**Breaking The Enigma EX03**

**פרטי המגישות:**  
לירון קוטב, ת"ז 318598380,lironkt@mta.ac.il

chenpk@mta.ac.il,208711978 חן פקמן, ת"ז

**\*מימשנו את הבונוסים הבאים:**

1)הגדלת ראש מגניבה-צא'ט (מוצג בכל אפליקציה בצד ימין למעלה עבור כל המסכים).

2) בואו נקל על המשתמש- העלאת סוכן בלחיצת כפתור (מוצג בצד ימין למעלה באפליקציית ה- (Allies . בבונוס זה לאחר לחיצה על כפתור ה-Add Agent נפתח מסך הlogin של הagent כאשר שם הallies נבחר עבורו ללא אופציה לבחור צוות allies אחר, ונשאר לו להזין את שאר הפרטים.

ממשנו בונוס זה באמצעות process builder היוצר תהליך עצמאי .   
הוא מפעיל את קובץ AddAgentBonus.bat היושב בתיקיית הallies וקובץ bat זה מריץ את AgentRun.bat היושב בתיקיית הAgent לפי הpath היחסי.

הערות שחשוב לנו לציין:

הסבר על המערכת:

יש לנו 3 אפליקציות כפי שנדרשנו

(AlliesRun.bat)Allies ,(AgentRun.bat)Agent , (UboatRun.bat)UBoat

בנוסף השתמשנו בTomcat כשרת, המודול של השרת נקרא Application.

המנוע של המערכת נמצא תחת המודול EnigmaEngine. לכל אחת מן האפליקציות יש מודול של שכבת UI

יתר על כך השתמשנו במודול נוסף שיש בו מידע שמשותף ל-3 האפליקציות ,שנקרא ClientTools.

חשוב לציין שהשרת הוא היחיד שמכיר את המנוע (EnigmaEngine) וכל בקשת HTTP שמגיעה משכבת הUI עוברת דרך השרת שמחזיר בתגובה את התוצאה הנדרשת.

קיים מודול שנקרא MiniEngine שבעזרת אפליקציית הסוכן מייצרת את המכונה (על ידי קבלת ה-inputStream מהשרת)

קיים נתק בין שכבת הUI לשרת ומתבצעת תקשורת רציפה בין 3 האפליקציות לשרת ולהפך, אך כמובן שכל אחת מן האפליקציות לא מכירה את השנייה ולהפך.

כמו כן קיים מודול של DTOObjects בדומה לפרויקטים הקודמים שהוגשו).

אפליקציית ה- UBoat

לאחר התחברות, מועברים למסך המכונה.

במסך זה ניתן לעלות קובץ xml המכיל את פרטי המערכת, לקבוע קונפיגורציות ולצפות בפרטים טכניים על המכונה. לאחר העלאת קובץ וקביעת קונפיגורציה ניתן לעבור למסך התחרות.

חשוב לציין שבעת העלאת קובץ ה-xml .התחרות מוצגת לצוותי ה-Allies אך התחרות לא תחל עד אשר יוזנו פרטי ה-UBoat במסך התחרות. סטטוס התחרות יהפוך ל Active כאשר הUBoat מזין את כל פרטי התחרות ונרשמו כל צוותי הallies הדרושים.

במסך התחרות לאחר הצפנת מחרוזות לא ניתן לחזור למסך המכונה ולשנות את הקונפיגורציה ההתחלתית עד אשר המשחק מסתיים.

כאשר כל הצוותים מתכנסים והתחרות מתחילה מופיע חיווי במסך התחרות, בסיום התחרות מופיע כפתור Logout.

בסיום התחרות קופצת הודעה המודיעה על סיום התחרות ועל הצוות המנצח, בעת לחיצה על OK המסך מתנקה.

אפליקציית ה-Allies

לאחר ההתחברות, מועברים למסך התחרויות.

שם ניתן לראות את התחרויות הקיימות במערכת מידע על הסוכנים המשויכים לאותו Allies.

על מנת להתחיל תחרות, ה-Allies צריך לבחור תחרות בעזרת לחיצה על הצ'ק בוקס המתאים עבור כל תחרות והכנסת גודל משימה.

