אותה אני הולך להפעיל גם בעצמי לטובת בדיקת התרגיל.

התוכנית מקבלת כפרמטר חיצוני את שם הקובץ המקומפל שלכם (.class) ובודקת אותו על מספר מקרי בדיקה.

בעותק התוכנית שקיבלתם כרגע, יש דוגמא למקרה בדיקה פשוט עליו תוכלו לנסות ולהתנסות ולוודא אם אתם בכיוון הנכון או אם לאו.

כדי להפעיל את תוכנית הבדיקה עליכם להקליד ב CMD:

…\> RunTester <your .class file name>

אם הכל עובד כשורה, תקבלו את הפלט הבא:

Test name: Rectangle Class

Testing Rectangle Class basics...

Testing [ getTotalNumberOfMethods ]: expecting answer [6] and got [6]

…

Test Score: 100

## מה מגישים ?

עליכם להגיש קובץ zip, הכולל **בדיוק** 2 קבצים (ו 2 קבצים בלבד !):

1. קובץ התוכנית שלכם בלבד, בגרסתו המקומפלת (.class).
2. קובץ קוד המקור (למקרה של בעיות חמורות בלבד)

שם קובץ הזיפ צריך להכיל את שמכם ואת הת.ז. (אין להגיש קובץ readme בתרגיל זה...)

**מה לא מגישים ?**

1. את קובץ הממשק שקיבלתם
2. תיקיית פרויקט...
3. כל זבל אחר שבמקרה יושב לכם ליד הקוד...

## 

## מקרה בדיקה לדוגמא

במקרה זה מתואר האובייקט rectangle היורש מאובייקט שנקרא polygon.

המימוש הוא חלקי ולא תמיד הגיוני – אז אל תתפסו לקטנות...

מימוש זה כבר מוטמע בתוכנית הבדיקה שקיבלתם ומופע שלו יינתן כקלט לתוכנית שלכם.

**public class** Polygon {  
  
 **private** Set<Point> **points**;  
  
 **public** Polygon() {  
 **points** = **new** HashSet<>();  
 }  
  
 **public int** getTotalPoints() {  
 **return points**.size();  
 }  
  
 **protected void** addPoint(**int** x, **int** y) {  
 **points**.add(**new** Point(x, y));  
 }  
}

**public class** Rectangle **extends** Polygon **implements** Comparable, Serializable {  
  
 **private int x**;  
 **private int y**;  
 **private final int SCALE** = 2;  
  
 **public static void** PRINT\_SOMETHING() {  
 System.***out***.println(**"this is a static method"**);  
 }  
  
 **public** Rectangle() {  
 **x** = -1;  
 **y** = -1;  
 }  
  
 **public** Rectangle(**int** x, **int** y) {  
 **this**.**x** = x;  
 **this**.**y** = y;  
 updateParent();  
 }  
  
 **private void** updateParent() {  
 addPoint(0, 0);  
 addPoint(**x**, 0);  
 addPoint(0, **y**);  
 addPoint(**x**, **y**);  
 }  
  
 **public int** calcArea() {  
 **return x** \* **y**;  
 }  
  
 **public int** calcPerimeter() {  
 **return** twice(**x**) + twice(**y**);  
 }  
  
 **private int** twice(**int** num) {  
 **return** 2 \* num;  
 }  
  
 @Override  
 **public int** compareTo(Object o) {  
  
 **return this**.calcArea() - ((Rectangle)o).calcArea();  
 }  
}

טקסס הולדם – גרסת ה"פוקר" הנפוצה ביותר בימינו !

## מטרת התרגיל(ים) בקורס

חלק א' – מימוש המשחק בתצורתו הבסיסית כאפליקציית Console לשחקן יחיד מול שחקנים ממוחשבים.

חלק ב' – מימוש המשחק כאפליקציה גרפית ב JavaFX בתוספת תמיכה בריבוי משתתפים (עדיין כל השחקנים מול אותו המחשב), כמו כן פיתוח שחקן ממוחשב.

חלק ג' – מימוש המשחק כ web application. הclient יפותח באמצעות html, javascript, css, והסרבר ימומש באמצעות servlets ויופעל באמצעות tomcat. במימוש זה תתבצע תמיכה בריבוי משחקים (כאשר כל משחק הוא מרובה משתתפים). במימוש זה כל שחקן יושב מול מחשבו שלו ואינו חשוף לשחקנים האחרים.

## הוראות הפעלה וחוקי המשחק הבסיסים

טקסס הולדם ("טקסס" בקיצור) הוא משחק קלפים ממשפחת משחקי הפוקר, המבוסס על קלפי קהילה.

בגרסת משחק זו, מלבד הקלפים המצויים בידיו של כל אחד מהשחקנים (2 קלפים סה"כ), מצויים גם קלפים על שולחן המשחק החשופים לכולם, והעומדים לרשות כולם.

המטרה, כמו בכל משחק קלפים מסוג פוקר, היא להצליח "להשיג"/"להחזיק" רצף של 5 קלפים המכיל תבניות שונות של מספרים/צורות.

ככל שהתבנית (רצף) נדירה יותר מבחינה הסתברותית, כך הרצף "חזק" יותר ומנצח רצפים אחרים.

(ראו [נספח א'](#PokerHandRank) לפירוט דירוג הידיים)

את המשחק משחקים עם חבילת קלפים רגילה, ללא ג'וקרים (52 קלפים סה"כ).

המשחק מורכב ממספר סיבובים (נקראים גם "ידיים" או "יד").

כל יד מתחילה מאפס ואין שום קשר (לפחות לא הסתברותי) בין יד אחת למשנתה.

היות ואנו חיים במדינה פרימיטיבית, פוקר עדיין נחשב כמשחק מזל ולכן ככזה הוא אסור עפ"י חוק. על כן (אבל לא רק) נוהגים לשחק פוקר עם "שווי" כסף בדמות צ'יפים (נקראים גם ז'יטונים).

לכל שחקן יש אוסף של צ'יפים בעלי ערכים שונים המהווים את הקופה שלו. כשלשחקן נגמרים הצ'יפים – הוא יוצא מהמשחק (table stacks - אפשר לסכן רק את מה שיש על השולחן).

שחקן יכול לבקש "לקנות" (buy) חבילה נוספת של צ'יפים. בעשותו כך הוא "מעשיר" את קופתו בגובה ה buy (נתון שנקרא מקובץ ה XML). בעולם האמיתי הוא היה משלם כסף אמיתי עבור זכות זו – אבל אצלנו זו פעולה חופשית ששחקן יכול לבצע במהלך המשחק, בין "יד" אחת למשנתה. (לא במהלך משחק יד).

כל יד מורכבת ממספר סבבי הימורים (4), כשלאורך היד יחשפו יותר ויותר קלפי קהילה.

המטרה של כל משתתף היא להרכיב את רצף חמשת קלפי הפוקר הטוב ביותר האפשרי משילוב של הקלפים שבידו וקלפי הקהילה.

פירוט מהלך "יד":

1. חלוקת 2 קלפים לכל שחקן. הקלפים מוסתרים מעיני היריבים וכל שחקן רואה את קלפיו בלבד.
2. ביצוע סבב הימורים ראשוני
3. חלוקת 3 קלפי קהילה (flop)
4. ביצוע סבב הימורים שני
5. חלוקת קלף קהילה נוסף (turn)
6. ביצוע סבב הימורים שלישי
7. חלוקת קלף קהילה נוסף (river)
8. ביצוע סבב הימורים רביעי
9. חשיפת המנצח

בכל סבב הימורים בוחרים השחקנים על כמה הם רוצים להמר בהתאם לכמה הם חושבים כי ה"יד" שהם יכולים להרכיב מקלפיהם הפרטיים ביחד עם קלפי הקהילה - היא היא המנצחת. ההימור מתבצע כסכום מספרי ומבוצע באמצעות הצ'יפים של השחקן. כששחקן מהמר בסיבוב מסויים הימור בגובה X – הדבר מחייב את כל שאר השחקנים להמר גם כן לפחות X או להמר יותר מ X (נניח X+Y) ואז יחוייבו כל שאר השחקנים להתאים את גובה ההימור שלהם ל X+Y.

לאורך היד נצברת קופה ראשית אליה מוכנסים כל ההימורים של כלל השחקנים. בסוף היד הקופה תועבר במלואה למנצח (או תתחלק בין המנצחים במידה ויש תיקו).

אנו נשחק משחק מסוג pot limit – שבו כל שחקן יכול להמר לכל היותר את גובה הקופה שנצברה עד לרגע זה על השולחן.

מי שבוחר שלא להמר את גובה ההימור הנוכחי של הסבב למעשה פורש ויוצא מן היד הנוכחית (רשאי להשתתף בידיים הבאות). כל הצי'פים שהכניס עד לשלב פרישתו לקופה הראשית – הופסדו ונתרים בקופה הראשית.

בכל סבב הימורים, ובמידה ואף שחקן לא הימר עדיין לפניו (כלומר הוא הראשון "לדבר"), שחקן יכול לבחור כי אינו רוצה לפרוש, אולם גם אינו רוצה להמר. פירוש פעולה זו היא שהשחקן רוצה לראות איך מגיבים השחקנים האחרים בסבב ולהגיב בהתאם. היא נקראת בעגה צ'ק – check. במידה ומי מהשחקנים במהלך אותו סבב הכריז על הימור בגובה מסוים – או אז יהיה חייב להחליט השחקן שעשה check אם הוא נענה לגובה ההימור, מעלה אותו, או פורש מהיד.

לשם הפשטות, נימנע ממצב של פיצול קופות (שבו שחקן מהמר את כל גודל קופתו ומעכשיו יש חלק מהקופה הראשית שמוקדש לו וחלק שמוקדש לשאר השחקנים). בהתאם לכך, אם שחקן מהמר את כל הקופה שלו (נניח בגובה X) אזי מעכשיו כל מי שרוצה יכול להשוות ל X בלבד (לא יכול להעלות מעבר לכך) ומרגע שכולם הימרו X – המשחק ממשיך ללא סבבי ההימורים.

במשחק טקסס הולדם נהוג כי אחד השחקנים מחלק את קלפי המשחק והוא נקרא "הדילר" (להלן "המחלק").

בכל יד שחקן אחר מחלק את הקלפים (כדי לשמור על איזון והוגנות וכדי למנוע הונאות קלפים).

