

תרגיל 2 בקורס ג'אוה ואינטרנט

מגשים:

איתמר ולר 205430291 Itamarwe@mta.ac.il

שירה יד שלום 315780486 Shirayd@mta.ac.il

הפעלת התוכנית

בתחילת התוכנית יש ללחוץ על כפתור Repository: בתחילת התוכנית המשתמש יוכל לבחור מתוך 3 אפשרויות איך לטעון את repository:

1. Load repository – לאחר בחירה באפשרות הרצויה, אם התוכנית תטען את repository בהצלחה היא תעדכן את הנתבי בחלון התוכנית למטה.
 - אם המשתמש בחר באפשרות Load from XML והקובץ XML לא יטען כנדרש התוכנית תקפיץ חלון התראה למשתמש שבה מפורטת הבעיה.
2. Start game – אין למשתמש אופציה ללחוץ על הכפתורים טרם נטען נתבי בהצלחה. במידה והנתבי נבחר בהצלחה התוכנית תציג את הכפתורים כנדרש.
3. Change user name - מאפשרת למשתמש לעדכן את שם המשתמש האקטיבי שבגינו מבוצעות כל הפעולות וכן מעדכנת בחלון התוכנית למטה. אופציה זו זמינה גם אם לא נטען Repository בהצלחה.

לאחר טעינת repository האופציות הפתוחות למשתמש:

תחת הכפתור Repository

Load from XML -טוענת את פרטי המערכת למנוע מתוך קובץ נתונים בפורמט. במידה והקובץ לא תקין, יפתח חלון המסביר בצורה ברורה מה לא תקין בקובץ.

Switch repository – מאפשרת לעבור ולהחליף את ה repository האקטיבי. מבקשת מהמשתמש לבחור ספרייה בה שוכנת repository היעד. במידה והספרייה של **M.A.Git** repository, ההחלפה תתבצע, במידה יקפוץ חלון.

Create new repository – מבקשת מהמשתמש לתת שם ולבחור תיקייה בה repository חדש עליו ירצה לעבוד.

תחת הכפתור Commit

Show current commit file system information -מציגה את כל הקבצים והספריות במע', כפי שהם נגזרים מהקומיט המוצבע ע"י ה HEAD branch. שם (נתבי מלא) סוג (Blob או Folder) SHA-1 מי ביצע את השינוי האחרון שבו. מתי חל העדכון האחרון שחל בו

Working copy status – מציגה את מצב הנתונים במע':

מציגה את שם ה repository את המיקום שלה, מציגה את שם המשתמש האקטיבי כרגע.

מציגה את השינויים הפתוחים המתקיימים ב working copy.

Commit – מבצעת קומיט חדש במע'. מבקשת מהמשתמש הודעה המתארת את אופי הקומיט.

תחת הכפתור Branch

List available branches – מציגה מידע על כל ה branch'ים המתקיימים במע'.

עבור כל branch מציג את המידעים הבאים:

1. שם ה branch
2. Sha-1 של הקומיט המוצבע ע"י ה branch
3. הודעה של הקומיט המוצבע ע"י ה branch

Create new branch – מייצרת branch חדש במע'. מבקשת מהמשתמש שם של branch ומקפיצה הודעה אם קיים branch תחת השם הזה. שואלת את המשתמש אם הוא חפץ גם לעבור אליו אחרי כן (checkout) או לא.

Delete branch – מוחקת branch במע'. מבקשת מהמשתמש לבחור branch למחיקה. מקפיצה חלון שגיאה אם המשתמש מנסה למחוק את ה head branch.

Checkout branch – מבצעת החלפה של ה Head branch. פורסת את working copy מבנה מע' הקבצים כפי שהוא נגזר מן הקומיט המוצבע ע"י ה branch החדש. במידה ויש שינויים "פתוחים" במע' מתריעה על כך למשתמש. המשתמש יבחר אם להמשיך או לבטל את הפעולה.

Show current branch history – מאפשרת לראות את כל היסטוריית הקומיטים של ה active branch, החל מהקומיט הנוכחי (המוצבע ע"י ה head branch) ועד לראשית הזמנים.

מציגה על כל קומיט את הפרטי באופן הבא:

- א. SHA-1
- ב. הודעת הקומיט
- ג. מתי נוצר
- ד. ע"י מי נוצר

Reset branch – מאפשרת למשתמש לשנות את הקומיט המוצבע ע"י ה head branch ל SHA-1 אחר. אם יש שינויים פתוחים במערכת – מתריעה על כך למשתמש. אם בוחר להמשיך – אזי כל השינויים האלה נמחקים ללא יכולת לשחזרם. בגמר הפעולה מציגה את המצב של הקומיט החדש המוצבע ע"י ה branch.