על מנת לבחור תחרות אחרת צריך להוריד את הסימון של הצ'ק בוקס ולבחור תחרות אחרת.

לאחר לחיצה על כפתור ה-Ready המידע נשמר.

לא ניתן להיכנס לתחרות שכבר מלאה. כמו כן במידה ובוחרים תחרות אשר ה-Allies שהתחרו עדיין לא לחצו על כפתור הclear - לא ניתן להירשם אליה.

סטטוס התחרות נהפך לאקטיבי רק לאחר שה-UBoat הזין את כל פרטי התחרות וכל צוותי ה-Allies הנדרשים נרשמו לתחרות.

במסך התחרות ניתן לצפות במידעים השונים אודות התחרות, בעבודות הסוכנים ובתוצאותיהם.

כאשר מצטרפים סוכנים חדשים לתחרות שכבר פעילה מופיע חיווי שמציין את שמות הסוכנים אשר הצטרפו לאחר התחלת התחרות ולכן לא פעילים עוד.

בסיום התחרות קופצת הודעה עבור כל צוותי ה-Allies שמשתתפים בתחרות המודיעה על סיום התחרות ועל הצוות המנצח. לאחר לחיצה על OK מופיע במסך הcontest כפתור clear ולאחר לחיצה עליו המסך של הAllies והמסכים של כל הסוכנים הרשומים אליו מתנקה והמשתמש מוחזר למסך התחרויות.

גם באפליקציה זו מופיע חיווי על מצב התחרות, האם היא התחילה, ובמידה ולא מה הסיבה שהתחרות עדיין לא פעילה.

אפליקציית ה-Agent

לאחר ההתחברות מופיעים לסוכנים מידעים שונים לגבי התחרות, תוצאות העבודה שלו ונתונים על התקדמות.

במידה והסוכן מנסה להירשם לAllies אשר נמצא בתחרות , מוצגת לו הודעה המציינת זאת , ואין לו אפשרות לבחור בצוות אחר. לאחר סיום התחרות מסך המידע של הסוכן נפתח והוא מצטרף לתחרות הבאה.

בסיום התחרות קופצת הודעה חיווי על סיום התחרות כאשר הוא מאשר את ההודעה המסך לא מתנקה.

אבל אם צוות ה-Allies שהסוכן מקושר אליו לחץ על כפתור clear - המסך יתנקה.

תיאור המחלקות בקצרה

אפליקציית ה Alliesפונה לservlet: *DMCreateMissionsServlet .*

Servlet זה יוצר את המשימות ומכניס אותם לblocking queue שמוגבל בגודלו ל1000.

במקביל בעת תחילת התחרות, אפליקציית הסוכן פונה לservlet : *AgentGetMissionsServlet* ולוקחת משימות לפי כמות המשימות שנבחרה.

לאחר שהסוכן מקבל את המשימות הוא יוצר עבור כל משימה אובייקט runnable (*AgentMissionRunnable*) ודוחף אותם לblocking queue אשר נמצא בתוך הthreadpool .

ברגע שהסוכן מסיים את המשימות הוא שולח שוב בקשה לאותו servlet , והתהליך מתרחש בצורה זו עד שנגמרות המשימות או עד לסיום התחרות.

בנוסף הסוכן שולח בקשה לקבלת המכונה באמצעות הservlet : *GetXmlInputStreamServlet* .

הסוכן מקבל InputStream (המכיל את הפרטים מהקובץ הXML) מservlet זה ומסרלז אותו לאובייקט *TheMachineEngine* המכיל את פרטי המכונה.

חלק מהמידע המוצג למשתמש ממומש באמצעות refresher על מנת להבטיח את נכונות המידע בצורה עדכנית.

עבור מידע אשר צפוי להיות מידע לא רב השתמשנו בpull all the information all the time ועבור מידע המייצג את תוצאות התחרות (כלומר מחרוזת מועמדת, קוד שנמצאה וכו..) השתמשנו בdelta fetching משום שצפויות להגיע המון תוצאות.