הדילר מסומן בכפתור מיוחד שנע בכל יד מקום אחד שמאלה בין השחקנים. השחקן שהכפתור נמצא מולו – הוא הדילר ליד הנוכחית.

היד (סבבי ההימורים) מתנהלת בכיוון מי שיושב לשמאל הדילר (יד שמאל של הדילר).

כדי למנוע מצב שבו כולם "דוגרים על הביצים" ולא מהמרים עד שלא מקבלים "פתיחה" טובה, בטקסס נהוגים מספר הימורי חובה שמחייבים בכל יד מספר שחקנים להמר לפחות פעם אחת בסכום מסוים, בין אם ירצו בכך ובין אם לאו. ההימור הכפוי מתרחש בסבב ההימורים הראשון בלבד, והוא קורה אוטומטית (עבור השחקנים שזה תורם) עוד טרם קבלתם את קלפיהם הפרטיים.

"הימורים כפויים" אלה נקראים blind(s) (שכן השחקנים מחוייבים להמר על "עיוור", ללא קשר לטיב הפתיחה שלהם). ישנם 2 הימורים כפויים: big blind ו small blind (קרויים בקיצור big ו small). על מנת שהמשחק יהיה כמה שיותר מאוזן בין כלל שחקניו, בכל יד משתנה זהות השחקנים שמחוייבים לתת big ו small והסדר נע עפ"י מיקום ישיבתם בשולחן, יחסית לדילר.

השחקן היושב משמאל לדילר (יד שמאל של הדילר) הוא השחקן שמחויב לשים small והשחקן שמשמאלו (2 מקומות משמאל לדילר) מחויב לשים big. כך, בכל סיבוב, משתנה זהות המחלק, וכתוצאה מכך גם זהות השחקנים המחוייבים בהימור כפוי.

בסבב ההימורים הראשון בלבד, הראשון ש"מדבר" (=מהמר) הוא השחקן היושב משמאל ל big. בשאר סבבי ההימורים הראשון שמדבר הוא השחקן היושב משמאל לדילר.

בכל סבב הימורים, לכל שחקן ישנם מספר אפשרויות תגובה:

1. F (**F**old) – השחקן מחליט לפרוש מ"היד" הנוכחית.
2. B (**B**et) – מתבצע ע"י השחקן הראשון להמר בסבב. השחקן מחליט על סכום ההימור שהוא מוכן לסכן. הוא עושה זאת ע"י הכנסת מספר המהווה את גובה הימור הרצוי.
3. C (**C**all) – השחקן שתורו להמר בוחר להיכנס לסבב ההימורים ע"י כך שהוא מוכן להמר את הסכום הנדרש. הוא אינו מנסה להעלות את סכום ההימור – אלא רק להשוות לסכום הנוכחי. בפעולה זו אין צורך להכריז מה גובה ההימור – שכן ברור כי הוא רק "משווה" לגובה ההימור הנוכחי.
4. K (chec**K**) – השחקן מבצע check ונותן צ'אנס לשחקן הבא אחריו להמר. שימו לב ש check ניתן לעשות כל עוד **אף** שחקן לא הימר לפניך. מכאן נובע על כן, כי check לא ניתן לעשות בסבב ההימורים הראשון בגלל הבליינדים.
5. R (**R**aise) – השחקן רוצה להשוות את גובה ההימור הנוכחי ואף להעלותו. העלאה היא לכל היותר בגובה הקופה הנוכחית שנצברה על השולחן עד לרגע ההימור וכמובן לכל היותר כגובה קופתו של השחקן (מינימום בין השניים). יש לציין רק את גובה ההעלאה כשאוטומטית יתווסף לה גובה ההימור הנוכחי.

להלן מהלך מפורט של יד:

1. הדילר מחלק קלפים לכל השחקנים. במציאות יש נוהג לאופן וסדר חלוקת הקלפים לשחקנים. במשחק נתעלם מנוהג זה. כל שחקן מקבל תצוגה של 2 הקלפים שהוא קיבל מהדילר

(שימו לב זה משתנה בהתאם לשכבת ה UI הרלבנטית. פרטים בגוף התרגיל).

השחקנים שמסומנים כ big ו small "מהמרים" את ה big וה small בצורה אוטומטית בלי קשר לקלפיהם.

1. מתבצע סבב הימורים ראשון, החל מהשחקן המופיע לשמאל ה big.

סבב הימורים זה חייב להגיע (לפחות) אל ה small שיתבקש להוסיף (לפחות) את ההפרש שבין ה small ל big (כל זאת בהנחה שאף אחד לא עשה R לפני כן).

כל עוד מישהו עושה R, הסבב ממשיך עד שכל השחקנים משווים (C) או פורשים (F)

1. הדילר מחלק את ה flop – 3 קלפי קהילה במרכז השולחן (נתעלם מהנוהג של "שריפת" קלף ראשון)
2. סבב הימורים שני. הראשון "לדבר" הוא השחקן משמאל לדילר (ה small)
3. הדילר מחלק את ה turn – קלף קהילה נוסף במרכז השולחן.
4. סבב הימורים שלישי. הראשון "לדבר" הוא השחקן משמאל לדילר (ה small)
5. הדילר מחלק את ה river – קלף קהילה נוסף (ואחרון) במרכז השולחן
6. סבב הימורים רביעי. הראשון "לדבר" הוא השחקן משמאל לדילר (ה small)
7. בגמר סבב ההימורים הרביעי נחשפים **כל** קלפי השחקנים (כל שחקן באזור שלו) ומוכרז המנצח – השחקן שהצליח להרכיב את ה"יד" הטובה ביותר עפ"י דירוג ידי הפוקר. השחקן הנ"ל גורף לקופתו הפרטית את כל הקופה שנצברה במהלך ה"יד". שימו לב כי במשחק אמיתי אין הכרח כי כל שחקן יחשוף את קלפיו. לשם השפטות נתעלם מנוהל זה).

נקודות לשים לב אליהם:

* במידה ובמהלך היד שחקן עושה raise, הרי שסבב ההימורים נמשך עד אשר לכל השחקנים תינתן ההזדמנות להשוות (או להעלות) את גובה ההימור או לפרוש מן המשחק.
* במידה ובמהלך היד מי מהשחקנים ביצע all in (הימר על כל גובה הקופה שלו, במידת האפשר), הרי שמרגע שכלל השחקנים משווים את ההימור שלו, אין טעם למהשיך בסבבי ההימורים (שכן השחקן אינו יכול להמר עוד), ולכן המשחק מתקדם אל סיומו תוך דילוג על סבבי ההימורים.
* במידה ויש כמה שחקנים בסוף שיש להם את אותו דירוג היד (יכול לקרות), אזי הם מתחלקים בגובה הקופה של ה"יד" בינהם. במידה והקופה "לא מתחלקת יפה" – השארית תישאר ל"יד" הבאה לרווחת כולם.
* במידה וכל השחקנים פרשו (F) במהלך ה"יד" – השחקן האחרון שנותר מנצח טכנית ומיידית ומקבל את כל הקופה שנצברה עד לרגע זה.

## הנחיות ספציפיות למימוש תרגיל "טקסס הולדם"

1. **המשחק ישוחק ברמה וריאציות, בהתאם ליכולות ממשק ה UI בכל תרגיל. כל המידע על סוג המשחק והנתונים הנדרשים לו יגיעו מקובץ XML, שיינתן כקלט במהלך התוכנית.**
2. **עליכם להיות מסוגלים לעקוב ולהציג בכל רגע נתון את מיקומו של השחקן יחסית לשאר השחקנים בשולחן. כמו גם נתונים סטטיסטים (במקומות בהם זה רלבנטי).**
3. **במהלך התרגילים תממשו גם שחקן ממוחשב.**

**שימו לב: זהו אינו קורס בבינה מלאכותית, ולא תיבחנו על איכותו ובינתו של השחקן הממוחשב.**

**השחקן הממוחשב יכול להיות "טיפש" – הוא בסה"כ צריך לשחק עפ"י הכללים. פעולותיו של השחקן הממוחשב מתבצעות מאחורי הקלעים. ורק תוצאתם מוצגת בסוף בממשק ה UI.**

1. **כל הקלטים והפלטים והאינטרקציה עם המשתמש תתבצע באנגלית בלבד, ומשמאל לימין (היכן שזה רלבנטי בממשקי ה UI השונים).**
2. **כחלק ממהלך ה"יד" תצטרכו לדעת להכריע איזו יד מנצחת ועם איזה רצף (דירוג) מתוך 7 קלפים: 5 קלפי קהילה ו 2 קלפי שחקן פרטיים.**

**לשם כך תוכלו להשתמש בספרייה צד שלישי (מסופקת) המאפשרת לבצע את החישובים הנ"ל וחוסכת לכם את העבודה הסיזיפית (והמאוד קשה !) הזו. אני ממליץ בחום שלא לנסות ולבזבז את הזמן היקר שאין לכם בתרגיל ולממש בעצמכם יכולת טכנית זו.**

**(ראו** [נספח ב](#thirdParty)**' להנחיות נוספות בנושא זה)**

## קווים כללים ומנחים לעיצוב המימוש

1. **המטרה העקרית היא לפתח רכיב שיהווה את מנוע המשחק ורכיבים נוספים (בכל תרגיל) המהווים ממשקי משתמש שונים שיתנהלו מעל (אל מול) מנוע המשחק.**

**מנוע המשחק יכיל את כל המידע הלוגי לגבי המשחק(ים) שמנוהל כעת, ניקוד, היסטוריית מהלכים וכו'.**

1. **אפשר לאמר כי כל תרגיל מגדיר שכבת ממשק שונה מהותית של המשחק אולם כל תרגיל מתנהל בסופו של דבר מול מנוע המשחק האחד והיחיד. על כן כדאי ורצוי עד מאוד שלמן ההתחלה מבנה הקוד והפרויקט ישקף נושא זה – כלומר יהיה לכם מודול של מנוע המשחק, שמתקיים "עצמאית" ולידו מודולים נוספים של הממשקים השונים (console, javafx, web) שיודעים "לדבר" עם מנוע המשחק. פרטים על מימוש שכזה כמובן יינתנו במהלך הקורס.**
2. **חישבו על האופן שבו תתארו משחק המורכב מכמה ידיים, המורכבות מכמה סבבי הימורים. מהם המחלקות העיקריות ? מי קשור למי ומי מכיל את מי ?**
3. **במהלך התרגילים תפתחו 2 סוגי שחקנים: אנושי וממוחשב. חישבו על הזהה ועל השונה שבינהם וממשו כך שיהיה זה שקוף וחלק ביותר לדעת/לבחור באיזה שחקן מדובר, מבחינת ניהול הקוד שלכם.**

תרגיל 1 – מימוש טקסס הולדם כאפליקציית Console (15%) - הגשה: 3.12.17

## פרטים יבשים

צפי תחילת עבודה: **12.11.17** תאריך הגשה: **3.12.17**

צפי זמן לביצוע: **3 שבועות** ציון אפשרי מקסימלי: **106**

משקל התרגיל: **15%** קושי: **סביר**

**מטרת התרגיל העיקרית**

1. הקמת תשתית לניהול המשחק

(שתשרת את התרגילים הבאים)

1. מימוש שחקן ממוחשב
2. מימוש ממשק משתמש כאפליצקיית Console פשוטה

## דרישות

1. בתרגיל זה תממשו את טקסס הולדם בגרסתו הבסיסית (GameType = "Basic").