תחת הכפתור Remote






Clone – על המשתמש לבחור ספרייה קיימת של **M.A.Git** (ה RR) כמו גם ספריית היעד אליה הוא ירצה לשכפל את ה repository (ה LR). בנוסף על המשתמש להכניס את שם ה repository החדש שיווצר.

Pull - פעולה זו מייבאת מה RR רק את הקבצים, ספריות, קומיטים של ה branch המדובר. לאחר מכן ה RB, מתעדכן, כמו גם את RTB להצביע על המיקומים הרלבנטים שלהם.

Push - דוחפת מידע חדש (קומיטים, ספריות, קבצים) מ LR ל RR. את המידע החדש ניתן לדחוף אך ורק אם ה RR "נקייה" משינויים.

Push new branch - דוחפת branch חדש ל RR. הדבר מתאפשר כמובן רק במצב שבו משתמש שיכפל את ה repository שלו מ repository אחר. (בנוסף)

כפתורים במערכת

מייבאת את כל המידע החדש <u>שהצטבר</u> ב RR אל ה LR.	 Fetch
דוחפת מידע חדש (קומיטים, ספריות, קבצים) מ LR ל RR.	 Push
מייבאת מה RR <u>רק</u> את הקבצים, ספריות, קומיטים של ה branch המדובר.	 Pull
מבצעת קומיט חדש במע.	 Commit
מבצעת מיזוג בין שני branch'ים שונים. התוצאה של המיזוג הוא קומיט חדש המכיל את סך השינויים והאיחודים שבוצעו בין 2 ה branch'ים.	 Merge


בתהליך המיזוג על המשתמש לבחור branch נוסף אשר ימוזג אל תוך ה head branch.


את התהליך אפשר להתחיל אך ורק כשאין שינויים פתוחים במערכת. (במידה ויש יקפוץ חלון).

במקרים בהם הקובץ שונה (כך או אחרת: עדכון, מחיקה, יצירה) **גם** ב branch הבסיס **וגם** ב branch היעד, **יחסית לאב הקדמון**, מצב זה נקרא קונפליקט. המשתמש יצטרך להכריע כיצד לפתור את הקונפליקט הנ"ל (כלומר איזו גרסה של קובץ היא הנכונה וצריכה להישמר כחלק מתהליך המיזוג).

יוצג למשתמש חלון של רשימת הקונפליקטים שנצברה ותאפשרו לו לפתור אותם. בחלון 4 'אזורים' המציגים את תכולת הקובץ עם הקונפליקט במצבו השונה:

1. תכולת הקובץ בבסיס (ours)
2. תכולת הקובץ באב הקדמון
3. תכולת הקובץ ביעד (theirs)
4. מצב הקובץ אחרי המיזוג (קובץ התוצאה)

כפתור ההתחלת הצגת הקונפליקטים.  Start

לאחר סיום פתרון הקונפליקט, יעבור לקונפליקט הבא.  Next

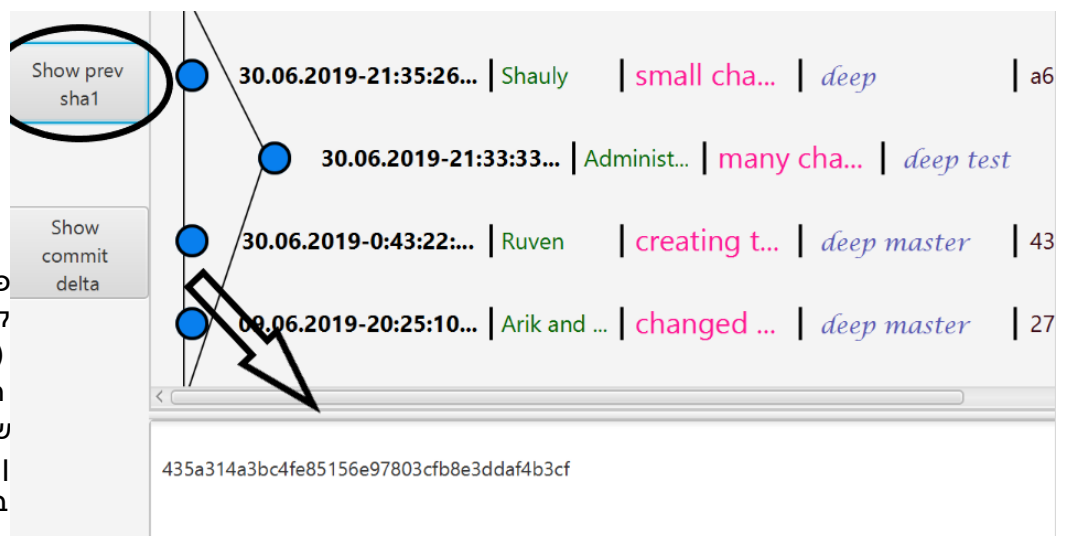
כשיסיים (כפתור Next), ישמור המשתמש את קובץ התוצאה (במקומו הרלבנטי ב WC) ויוכל לבחור את הקונפליקט הבא מרשימת הקונפליקטים לטיפול.