ה blinds נותרים קבועים ואינם משתנים לאורך המשחק.

1. בגרסה זו ישתתפו בדיוק שחקן אנושי אחד, המשחק מול 3 שחקנים ממוחשבים (בדיוק. בלי שום יוצא מן הכלל.)
2. בכל שלב במשחק יוצגו למשתמש רשימת פקודות אפשריות, שבאמצעותן הוא ינווט את דרכו במשחק (פירוט בהמשך)

שימו לב, חלק מהפקודות דורשות קלטים שונים מן המשתמש. יש לדאוג להבהיר בכל רגע נתון מהם הקלטים הנדרשים מהמשתמש בכל פקודה, ומהו האופן והפורמט בו מצפים לקבל אותם.

1. יש להציג ולצייר למשתמש את השחקנים במשחק. כאמור, יש חשיבות לסדר השחקנים ולמצבם הייחודי במהלך המשחק (big, small, diller). היות וזוהי אפליקציית console, ולשם הפשטות, ניתן לסרטט את השחקנים השונים על המסך כאילו היו 4 קודקדוים של ריבוע. (דוגמה בהמשך).
2. היות וזהו תרגיל console, כל קלף במשחק יוצג באמצעות 2 תווים.

תו ראשון הוא מספר הקלף (,T,J,Q,K,A2,3,4,5,6,7,8,9)

תו שני הוא הצורה/סדרה (נקרא גם suite) של הקלף: לב – **H**eart, יהלום – **D**iamond, תלתן - **C**lub, עלה – **S**pade

שימו לב כי אין צורך לתאר את צבע הקלף: לב ויהלום הם אדומים (תמיד) ועלה ותלתן הם שחורים (תמיד).

כך לדוגמא:

הרצף הבא מתאר יד של שחקן שקיבל נסיך לב, ושמונה עלה:

8S, JH

הרצף הבא מתאר flop של אס יהלום, 2 תלתן ו 8 לב

AD, 2C, 8H

שימו לב כי האותיות ב capital case בלבד.

1. בתרגיל זה עליכם לממש שחקן ממוחשב. זהו אינו קורס בינה מלאכותית והשחקן הממוחשב אינו נדרש להיות חכם או בעל בינה מלאכותית משום צורה שהיא. הוא יכול להיות טמבל מושלם וצריך אך ורק לשחק על פי הכללים. פועלו של השחקן הממוחשב מגיע לידי ביטוי אך ורק במהלך שיחוק "יד" (כחלק מפקודה 4, מפורטת בהמשך), והוא אינו יכול לבחור או להיות פעיל באף אחת מהפקודות האחרות שאינן כחלק מהתנהלות "יד".
2. ישנן פקודות שאין הגיון לבצע אותן אם לא קדמו להם פקודות אחרות. במידה וזה קורה יש להציג הודעת שגיאה רלבנטית למשתמש ולאפשר את המשך מהלך המשחק.
3. נתוני המשחק נמצאים בתוך קובץ XML, אותו תקבלו כקלט מהמשתמש. קובץ ה XML מכיל נתוני משחק מגוונים, כגון גודל ה buy, small, big וכו' (פרטים ב[נספח ג'](#_הגדרת_סוג_משחק))
4. יש לבצע בדיקת תקינות לקלט ולהגיב בצורה נאותה לקלט שאינו תקין.

בדיקות התקינות שעליכם לבצע בהקשר של טעינת קובץ המשחק הינם:

* 1. הקובץ קיים, והוא מסוג XML (די לבדוק לשם כך כי הוא נגמר בסיומת .xml)
  2. כשטוענים קובץ חדש, יש לוודא כי אין משחק שכבר מתנהל במע' (אפשרי גם ע"י חסימת פקודה מס' 1 מלהתבצע במידה ומשחק פעיל כבר קיים במע').
  3. Big חייב להיות גדול מ small
  4. Hands count חייב להיות **גדול או שווה** למספר השחקנים (בתרגיל 1 – פירוש הדבר גדול או שווה ל 4) וחייב להיות **כפולה שלמה שלהם** ( = להתחלק ללא שארית) כך שלכל שחקן יהיה סיכוי שווה לחוות את את כל התפקידים באותה כמות פעמים.
  5. בדיקת תקינות קלט כחלק מהאינטרקציה עם המשתמש – אם אתם מצפים לקבל מספר – לא לקרוס כי הכניסו לכם בטעות (או בכוונה) טקסט.

1. **אין צורך** להשתמש בצבעים שונים במהלך תרגיל זה בעת ההדפסה ל console.

כמו כן **אין צורך** לנקות את המסך בין פקודה לפקודה.

יתרה מזאת, ישנו צפי רב (ניסיון מהסמסטרים הקודמים) כי ניסיון לעשות כן תוך שימוש בספריות צד שלישי קורס אצל הבודק

ומעוות את כל תצוגת המסך וגורם לחוסר יכולת לבדוק את ההגשה. גם אם בדקתם את זה אצלכם וזה עבד. גם אם בדקתם

במחשב של השכנה וזה עבד.

1. כאשר שחקן אנושי פורש במהלך יד, היד הנוכחית מסתיימת. השחקנים הממוחשבים שעוד נמצאים במשחק מתחלקים שווה בשווה בגובה הקופה שנצברה עד כה. במידה והחלוקה אינה יוצאת זהה לכולם – השארית תישמר בקופה הראשית כספתח ליד הבאה בתור. במילים אחרות אין צורך לתמוך במצב שבו מתנהל משחק עם שחקנים ממוחשבים בלבד.
2. המשחק כולו מסתיים כאשר תמו כל הידיים שהוקצו לו (נתון שמגיע מקובץ ה XML), או אם השחקן האנושי החליט לפרוש באופן יזום ("לקום מהשולחן"). בשני המקרים, בגמר המשחק יוצגו סטטיסטיקות מתאימות לגבי כל השחקנים (מפורטות בהמשך)
3. הפקודות שיש לתמוך בהן (סדר הפקודות מחייב):
4. קריאת קובץ פרטי משחק מהמשתמש.

פקודה זו טוענת "הגדרת משחק" למערכת. בהמשך אפשר יהיה לבחור ממש לשחק בו.

הקובץ יהיה קובץ XML שפרטיו מובאים [בנספח](#_הגדרת_סוג_משחק) לתרגיל זה.

(שימו לב כי קובץ ה xml מכיל פרטים נוספים שאין להם חשיבות לתרגיל זה אלא רק לתרגילים הבאים וזאת כדי לפשט את התהליך בו אתם משתמשים בקובץ).

יש לוודא תקינות הנתונים המופיעים בקובץ כפי שהוגדרו בסעיף 9.

(כחלק מבדיקת התרגיל יטענו למערכת קבצים שאינם חוקיים כדי לבדוק מהי התגובה).

כחלק מפקודה זו תייצרו ותציגו את שולחן השחקנים.

מקומות השחקנים מוגרלים, כמו גם השחקן שנבחר להיות הדילר (ובהתאם לכך ה small וה big).

כל שחקן מקבל קופה בגובה buy אחד בתחילת המשחק.

הפלט של פקודה זו צריך להיות כפי שמתואר בפקודה מס' 3.

שימו לב ניתן לבצע את פקודה מס' 1 כמה פעמים רצופות שרוצים, כל עוד לא בוצעה פקודה מס' 2 (ראו בהמשך).

1. התחלת משחק

הפקודה מאתחלת את המשחק ומציגה לשחקנים את מצב המשחק. (כפי שמתואר בפקודה 3) מרגע שפקודה זו בוצעה בהצלחה, אין לאפשר הרצה חוזרת של פקודות 1 ו 2 (שכן המשחק כבר מתנהל...).

אין לאפשר בחירה של פקודה מס' 2 כשאין שום קובץ טעון במע' (יש להציג הודעת שגיאה במקרה זה).

כשבוחרים את פקודה מס' 2 היא פועלת בהקשר הקובץ האחרון שנטען למערכת (במקרה ובו היו כמה הפעלות של פקודה מס' 1...)

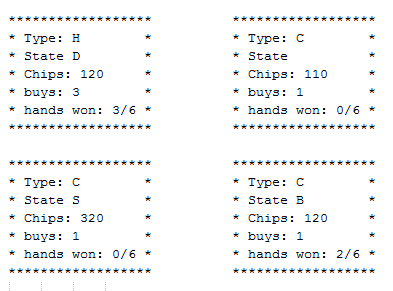
1. הצגת מצב המשחק

לכל שחקן יש לייצר "אזור" על המסך שבו יוצגו כלל הפרטים בהקשרו.

האזורים צריכים להיות בפורמט של שולחן המשחק – 4 קודקודים של ריבוע.

פרטים אלו כוללים:

1. סטטוס השחקן – D\S\B - דילר, big, small – ליד small\big יש להדפיס את גובה הסכום המדובר (big\small)
2. סוג השחקן – C\H
3. כמות הצ'יפים שבקופתו (מספר סופי).
4. כמות ה buy'ים שהשחקן ביצע.
5. כמות הידיים שניצח מתוך כמות הידיים ששוחקו עד עכשיו

הנה דוגמא אפשרית לאיך להציג את המידע:

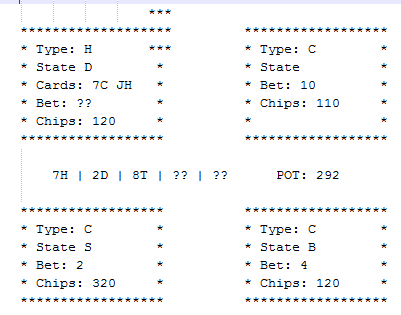
1. שיחוק יד אחת

ברגע שמתחילים לשחק יד – המשחק מתנהל רצוף לאורך כל היד.

כדי להציג את תוכן היד בשלבים השונים של המשחק, יש להציג את הפרטים הבאים:

1. סטטוס השחקן – D\S\B - דילר, big, small (כמו כן להדפיס את גובה ה big/small לשחקנים הרלבנטים)
2. סוג השחקן – C\H – ממוחשב, אנושי
   1. אם השחקן אנושי יש להציג גם את סימני הקלפים שקיבל
3. גובה ההימור הנוכחי
4. כמות הצ'יפים שנותרו בקופתו (לאחר מתן גובה ההימור הנוכחי)
5. סימון המציג את תור השחקן הנוכחי
6. במרכז השולחן יש להציג את 5 קלפי הקהילה שייפתחו לאורך היד.
7. ליד קלפי הקהילה יש להציג את גובה הקופה הנוכחית של היד.

למידע זה נקרא הצגת "**מצב היד**".

הנה דוגמא אפשרית לאיך להציג את המידע. התמונה מתארת מצב אחרי ה flop, כשתור הדילר להמר:

תהליך המשחק יתרחש כמתואר בתיאור הכללי של המשחק.

הנה כמה דגשים הרלבנטים לממשק console:

* השחקן האנושי היחיד מקבל תצוגה של 2 הקלפים שהוא קיבל מהדילר.
* בכל סבב הימורים על השחקן לבחור אחת מ 5 האפשורויות שמתוארות מעלה. שימו לב שיש אפשרויות שדורשות אחריהן בקשת קלט נוספת (למשל R דורש הכנסת גובה ההעלאה). במידה ומתבצעים טעויות מצד השחקן יש להתריע על כך בצורה מסודרת ונאותה (לא stack trace או כישלון כזה או אחר). דוגמה לכך למשל כששחקן מנסה לעשות R עם יותר מגובה קופתו הפרטית או כל מה שנצבר עד כה על השולחן.
* בגמר סבב הימורים (כל אחד מהם) ו**לפני** חשיפת הקלפים הבאים (flop, turn, river), יש לעצור ולבקש את אישורו של השחקן האנושי לפני המשך ה"יד", כמו גם להציג את מצב היד עד עכשיו.
* בכל שלב שבו השחקנים הממוחשבים נדרשים לבצע את מהלכם – הם מבצעים אותו מאחורי הקלעים ורק תוצאותיו מוצגות בפני המשתמש. אם יש רצף של כמה שחקנים ממוחשבים אחד אחרי השני (ובהכרח יש רצף שכזה) הרי שכולם מבצעים את מהליכיהם מאחורי הקלעים והתוצאות מוצגות בסוף כמקשה אחת לשחקן האנושי עת מגיע תורו.
* בגמר סבב ההימורים האחרון ולפני חשיפת המנצח וקלפי השחקנים – יש לחכות גם כן לאישורו של השחקן האנושי (היחיד) במשחק. המנצח צריך להיות מוכתר בצורה פורמלית ומלאה תוך הצגת דירוג היד שלו, וסכום הזכייה. במידה ומדובר בכמה זוכים (תיקו) יש לפרט את המידע הנ"ל על כל אחד מהזוכים.

לאחר שלב זה נגמרה ה"יד" וחוזרים לתפריט הראשי.

כפתור הדילר יזוז שחקן אחד שמאלה (ובהתאם לכך גם מיקומי ה small וה big).

1. קבלת סטטיסטיקות

פקודה זו תחזיר בכל רגע נתון נתונים לגבי כלל המשחק.

נתונים כללים שיש להציג:

1. כמה זמן עבר מאז תחילת המשחק (מאז פקודה מס' 2) – יש לתת את הנתון ב שניות : דקות בלבד, למשל 76:49
2. כמה ידיים שוחקו עד עכשיו מתוך כמה סה"כ ידיים עד לתום המשחק.
3. מה גודל הקופה המקסימלי האפשרי לכל המשחק עד עכשיו (סה"כ ה buy'ים שבוצעו עד עכשיו של כל השחקנים).
4. תוכן של פקודה מס' 3
5. Buy נוסף

פקודה זו מאפשרת לשחקן לרכוש כמות צ'יפים נוספת. אל קופתו יתווסף גובה ה buy (כפי שמוגדר בקובץ ה XML) כמו גם לגובה הקופה המקסימלי האפשרי במשחק.

1. פרישה/סיום מהמשחק

שחקן יכול לבקש לפרוש גם טרם סיומו הפורמלי של המשחק. בגמר פקודה זו יוצגו נתוני פקודה 5.

אין צורך לתמוך (בשלב זה) בהמשך משחק נוסף לאחר סיום משחק אחד (אלא אם אתם מבצעים את בונוס מס' 1)

## בונוסים

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | סוג | מהות | למה שווה לי ? | כמה שווה לי ? |
| 1 | הגדלת ראש מינורית | לאפשר את המשך הרצת התוכנית לאחר פקודה מס' 6. ברגע שמשתמש בוחר לסיים את המשחק – הוא חוזר למצב ההתחלתי המאפשר לו לטעון משחק חדש, או להתחיל מחדש את המשחק הנוכחי.  במידה ומממשים בונוס זה יש להוסיף פקודה נוספת (נגיד 8..) שמאפשרת סיום אמיתי וסופי של התוכנית ויציאה מסודרת ממנה. | זה מה-זה בקטנה שחבל... | 2 נקודות  (ועד ל 100) |
| 2 | הגדלת ראש נאה | טעינה ושמירה של משחק.  אפשרו לשמור משחק, ולטעון אותו בהמשך (יש להוסיף פקודות ייעודיות לשם כך).  ברגע ששומרים משחק יש לבחור שם של קובץ יעד ולשמור לתוכו את המשחק. המשחק יכול להמשיך כרגיל לאחר נק' השמירה.  ברגע שטוענים משחק – הוא ממשיך בדיוק מהפוזיציה בה הוא נשמר בפעם האחרונה.  השמירה והטעינה של המשחק יכולה להתבצע לאיזה פורמט קובץ שאתם חפצים – החלטה שלכם.  יש להוסיף לתפריט הראשי פקודות מתאימות לביצוע שמירה וטעינה של משחק.  שמירה של משחק ניתן לבצע רק בין "ידיים" ולא במהלך ביצוע "יד". | מגניב לאללה... | 6 נקודות  (**מעל ל 100**) |

## סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. jar (אחד או יותר) שהוא כל הקוד שלכם, בצירוף קובץ אצווה (batch) שהפעלתו תריץ את התוכנית

(כלומר תבצע java -jar <class name> וכו').

1. קבצי קוד המקור של הפרויקט שלכם.
2. קובץ ההגשה יכיל גם קובץ **readme** שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, כמו גם כל דבר נוסף העולה על דעתכם שחשוב שהבודקת תדע.
3. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות ותפקידם.
4. יש לכלול בקובץ ה **readme** גם פירוט של המגישים שם, ת.ז. ואי מייל זמין ורלבנטי (!!) – במידה ויהיה צורך ליצור קשר.
5. במידה ומימשתם את מי מהבונוסים לעיל, ציינו זאת בקובץ ה readme כדי שהבודקת תדע לבדוק בהתאם.

בונוס שימומש אבל לא יתועד – לא ייבדק !

## שאלות ותשובות

|  |  |  |  |
| --- | --- | --- | --- |
| # | תאריך | שאלה | תשובה |
| 1 | 27.11.17 | האם יש להדפיס את גובה ה Big וה Small כחלק מהתוכן של פקודה מס' 3 ו 4 ? | כן צריך לעשות זאת. אפשר כמספר שמופיע בסוגריים ליד האות B\S ב State של כל שחקן רלבנטי. כך תהיה נראות של הנתון הנ"ל גם במעמד קריאת קובץ ה XML (פקודה מס' 1) והמשתמש יוכל לבחור אם רוצה להמשיך עם משחק שכזה או לאו.  כנ"ל בפקודה מס' 4 המתארת את "מצב היד"  מעודכן גם בקובץ התרגיל (3a ו 4 a) |
| 2 | 27.11.17 | האם אפשר לבצע את פקודה מס' 1 כמה פעמים לפני שמחליטים לבצע את פקודה מס' 2 ? (כלומר לטעון כמה קבצי משחק ברצף מבלי באמת "לשחק" בהם ?) | כן יש לאפשר יכולת זו. אפשר יהיה לטעון כמה קבצים (פקודה מס' 1) ורק בסוף להחליט שמבצעים את פקודה מס' 2. פקודה מס' 2 תבוצע כמובן בהקשר הקובץ האחרון שנטען. מעודכן גם בקובץ התרגיל (13.1 ו 13.2) |

תרגיל 2 – מימוש טקסס הולדם כאפליקציית JavaFX (35%) – הגשה: 7.1.18

## פרטים יבשים

צפי תחילת עבודה: **17.12.17** תאריך הגשה: **7.1.18**

צפי זמן לביצוע: 3 **שבועות** ציון אפשרי מקסימלי: **105**

משקל התרגיל: **35%** קושי: **מאתגר**

**מטרות התרגיל העיקרית**

1. מימוש המשחק באמצעות JavaFX, בתצורה של ריבוי משתתפים
2. מימוש פיצ'ר: בליינדים משתנים
3. מימוש "שידור חוזר" (replay) של "יד"

## דרישות

1. המטרה המוצהרת של תרגיל זה היא לייצר פלטפורמה שבה קל יהיה לתרגל ולהתאמן.

בתרגיל זה יש לתמוך בתצורה נוספת של משחק: משחק מרובה משתתפים ידועים מראש (בין 3 ל 6).

כמו כן תממשו גם את היכולת לנתח "יד" בדמות replay על מהלך היד, שבו כל המידע גלוי לכל דיכפין !