לאחר מכן המשתמש יכניס הודעה המתארת את הקומיט ויתבצע קומיט של כל הקבצים הפתוחים במערכת.

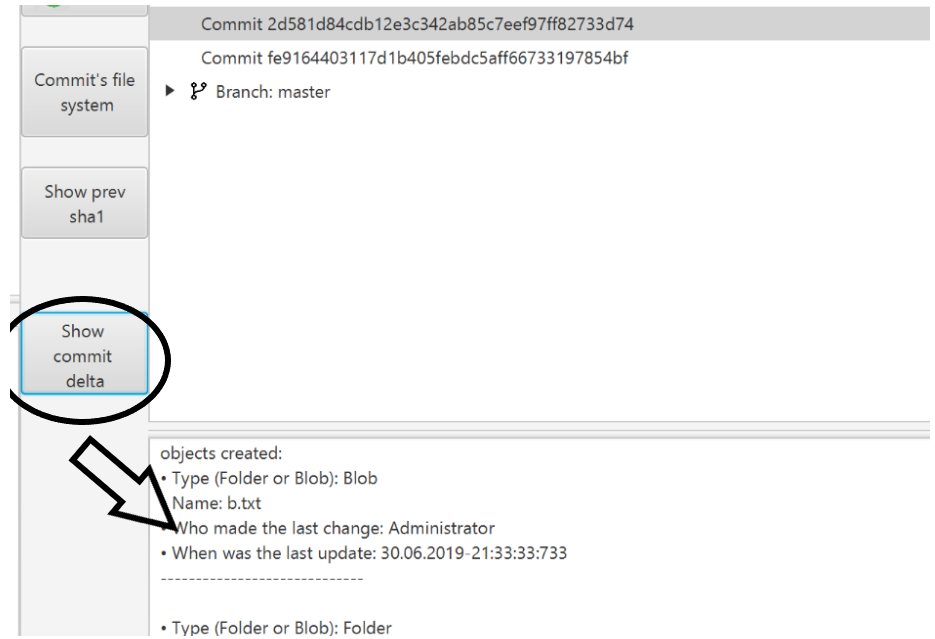
כפתורים נוספים והסבר על חלונות:



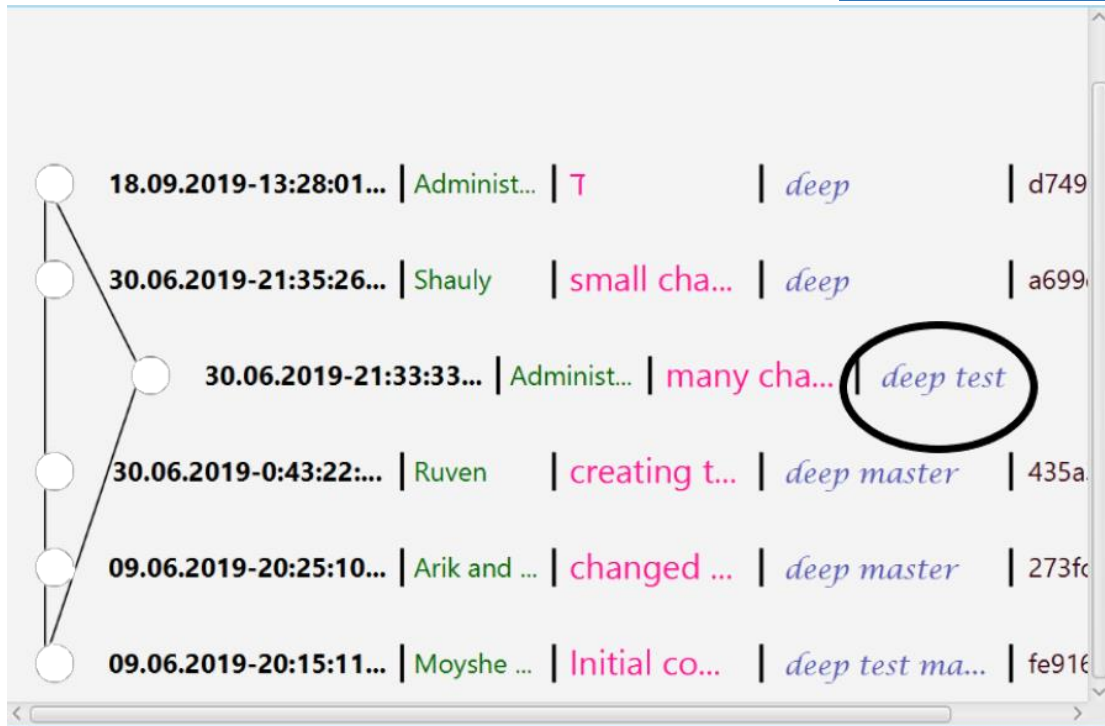
לחיצה על הכפתור (הכפתור יהיה זמין אך ורק אם נבחר קומיט או ע"י בחירה מעץ הקומיטים (graphic) או ע"י בחירת קומיט דרך העץ (text), לאחר מכן המידע יתעדכן בחלון מימין למטה (כפי שמוצבע בחץ השחור). בחלון זה יוצג ה-Sha1 של הקומיט הקודם (האב), במידה ויש שני אבות, אז שניהם יוצגו. שניהם יוצגו בחלון בחץ השחור).