בתרגיל זה יושבים כלל השחקנים מול מסך משותף. הקלפים של כולם מוסתרים מעיני כל. כדי ששחקן (אנושי) יוכל לראות את קלפיו הוא יוכל לעשות זאת בתורו בלבד (אחרי שלכאורה ביקש מחבריו שלא להסתכל).

1. נתוני המשתתפים (כמות, שמות, סוג) ידועים מראש ויתקבלו כחלק מקובץ ה xml ([נספח ד'](#_הגדרת_סוג_משחק))

לכל שחקן יש id ייחודי. id **לא חייב** להיות מספר רץ – אלא בסה"כ מספר חיובי שלם...

1. בתרגיל זה תממשו פיצ'ר נוסף של המשחק: בליינדים משתנים.

המידע אודות פיצ'ר זה מצוין בקובץ ה XML, בדמות המאפיין fixed=false.

במצב זה, אחרי שמתבצע סבב ידיים מלא (כלומר כפתור הדילר חוזר לשחקן הראשון ממנו הוא התחיל), הבליינדים יגדלו בשיעור קבוע – עפ"י נתון המגיע מקובץ ה XML – additions.

יש מגבלה על כמות הסבבים (סבב = אחרי שכפתור הדילר משלים מעבר אחד על כל השחקנים) – גם נתון המגיע מקובץ ה -XML – max-totlal-rounds.

יש לחשב ולוודא כי גובה ה big המקסימלי לא יעלה על חצי מגובה ה buy.

את גובה הב big המקסימלי יש לחשב בצורה הבאה:

TotalRounds = total hands \ number of players

Max big = min (totalRounds X additions, max-total-rounds X additions ) + starting big

1. מהלך המשחק:
   1. תתבצע טעינה של משחק מקובץ ה xml. בסיומה יוצגו פרטי המשחק ופרטי השחקנים במשחק.

המטרה היא לאפשר לשחקנים להתרשם ממבנה המשחק ולהחליט אם ברצונם להמשיך למשחק זה או לטעון קובץ משחק אחר. על כל משחק יש להציג את הפרטים הבאים:

* כמות הידיים
* גובה ה buy
* גודל ה big וה small (והאם ובכמה הם ישתנו לאורך המשחק)
* רשימת/טבלת השחקנים (כולל שם, id וסוג השחקן).

טעינת הקובץ צריכה להתבצע באמצעות מנגנון ה task, תוך הצגת חיווים על התקדמות הביצוע למשתמש. לצורך המחשת התהליך עליכם לשתול השהיה מלאכותית כדי שהפעולה לא תיגמר מהר מדי.

תהליך הטעינה יכלול (לפחות) את השלבים הבאים:

* טעינת קובץ
* וידוא קלט תקין

עליכם לבצע בדיקות קלט תקין על קובץ המשחק.

כל הבדיקות שבצעתם בתרגיל הקודם תקפות, ובנוסף יש גם לוודא כי:

* אין שני שחקנים עם אותו id.
* **רק אם** פיצ'ר הבליינדים המשתנים מאופשר (fixed=false), יש לוודא כי גובה max-big-blind אינו עולה על חצי מגובה ה buy (כפי שמוסבר בתרגיל 3)

**שימו לב**: צריך לאפשר במצב זה לטעון קובץ משחק נוסף, כל עוד לא לחצו על כפתור 'התחל משחק'.

טעינת קובץ ה xml תתבצע באמצעות file chooser dialog בלבד ! (אין להניח שהקובץ יחכה לכם בספרייה ייעודית, או שהבודק יקליד את תוכנו לתוך תיבת טקסט וכו'.).

הקובץ יכול להימצא בכל ספרייה חוקית על המחשב (כולל ספריה עם רווחים). נא לוודא כי הדבר אינו מכשיל אתכם.

* 1. יש ללחוץ על כפתור תחילת המשחק כדי להיכנס למצב משחק פעיל.

משלב זה המשחק הופך פעיל ובכל רגע נתון יוצגו הנתונים הבאים:

* מספר יד נוכחי מתוך כמות הידיים במשחק
* גובה הקופה המקסימלי (כמות ה buy'ים) שנעשו עד כה.
* מיקומו של הדילר (ובהתאם לזה גם סימון ה small וה big).
* ליד ה small וה big יש לציין את גובהם כרגע.
* ליד כל שחקן יש לציין את הפרטים הבאים:
  + מיקומו בשולחן
  + גובה הקופה האישית שלו
  + כמה buy'ים הוא כבר ביצע
  + כמה ידיים זכה עד עכשיו

היד מתנהלת כפי שהתנהלה בתרגילים הקודמים.

ברגע שמגיע תורו של שחקן ממוחשב הוא מבצע את מהלכו מאחורי הקלעים.

ברגע שמגיע תורו של שחקן אנושי (בסבבי ההימורים השונים) הוא נדרש למלא את גובה ההימור שלו, או לפרוש מן היד או לעשות check (במידה והוא יכול, עפ"י חוקי ההימורים). כדי ששחקן יוכל לראת את קלפיו, יש לאפשר לו (כשזה תורו בלבד !) ללחוץ על כפתור שמציג את קלפיו **אך ורק וכל עוד** הכפתור לחוץ (כלומר ברגע שהמשתמש מרפה את לחיצת כפתור העכבר מהכפתור – הקלפים מוסתרים בשנית).

* 1. כאשר מגיע תורו של שחקן אנושי, הוא יכול להחליט לפרוש מן המשחק (באמצעות כפתור ייעודי). כל עוד ישנם עוד שחקנים אנושים אחרים במשחק – המשחק נמשך. אם זהו השחקן האנושי האחרון שפורש – הרי שאז המשחק נגמר.
  2. בגמר "יד" תינתן אפשרות למשתמש לבחור אם להמשיך ל"יד" הבאה או להיכנס למצב של replay (המאפשר "ניתוח יד".)

במצב replay, אפשר לנוע קדימה ואחורה בין סבבי ההימורים השונים של היד שהרגע הסתיימה, באמצעות כפתורים ייעודיים, Prev ו Next. במצב replay, כל קלפי השחקנים (אנושיים וממוחשבים כאחד) גלויים לכל !

בכל תנועה קדימה או אחורה יוצגו פרטי ה"יד" כפי שהתנהלו בעבר, ובפרט, תור מי היה להמר, כמה הימר, האם פרש וכו'. חישבו על כך כעל הקפאת מצב בכל שלב של היד. ההתנהלות קדימה ואחורה צריכה להיות ברמה של הימור של כל אחד מהשחקנים (בהתאם לדוגמא שתינתן בכיתה). בכל רגע נתון אפשר לצאת ממצב ה replay ולהמשיך במשחק בהתאם להתנהלותו (עם כפתור ייעודי). במידה וזו היד האחרונה – הרי שיציאה ממצב replay למעשה מסיימת את המשחק הנוכחי.

**למען הסר ספק וכל התחכמות שהיא, אין לממש replay באמצעות screenshot של מרכיבי ה UI !**

* 1. בגמר המשחק (בין אם בצורה רגילה ובין אם בשל פרישת כל השחקנים האנושיים) יש להציג את השחקנים לפי סך הניקוד שהם השיגו.

יש להציג טבלה מסודרת שמראה את כל השחקנים, ועל כל שחקן מידע לגבי:

* שמו
* סוגו (אנושי או ממוחשב)
* כמה buy'ים עשה
* כמה ידיים ניצח
* גובה הקופה שהרוויח.

על הטבלה להיות ממוינת לפי גובה הקופה שמרווחת (כך שהשחקן המנצח נמצא בראש הטבלה).

1. מהלכו של שחקן ממוחשב:
   1. בקובץ ה xml, אחד מהפרטים המסופקים לגבי כל שחקן הוא האם מדובר בשחקן מחשב או שחקן אנושי.
   2. במידה ומדובר בשחקן מחשב, בכל פעם שמגיע תורו, מהלכו מבוצע אוטומטית ע"י מנוע המשחק, כך שהתור עובר מייד לשחקן הבא אחריו. שחקן מחשב מוגבל לחוקי התורות והמהלכים כפי שמפורט בסעיף 3.3.
   3. יש לתמוך במצב שבו חלק מהשחקנים הם אנושיים וחלקם ממוחשבים. אין צורך לתמוך במצב שבו כל השחקנים הם ממוחשבים
   4. שחקן מחשב אינו יכול לפרוש מן המשחק. הוא כן יכול "לפרוש" מה"יד" ע"י כך שאינו מהמר, עפ"י כל חוקי המשחק.
   5. כאמור, גם כאן אין שום צורך לממש בינה מלאכותית עבור השחקן הממוחשב. מספיק רק שישחק עפ"י חוקי המשחק. (הוא יכול להיות טיפש לחלוטין ופשוט להגריל את המהלך הבא שהוא מבצע כל עוד הוא חוקי).
2. כאשר המשחק מסתיים יש לאפשר למשתמש לטעון קובץ משחק חדש, או להתחיל מחדש את המשחק הנוכחי.
3. כחלק מבדיקת המשחק ישונה גודל המסך (resize) ותיבדק המע' שלכם במסך בגודל שונה. עליכם לדאוג לסידור נכון של רכיבי ה ui ולוודא את תקינותם גם בגודל קטן. מומלץ מאוד להשתמש ב scroll pane (יש ללמוד עליו לבד) כדי להציג תוכן גדול בתוך שטח מסך קטן. בפרט עליכם לדאוג להראות את לוח המשחק (שיכול להגיע בגדלים שונים ומשונים) בצורה נאותה כך שאף אחד מרכיביו לא יוסתר בגלל גודל מסך קטן מדי. כמו כן חישבו איך לא להסתיר את טבלת השחקנים במשחק כמו גם את היכולת לבצע מהלך ולהתקדם במשחק.

למען הסר כל ספק (!) אין "לטפל" ב resize ע"י כך שפשוט תמנעו מהמסך להיות resizable.

**במידה ומימשתם בונוסים בתרגיל הקודם אין הכרח לגרום להם לפעול גם בתרגיל זה, אולם אם זה מתאפשר זה יחמם את ליבי (למען הסר ספק, חימום ליבי אינו מתורגם להעלאת נקודות).**

## בונוסים

היכן שזה רלבנטי (בונוס 1 ו 2) על הבונוסים להגיע "מכובים" בתור התחלה כך שהבודקת "תפתח" אותם רק לטובת הבדיקה שלהם.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | סוג | מהות | למה שווה לי ? | כמה שווה לי ? |
| 1 | הגדלת ראש נאה | אפשרות החלפת skin למשחק. בבונוס זה תממשו (לפחות) 2 ערכות צבעים נוספות על הערכה הדיפולטית, ותאפשרו למשתמש להחליף את ערכות הצבעים. שימו לב ערכת הצבעים כוללת (לכל הפחות):   1. החלפת רקע של כל המסך 2. החלפת המראה של הכפתורים על המסך 3. החלפת הפונט והגודל של כל ה label על המסך | כי זה בכל זאת תרגיל בממשק גרפי – ומה יותר גרפי מזה ??? | **5 נקודות**  **(ועד ל 100)** |
| 2 | הגדלת ראש קלילה | ממשו את האנימציות הבאות:   * חלוקת הקלפים לשחקנים כמו גם קלפי הקהילה * מתן ההימור בכל סבב יבוצע ע"י הזזת סכום ההימור מצד השחקן אל אזור הקופה הראשית * חשיפת והסתרת קלפי השחקן הנוכחי, בזמן שהכפתור לחוץ * כל אנימציה אחרת שתמצאו לנכון תתקבל בברכה...   האנימציה צריכה להתבצע לכל היותר במשך 2 שניות  יש לאפשר גם לנטרל את האנימציות הנ"ל, כך שהיא לא תאיט את התקדמות המשחק. | כל כך קל... | 5 נקודות  **(מעל ל 100)** |
| 3 | הגדלת ראש חביבה | אפשרו למיין את טבלת השחקנים המנצחים המוצגת בסיום כל משחק עפ"י שאר העמודות המוצגות בה, ולא שתהיה ממוינת אך ורק עפ"י גובה הקופה. | כי מן הסתאם כבר יש רכיב UI שיודע לתמוך בזה וכמה מסובך זה כבר יכול להיות ??? | 6 נקודות  (ועד ל 100) |

## 

## סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. jar (אחד או יותר) שהוא כל הקוד שלכם, בצירוף קובץ אצווה (batch) שהפעלתו תריץ את התוכנית
2. קבצי קוד המקור של הפרויקט שלכם.
3. קובץ ההגשה יכיל גם קובץ readme שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, כמו גם כל דבר נוסף העולה על דעתכם שחשוב שהבודקת תדע.
4. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות החדשות ותפקידם. (חישבו מה יסייע לבודקת להיכנס ביתר קלות לקוד שלכם ולהבין מי נגד מי...)
5. במידה ומימשתם את מי מהבונוסים לעיל, ציינו זאת בקובץ ה readme כדי שהבודקת תדע לבדוק בהתאם. את מי מהבונוסים שמימשתם בתור task – יש לציין במפורש מיהו הclass שיורש מ task וכו'.

בונוס שימומש אבל לא יתועד – לא ייבדק !

## שאלות ותשובות

|  |  |  |  |
| --- | --- | --- | --- |
| # | תאריך | שאלה | תשובה |
|  |  |  |  |

תרגיל 3 – מימוש טקסס הולדם כאפליקציית WEB (45%) –הגשה: 25.2.18

## פרטים יבשים

צפי תחילת עבודה: **21.1.18** תאריך הגשה: **25.2.18**

צפי זמן לביצוע: **4 שבועות** ציון אפשרי מקסימלי: **120**

משקל התרגיל: 45**%** קושי: **ווא'עליה אלל'בחטי**...

**מטרות התרגיל העיקרית**

1. מימוש המשחק כאפליקציית web.
2. מימוש תצורה של ריבוי משחקים

## דרישות

1. בתרגיל זה יש לפתח מצב של ריבוי משחקים.

כל משתמש (= שחקן) שנכנס למע' עובר תהליך של רישום (sign up) ויכול להעלות קובץ משחק, שאליו נרשמים שחקנים שמעוניינים לשחק בו. כשמתקיימים התנאים הנדרשים, מתכנסים כל השחקנים שנרשמו עבור משחק מסויים והמשחק מתקיים.

1. בתרגיל זה כל שחקן משחק מ browser שלו, רואה רק את קלפיו (וקלפי הקהילה), והאינטראקציה שלו עם השחקנים האחרים מתבצעת למעשה רק ע"י השרת המתווך בינהם.

פורמלית, יש לתמוך בחווית משחק תקינה מעל chrome בלבד.

גם אם פיתחתם מעל FF או IE (טפו..) – וודאו כי אתם עובדים בצורה מוצלחת מעל chrome. האפליקציה תיבדק רק מעל chrome. בקיצור chrome, קאפיש ?

1. ככלל, כל פרטי המידע במע' צריכים להתעדכן בצורה אוטומטית עבור כל שחקן (כל browser) בצורה של Pull כפי שלמדנו בדוגמא הסופית.
2. אין צורך לתמוך ב back כאמצעי למעבר בין המסכים השונים. כמו כן הניחו כי לא נתעלל באפליקציה ולא נשנה את שורת הכתובת, לא נמחק cookies וכו'. הבדיקה תתבצע אך ורק ע"י שימוש בפונקציונליות שתאפשרו במסכי המשחק השונים.
3. אין צורך לבצע שמירה של נתונים מעבר למופע הנוכחי של השרת. במילים אחרות כשהשרת יורד כל הנתונים נעלמים. לא שומרים את המשתמשים שנרשמו, את המשחקים שקויימו וכו'.
4. יש לדאוג ולוודא כי מסכי המשחק לסוגם מתנהגים יפה ב resize ולא מחסירים שום פרט (כשם שהיה נהוג בתרגיל 2).
5. אין צורך לתמוך בפיצ'ר replay בתרגיל זה. (אלא אם תממשו את הבונוס מתנה מס' 2...)
6. בתרגיל זה ישנם 3 שלבים עיקריים:
   1. **מסך רישום למע' (sign up)**

במסך זה יש לאפשר למשתמש להירשם למע'.

לכל משתמש יש שם משלו, ואין לאפשר כמה משתתפים בעלי אותו שם.

(עליכם כמובן לוודא זאת ולהתריע במידה ומשהו מנסה להירשם למע' תחת שם שכבר קיים.)

במסך זה תיבחרו גם את סוג המשתמש: אנושי או ממוחשב. (כן, אתם (בני אנוש) מבצעים רישום של שחקן ממוחשב).

אחרי רישום מוצלח תעברו למסך מס' 2...

* 1. **מסך ריכוז כל המשחקים הקיימים**

במסך זה ירוכזו 2 רשימות עיקריות:

* רשימת כל השחקנים הפעילים כרגע במע'
* רשימת כל המשחקים האפשריים במע'

כל שחקן שנרשם למע' יכול להעלות ולפרסם משחק משלו, אולם אינו חייב !

המשתמש בוחר קובץ משחק (ממחשב הקליינט) ומעלה אותו אל השרת. ההעלאת הקובץ יכולה (וצריכה) להתבצע בהתאם לדוגמא שנלמדה בכיתה. אין שום צורך להשתמש בספריות צד שלישי (apache commons וכיוצב') בשביל להעלות את הקובץ. כשתוכן הקובץ מגיע לשרת, יבוצעו בו בדיקות הקלט כפי שמפורט מטה.

באם הקובץ תקול – הודעה מתאימה הכוללת את סיבת התקלה צריכה להופיע למשתמש, והקובץ **אינו** מועלה בסופו של דבר לשרת. אם הקובץ תקין – הודעה מתאימה צריכה להגיע למשתמש.

קובץ המשחק הוא כמובן קובץ ה xml בפורמט של תרגיל 3 בלבד !.

בקובץ מסוג זה, GameType יהיה מסוג DynamicMultiPlayer.

משחק מסוג DynamicMultiPlayer השחקנים אינם ידועים מראש - אם כי מצטרפים באופן דינמי למשחק.

בקובץ זה יופיע גם מבנה הנתונים כמתואר [בנספח ד'](#_מבני_נתונים_נוספים_1) של DynamicMultiPlayer.

קובץ מסוג DynamicMultiPlayer מגדיר את

* מבנה המשחק (בדומה לתרגיל 1)
* שם המשחק (game-title)
* כמות שחקנים שצריכים להצטרף למשחק לפני שהוא יכול להתחיל (יכולים להיות אנושיים או ממוחשבים)

יש לבצע בדיקות קלט על הקובץ, ובמידה והוא לא תקין להתריע בפני המשתמש ולא לטעון את המשחק. בפרט יש לבדוק את הדברים הבאים:

1. לכל משחק יש שם/כותרת. אין לאפשר לטעון משחק בשם שכבר קיים במע'.
2. כל בדיקות התקינות שביצעתם בתרגיל 1 ו 2 צריכות לבוא לידי ביטוי גם כן.

מרגע שמשתמש העלה משחק בהצלחה – הוא יופיע ברשימת המשחקים האפשריים.

על כל משחק שכזה ברשימה יש לתת את הפרטים הבאים:

* שם המשחק
* שם המשתמש שהעלה את המשחק
* כמות ידיים למשחק.
* גובה ה Buy.
* כמות שחקנים נדרשת + כמות שחקנים שנרשמו בפועל למשחק

(כמה באמת החליטו שהם רוצים לשחק את המשחק הספציפי הזה)

* מידע על בליינדים: יש לכלול את הרמות ההתחלתיות של ה Big ו Small, כמו גם מידע על השתנות הבליינדים במידה ומוגדרת כזו.

הן רשימת השחקנים הרשומים במע' והן רשימת המשחקים האפשריים במע' צריכה להתעדכן באופן אוטומטי לכל היותר עד 2 שניות. (יש לעבור בצורה של pull מהשרת, כפי שהודגם בכיתה בדוגמא הסופית).

כאשר מעדכנים את פרטי המשחק האפשרי, יש לעדכן גם את כמות השחקנים הרשומים לו כרגע.

יש לאפשר למשתמש לבצע Logout מהמע' ע"י כפתור ייעודי. במידה וביצע Logout הוא נמחק מרישומי המע' ויכול לבצע רישום מחדש (גם באותו השם). במידה וקיימים כמה browser'ים שונים שמהם מחובר המשתמש, הרי שהחל מעכשיו הוא כבר לא קיים יותר במע' ולכן הוא יתנתק ממנה גם בכל שאר ה browser'ים.