לחיצה על הכפתור (הכפתור יהיה זמין אך ורק אם נבחר קומיט או ע"י בחירה מעץ הקומיטים (graphic) או ע"י בחירת קומיט דרך העץ (text), לאחר מכן המידע יתעדכן בחלון מימין למטה (כפי שמוצבע בחץ השחור). בחלון זה יוצגו פרטים על המידע שהשתנה בקומיט זה (דלתא מול קומיט קודם - יפתח חלון אשר יאפשר לו לבחור דלתא מול מי הוא רוצה לראות.



הסבר על עץ הקומיטים:



לחיצה על הbranch, תפתח חלון בו עלינו יהיה לבחור את ה branch עבורו נרצה להדגיש את הקומיטים השייכים עליו. (בנוס)

לחיצה על קליק ימני בעכבר תפתח תפריט הקשר (context menu). מתפריט ההקשר ניתן יהיה לבצע את הפעולות השונות בהקשר של קומיט/branch זה ביתר קלות:

1. יצירת branch חדש המצביע על קומיט זה
2. איפוס head branch לקומיט זה
3. מיזוג ה branch המצביע על קומיט זה עם head branch
4. מחיקת branch (במידה ויש כזה המצביע על קומיט זה)

(בנוס)

רשימת בונוסים שמומשו:

1. אנימציות- הסבר איך להפעיל בנספח עץ הקומיטים.
2. אפשרות החלפת skin למשחק – דרך הכפתור View.
3. תפריט הקשר (context menu) לבחירה של כל קומיט בעץ הקומיטים- הסבר איך להפעיל בנספח עץ הקומיטים.
4. היכולת לדחוף branch חדש ל RR

מחלקות עיקריות

להלן המחלקות העיקריות בתוכנית (לפי Package):

GUI

Components

Main

-MagitController

המחלקה האחראית אחראית על קבלת מידע מהמשתמש מחלקה זו גם מפעילה את המשימה הנדרשת.

Merge

-MergeController

המחלקה האחראית אחראית על קבלת מידע מהמשתמש איך לטפל בקונפליקט. מחלקה זו גם מפעילה את המשימה הנדרשת.

Visual

layout

CommitTreeLayout

Node

CommitNodeController

המחלקה האחראית על הוספת עץ הקומיטים לחלון התוכנית.

CommitNode

Main המחלקה האחראית על ריצת התוכנית.

Engine

Magit – המחלקה האחראית על הלוגיקה של המשחק. מחלקה זו מכילה בתוכה מתודות Public המציעות החלפת משתמש, אתחול repository, ביצוע קומיט, מחיקת branch כיו"ב. לצורך ניהול התוכנית היא שומרת בתוכה data members לצרכים הבאים:

- String username – שם המשתמש.
- Set<Repository> repos – קבוצה של repository שקיימים במערכת.
- Repository activerepo – repository עליו אנו עובדים כרגע.

MagitObject

מחלקה שאחראית על ניהול האובייקטים בMagit.

Blob

מחלקה שאחראית על ניהול הקבצים בMagit (יורשת מMagitObject).

Folder

מחלקה שאחראית על ניהול התיקיות בMagit (יורשת מMagitObject).

Commit

מחלקה שאחראית על ניהול הקומיטים בMagit (יורשת מ[MagitObject](#)).

[Branch](#)

מחלקה שאחראית על ניהול הbranches בMagit.

[Repository](#)

מחלקה שאחראית על ניהול הRepository בMagit.

[Settings](#)

מחלקה שאחראית על הגדרות כגון משתנים גלובליים וכו'.

[Utils](#)

מחלקה שאחראית על פונקציות שימושיות כגון zip,unzip וכו'.

[Xml](#)

מחלקה שאחראית על ניהול הxml בMagit.

[LoadFromXMLTask](#)

מחלקה שאחראית על ניהול טעינת הXML-TASK.