כל שחקן יכול להשתתף במשחק יחיד בלבד. הוא עושה זאת ע"י בחירת המשחק הרלבנטי מהרשימה ולחיצה על כפתור כניסה למשחק.

שחקן יכול לבחור לשחק משחק שהוא בעצמו העלה, או כל משחק ששחקן אחר העלה.

שחקן יכול לפרוש ממשחק שנרשם אליו (ממסך מספר 3) ולחזור חזרה למסך משחקים אפשריים.

כל עוד המשחק לא החל (אין מספיק שחקנים למשחק כנדרש עפ"י הגדרתו) – הוא יכול להירשם לאותו המשחק (או לכל משחק אחר) כמה שירצה.

אם פרש לאחר שהמשחק החל (מפורט בהמשך) – לא יוכל לחזור ולהירשם למשחק "פעיל", גם אם זה המשחק ממנו יצא אך זה לא מכבר.

במידה ונרשמתם כשחקן ממוחשב – אתם (בני האנוש = הבודק) מבצעים גם את הרישום שלו למשחק מסוים.

למעשה פעילותו של השחקן הממוחשב באה לידי ביטוי רק במהלך המשחק עצמו. את כל הבירוקרטיה של הרישום למע', הצעת משחק ורישום למשחק קיים – מבצעים אתם, בני האנוש, בשמו.

ברגע ששחקן בוחר להיכנס למשחק הוא מייד מועבר למסך מספר 3.

עבור כל משחק יוגדרו כמות השחקנים שישחקו בו. ברגע שנכנסים למשחק מספיק שחקנים – הוא מתחיל בצורה אוטומטית. (בשלב זה כל השחקנים הרשומים בו כבר נמצאים במסך מספר 3)

משחק שמתחיל מסומן בצורה של "פעיל" במסך מספר 2, ולא ניתן כמובן להמשיך ולהצטרף אליו. במידה ופורשים ממנו אנשים במהלך המשחק, אחרי שכבר החל – יש לציין זאת בדמות כמות השחקנים שרשומים למשחק. (כך שייתכן שבמשחק אמורים להשתתף 5 שחקנים, הוא מסומן כ"פעיל" אך יש בו כרגע רק 3 שחקנים רשומים (כי 2 פרשו))

כשמשחק נגמר (כך או אחרת) יש להפוך אותו למשחק זמין מחדש, לאפס את כמות השחקנים שרשומה אליו ולאפשר להירשם אליו מחדש כמשחק רגיל לכל עניין ודבר.

* 1. **מסך חדר משחק בודד למספר משתתפים**

במסך זה ישוחק משחק בודד כפי שמימשתם בתרגיל 2.

כל אחד מהשחקנים יכול לראות את המידעים הבאים:

* קלפי הקהילה
* גובה הקופה של ה"יד"
* מספר היד המשוחקת (מתוך כמה ידיים מוגדרות למשחק סה"כ)
* סימון הדילר, small ו big (כמו גם את גובה ה small וה big).
* עבור כל שחקן במשחק יש להציג את הפרטים הבאים:
  + מיקומו בשולחן
  + אנושי או ממוחשב
  + גובה הקופה הפרטית שלו
  + בכמה ידיים ניצח עד עכשיו

ברגע שנרשמים מספיק שחקנים למשחק מסוים (עפ"י דרישות המשחק) – הוא מתחיל אוטומטית. סדר השחקנים הוא עפ"י סדר הרישום שלהם למשחק, החל מהדילר שמאלה.

יש לציין שהמשחק החל בצורה ברורה לככל השחקנים.

בכל רגע נתון יש להציג את רשימת השחקנים ובנוסף מיהו השחקן שזהו תורו "לדבר".

**מהלך המשחק**

כל עוד אין זה תורו של השחקן – הוא יכול רק להביט בלוח ואינו יכול לשנותו.

ברגע שמגיע תורו הוא יכול לבצע מהלך עפ"י חוקי המשחק כפי שהוגדרו לפני כן.

**סיום משחק**

במהלך משחק "יד", שחקן יכול לבחור לפרוש מן ה"יד", כפי שמוגדר בחוקים ומומש כבר בתרגיל 2.

בלי קשר לכך, שחקן יכול להחליט כי הוא פורש מן המשחק לחלוטין באמצעות כפתור ייעודי.

במצב זה המשחק ממשיך (למרות שכבר אין בו את כמות השחקנים הנדרשת עפ"י הגדרתו).

אין לאפשר רישום של שחקנים נוספים למשחק זה (על אף שבדיוק התפנה בו מקום).

השחקן הפורש לא יכול לחזור בו מפרישתו. אם פרשת – נסעת...

במידה ופורשים כל השחקנים – האחרון שנשאר מנצח ניצחון טכני. יש להודיע זאת (לשחקן הבודד שנשאר כמובן) בצורה ברורה.

במידה והמשחק מסתיים בצורה רגילה (נגמרו כל הידיים המוגדרות למשחק) יש להודיע זאת בצורה ברורה לכלל השחקנים.

בגמר המשחק, בצורה כזו או אחרת, אחרי קבלת ההודעה הרלבנטית מועברים המשתמשים חזרה למסך מספר 2

**שחקן ממוחשב**

רק עכשיו מגיעה לידי ביטוי יכולתו של השחקן הממוחשב.

ברגע שמגיע תורו הוא מבצע את מהלכיו, כפי שהיה בתרגיל מס' 2.

שחקן ממוחשב אינו יכול לבחור לפרוש מן המשחק. הוא כן יכול לבחור "לפרוש" מן היד בכך שאינו מהמר. כל פעולותיו מתבצעות למעשה אך ורק מאחורי הקלעים ולא באופן פעיל אל מול ממשק המשתמש.

לא ישוחק משחק שבו יש **רק** שחקנים ממוחשבים

## בונוסים

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | סוג | מהות | למה שווה לי ? | כמה שווה לי ? |
| 1 | הגדלת ראש נאה | לאפשר לשחקן להשתתף ביותר ממשחק אחד. במצב זה צריך לאפשר לשחקן להיכנס למע' בשנית, הפעם כמשתמש רשום (login). אז הוא מגיע למסך 2 ויכול לבחור להרשם למשחק נוסף. | לא סטנדרטי. | 7 נקודות  (עד ל 100) |
| 2 | שדרוג מכובד ! | אפשור יכולת ה replay שממשתם כבר בתרגיל 2. בשונה מתרגיל 2, יכולה ה replay תתבצע כאן על כלל הידיים, ולא רק על היד האחרונה ששוחקה. שחקן יכול לבחור replay רק כשפרש מן המשחק, או כשהמשחק הסתיים, ולפני שהוא חוזר חזרה למסך מס' 2. (זאת כדי לא לעכב את השחקנים האחרים במהלך ה"ידיים" בתוך המשחק). במצב זה השחקן יצטרך לבחור את היד שעליה הוא רוצה לראות replay ומשם ההתנהלות ממשיכה כפי שהוגדרה בתרגיל 2. שחקן יכול לבחור replay עבור כל יד ששוחקה (בהשתתפותו) ובתוך כל יד ללכת "קדימה" ו"אחורה" באמצעות כפתורים ייעודיים | אם מימשתם נכון, זה אמור להיות רק עבודה בצד הקליינט (יחסית בקטנה) ויהיה מגניב לאללה... | 10 נקודות **(מעל ל 100)** |
| 3 | הגדלת ראש נאותה | אפשרו לשחקן להשתתף במשחק כצופה מן הצד. במצב זה שחקן יוכל עדיין להיכנס לחדר משחק של משחק שכבר החל, ויכול לראות את מהלך המשחק אך הוא אינו שחקן פעיל בעצמו ולא יכול לשחק.  השחקן הצופה מן הצד יראה בכל רגע נתון את את **כל** הקלפים של כלל השחקנים (אנושים או ממוחשבים).  השחקן הצופן מן הצד יקבל את כל החיווים שמקבלים כלל השחקנים הפעילים (כמו שחקן פרש, המשחק נגמר, תור מי עכשיו, מספר מהלכים עד עכשיו וכו').  במצב זה יש להראות בנוסף לרשימת השחקנים הפעילים במשחק גם רשימה של כל השחקנים הצופים מן הצד. יש להראות את 2 הרשימות הנ"ל גם עבור השחקן הפעיל וגם עבור הצופה מן הצד.  במסך של חדר המשחקים הפעילים יש להוסיף עמודה נוספת עבור כל משחק שמציגה כמה שחקנים צופים מן הצד כרגע נמצאים בו (בנוסף לעמודה שמציגה כמה שחקנים פעילים משחקים בו).  ניתן להצטרף לצפות במשחק לפני שהוא מתחיל.  יש לאפשר לשחקן הצופה מן הצד לפרוש מצפייתו מן הצד ולחזור למסך המשחקים הפעילים. (יש לעדכן כמובן את מונה השחקנים הצופים מן הצד, כמו גם את רשימות השחקנים במקומות הרלבנטים). | פונקציונליות מגניבה, ממש ! | 10 נקודות  **(מעל ל 100)** |
| 4 | הגדלת ראש מגניבה ! | אפשרו לשחקנים לקיים chat בינהם תו"כ התנהלות המשחק עצמו. ה chat צריך להתנהל כולו אל מול השרת כמובן בצורה שבה כולם רואים את מה שכולם כותבים. (מאוווווווווווד דומה לדוגמא הסופית)  הצ'ט מתבצע בין השחקנים הרשומים למשחק בלבד (ולא השחקנים הצופים במידה ומישהו מימש את בונוס 4) | היש מדהים מזה ??? | 10 נקודות  (ועד ל 100) |

## סבבה, סיימתי. מה ואיך להגיש ?

יש להגיש קובץ zip המכיל:

1. WAR אחד בדיוק שיושם בספריית tomcat ויעבור deployment אוטומטי. על WAR זה לכלול את כל התלויות שלכם,

אין להניח כי יסופקו לכם תלויות מבחוץ (למשל Gson וכו').

1. קבצי קוד המקור של הפרויקט שלכם (גם צד השרת וגם צד הלקוח).
2. קובץ ההגשה יכיל גם קובץ readme שיכיל הסבר על המערכת, בחירותיכם השונות במקומות שבהם היו לכם בחירה, כמו גם כל דבר נוסף העולה על דעתכם שחשוב שהבודקת תדע.

בפרט **חייב** להכיל קובץ ה readme את ה url שאיתו מנווטים אל מסך ה login של המשחק שלכם.

1. יש לכלול בקובץ ה **readme** גם תיעוד והסבר כללי (וממצה) של המחלקות העיקריות החדשות ותפקידם. (חישבו מה יסייע לבודקת להיכנס ביתר קלות לקוד שלכם ולהבין מי נגד מי...)
2. במידה ומימשתם את מי מהבונוסים לעיל, ציינו זאת בקובץ ה readme כדי שהבודקת תדע לבדוק בהתאם.

בונוס שימומש אבל לא יתועד – לא ייבדק !

הגשה באיחור, שאינה באישור, תבטל כל מימוש בונוס. אין להגיש באיחור בשביל להספיק לעשות

בונוסים. תכננו את הזמן בהתאם.

## שאלות ותשובות

|  |  |  |  |
| --- | --- | --- | --- |
| # | תאריך | שאלה | תשובה |
| 1 |  |  |  |
|  |  |  |  |

נספח א' – דירוג ידיים בפוקר - מן החלש ביותר (10) לחזק ביותר (1)

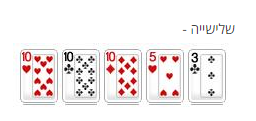


10

4









3

9



2

1

8

7

6

5

נספח ב' – שימוש בספרייה חיצונית לטובת חישוב דירוג ידיים

במהלך המשחק תדרשו להכריע איזו יד היא המנצחת מבין שלל השחקנים שיושבים לשולחן (בין אם הם אנושיים או ממוחשבים).

קביעת ה"יד" המקסימלית האפשרית מתוך רצף של 7 קלפים אינה משימה קלה כלל וכלל (מחד) ואולם בסופו של דבר היא טכנית לחלוטין (מאידך).

ברשת כבר ישנם הרבה אנשים/חברות שחשבו לממש פוקר בעצמם וכבר מצאו עצמם עומדים אל מול סוגייה זו.

הם, בניגוד לכם, לא היו בקונטקס ובלחץ של תרגיל בקורס אקדמי ולהם היה את הזמן, המותרות (ובואו נקווה בעיקר המקצועיות) לשבת ולכתוב ספריות צד שלישי שכאלה שמבצעות בדיוק את המטלה הנ"ל.

אתם נדרשים אם כן להשתמש בספריה צד שלישי שכזו לטובת העבודה הנ"ל. מי שיחליט לממש זאת בעצמו לוקח על עצמו את הסיכון שאם יתגלו טעויות במימוש שיבואו לכדי ביטוי בכך שלא תיבחר היד המנצחת המקסימלית או אם המימוש לא יהיה יעיל וייקח זמן רב (וזו הגדרה סובייקטיבית לכשעצמה) – הוא יאבד ניקוד רב ללא יכולת לתקן.

**אל תכתבו ספרייה שכזו בעצמכם במסגרת ולטובת התרגיל !**

ברשת קיימים מספר רב (מדי !) של מימושים של ספריות שכאלה.

בחנתי ספרייה אחת כזו בעצמי ונראה ודומה כי היא מספקת את העבודה בצורה סבירה ונעימה.

כחלק מדרישות הקורס, תצטרכו לחקור אותה גם אתם ולהשתמש בה גם כן.

עליכם להבין כיצד לעבוד עם הספרייה ואיך להתממשק איתה:

<https://github.com/rundef/poker-odds-calculator>

הספרייה הנ"ל מועלית למאמא לנוחיותכם בדמות קובץ jar.

יש להשתמש בספרייה אך ורק כ jar חיצוני. אין להוסיף את הקבצים שלה לקבצי הפרויקט עצמם.

**שימו לב**: אינכם מחויבים לעבוד בהכרח עם ספרייה זו.

יש ברשת עוד ספריות [רבות](https://github.com/search?l=Java&q=poker+hand+calculator&type=Repositories&utf8=%E2%9C%93) ואתם רשאים לבחור כל ספרייה אחרת שאתם מוצאים לנכון (שימו לב כי מהמחקר הקצר שעשיתי רובן פועלות באופן דומה ובעלות ממשק זהה).

אם אתם בוחרים ספרייה בעצמכם, שימו לב כי:

* 1. הספרייה חייבת להיות ג'אוואית, בפורמט של jar חיצוני שיתווסף לפרויקט שלכם.

אין לקחת ספרייה בשפה אחרת ולהניח את קיומם של תשתיות הרצה אחרות אצל הבודק כמו פייטון, דוט נט וכו'.

* 1. הספרייה חייבת להיות פנימית בתוך המע' שלכם, כלומר אין להסתמך על ספרייה ברשת ולהוציא אליה בקשות לקבלת המידע.

נספח ג' – תיאור המשחק באמצעות קובץ XML

תיאור המשחק על אופניו השונים נתון באמצעות קובץ XML.

במהלך הבדיקה (של שלל התרגילים), תבדק המערכת באמצעות מספר קבצים שונים, חלקם חוקיים וחלקם תקולים, במטרה לראות האם וכיצד המערכת מגיבה לשגיאות.

בחנו היטיב את קבצי הדוגמא שהועלו למע' ה Mama וודאו כי אתם מבינים את פרטיהם ומבניהם.

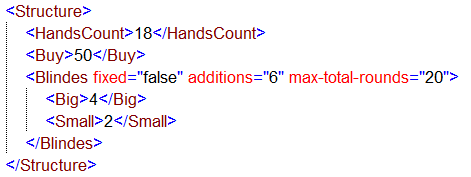
## הגדרת סוג משחק - GameType:

זהו אלמנט המגדיר את סוג המשחק:



* לתוכן האלמנט מספר ערכים אפשריים:
  + Basic – משחק בסיסי, מותאם לתרגיל 1. יש בדיוק 3 שחקנים ממוחשבים ושחקן אנושי אחד.
  + MultiPlayer – משחק למספר שחקנים (בין 3 ל 6). ישוחק בתרגיל 2
  + DynamicMultiPlayer – משחק למספר שחקנים שאינם ידועים מראש. ישוחק בתרגיל 3..

## : Structureהגדרת מבנה המשחק -

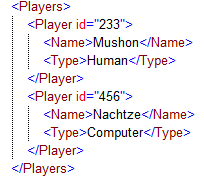


אלמנט זה מכיל את שלל הפרטים הנדרשים כדי להקים את המשחק:

* HandsCount – מתאר את כמות הידיים שישוחקו במשחק. מספר חיובי ושלם
* Buy – מתאר את כמות הצ'יפים שיקבלו המשתמשים. מספר חיובי שלם.
* Blinds – מתאר את גובה הבליינדים ואת התנהגותם לאורך המשחק:
  + Big ו Small מתארים את גובה ההימורים ההתחלתיים בהתאמה.
  + Fixed – מאפיין המתאר האם הבליינדים קבועים (true) או משתנים (false) לאורך המשחק. בוליאני.
  + Additions – במידה והבליינדים אינם קבועים (fixed=false), מתאר כמה יש להוסיף לכל אחד מהם (big ו small) בכל שלב של העלאת הבליינדים. מספר חיובי שלם.
  + Max-total-rounds – מתאר את סה"כ כמות הפעמים שבהם תתבצע העלאת של big (ו small). תקף רק במידה והבליידים משתנים לאורך המשחק (fixed=false). מספר חיובי שלם.

## מבני נתונים נוספים לסוגי המשחקים השונים

* כאשר GameType הוא MultiPlayer, נוסף מבנה נתונים נוסף המתאר את השחקנים שישחקו במשחק.



פרטי השחקנים מופיעים באלמנט Players (זהו אלמנט אופציונלי שיופיע **אך ורק** בתרגיל 2).

כל Player מכיל נתונים לגבי השחקן:

1. Id - מס' מזהה. חייב להיות unique. מספר שלם כלשהוא עם 5 ספרות לכל היותר (לאו דווקא בסדר רץ)
2. Name - שם השחקן. מחרוזת תווים המתארת את שם השחקן.
3. Type - סוג השחקן. ערכים אפשריים הם:
   1. אנושי (Human)
   2. ממוחשב (Computer)

* כאשר GameType הוא DynamicMultiPlayer, זה סוג משחק הזהה בחוקיו ובהתנהלותו למשחק מרובה משתתפים (תרגיל 2), למעט העובדה כי נתוני השחקנים אינם נתונים מראש אם כי ייבחרו דינמית כחלק ממהלך האפליקציה.

מבנה הנתונים הבא מתאר את משחק זה (זהו מבנה אופציונלי, שיופיע **אך ורק** בתרגיל 3):



הסבר:

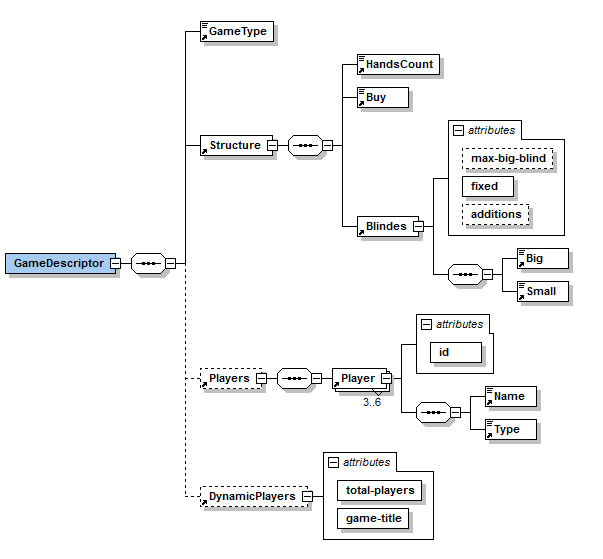
1. game-title - לכל משחק יש שם unique שמזהה אותו.
2. total-players - כל משחק מגדיר את סך השחקנים שמשחקים בו. כמות השחקנים יכולה להיות בין 2 ל 6.

נספח ד' – קובץ סכמה לוידוא קובץ ה XML

קובץ ה XML מקושר לסכמה שמגדירה את חלקיו השונים ואת חוקיותם.

מומלץ להשתמש בסכמה כדי לוודא כי הגדרתם את הקובץ נכונה.

(הסכמה הועלתה למערכת ה Mama).

להלן תרשים המתאר את מבנה הקובץ כפי שמוגדר ע"י הסכמה: