

מגישות

נועה ג'יראד – 318202975, noajh@mta.ac.il
חמוטל אזולאי – 205976731, hamutalaz@mta.ac.il

בנוסים

1. **ציאט:** מופיע במיקומים שונים בכל המסכים באפליקציות השונות.
- * השארנו את מימוש ה-Skins והמילון מתרגיל 2.

המחלקות אותן מימשנו

מודול ה-Server:

1. AgentsServlets package
2. AlliesServlets package
3. ChatServlets package
4. SharedServlets package
5. UboatServlets package

Packages המכילים את כל הסרבלטים הרלוונטיים לכל אובייקט.
Utils package: מכיל את כל המטודות והעזרים הנחוצים לתפקוד השרת, כמו constants, sessions
utils ועוד.

מודול ה-Engine:

:EngineManager package

- **EngineManager**: מחלקה המנהלת את המכונה. מקבלת את קלט המשתמש משכבת ה-UI ותפקידה לשגר את ביצוע הפעולות הנדרשות לחלקים המתאימים במנוע המכונה ולתכלל את הפעולות המתבצעות בחלקיו השונים של המנוע. מחלקה זו אחראית בין היתר על השמירה לסטטיסטיקה, קריאת מכונה מקובץ ושמירת מצב נתון של מכונה.
- **ReadXML**: מחלקה זו אחראית על קריאת קובץ ה-xml בלבד, מבלי לבדוק את תקינות המידע הנמצא בו (פרט לקיומו והתאמתו לסכמה). מחלקה זו מחלצת את המידע מן הקובץ ויוצרת מחלקת CTE-Enigma, ממנה תיווצר בעתיד המכונה הניתנת לשימוש.
- **MachineFactory**: מחלקה שתפקידה ליצור מכונה חדשה על פי המידעים שחולצו מקובץ ה-xml. המחלקה מקבלת מופע של CTE-Enigma ויוצרת מכונה איתה המשתמש יכול לעבוד. המחלקה בודקת את תקינות קובץ ה-xml בהתאם להגדרת מכונה תקינה (כפי שמוגדר בסעיף 1) ומחזירה שגיאות אל ה-EngineManager במידה והן קיימות.
- **ConfigParser**: מחלקה שאחראית לקבל את קלט המשתמש בנוגע לאתחול תצורת קוד ראשונית של המכונה, לבצע עליו בדיקות תקינות ולזרוק שגיאות מתאימות. במידה וכל בדיקות התקינות מתבצעות בהצלחה, ה-EngineManager יקבל בעתיד מופע של מחלקה זו וידע לאתחול את המכונה בהתאם למידע המופיע במחלקה זו.
- **EncryptionData**: מחלקה המכילה מידע הצפנה בודדת, בין היתר מידע כמו מחרוזת מקורית, מחרוזת מוצפנת וכו'.
- **UBoatEngineManager**: interface המכיל את סט הפעולות ש-UBoat אמור לבצע במסגרת תפקידו.

:Enigma Machine package

- **Enigma**: מחלקה המכילה את כל החלקים הדרושים להרכבת המכונה עצמה ובהם הרוטורים, המשקפים והפלאגים. המחלקה מכילה את הפונקציונליות הדרושה להצפנה ותפעול המכונה כמו איפוס המכונה, הזזת הרוטורים ועוד.
- **Rotor**: מחלקה המכילה את המידע הנחוץ לרוטור. המחלקה מחזיקה שתי רשימות right, left המכילות את התווים המוגדרים על הרוטור ואת מיקום זיז הפסיעה ברוטור. המחלקה מכילה מטודות לפונקציונליות של הרוטור כמו קידומו ואתחולו.
- **Reflector**: מחלקה המכילה את המידע הנחוץ למשקף. המחלקה מכילה מפה של כלל המיפויים בתוך הרפלקטור ובעלת פונקציונליות המאפשרת להזין ערך ולקבל את המיפוי שלו.
- **ReflectorID**: Enum שתפקידו להחזיק את ה-reflectorID בתצורתו כספרה רומית. מכיל פונקציונליות המאפשרת להמיר ספרה רומית למספר ולמחרוזת ולהפך.
- **NotchesState**: Enum שתפקידו להוות אינדיקציה האם המידע המתבקש הוא אודות מיקום זיזי הפסיעה בעת אתחול תצורת הקוד של המכונה, או מיקומם הנוכחי. לא מכיל שום פונקציונליות ומהווה קבוע של התוכנית.
- **MachineData**: מחלקה המכילה את כל המידע הנחוץ להגדרת מכונה חדשה שיובא מקובץ xml. מידע זה נאסף על ידי ה-MachineFactory ומועבר לבנאי של ה-Enigma אשר יוצר על סמך מידע זה מכונה חדשה. מחלקה זו מכילה מידע אודות הרוטורים, מיקומם וזיזי הפסיעה שלהם, ה-ABC של המכונה והרפלקטורים. בעלת פונקציונליות המאפשרת לקבל את המידע ולהגדירו.

:XMLFilesEx3 package

- **Package** המכיל מחלקות שנוצרו באופן אוטומטי על ידי סכמת ה-xml. משמשות לצורך טעינת קובץ xml חדש המכיל מכונה.

:Chat package

- **ChatManager, SingleChatEntry**: מחלקות המשמשות את הציאט בצד השרת.

:Contest package

- **Contest**: מחלקה המכילה מידע אודות תחרות.
- **Battlefield**: מחלקה המכילה מידע אודות battlefield כפי שהופיע בקובץ שהועלה על ידי ה-Uboat.
- **AllContests**: מחלקה המכילה מידע אודות כלל התחרויות שהועלו לשרת.

:Users package

- **UserManager**: מחלקה המכילה מידע אודות כלל המשתמשים הרשומים לשרת.
- **UBoat**: מחלקה המכילה את ייצוג הישות UBoat בשרת והפעולות אותן הוא מבצע.
- **Allie**: מחלקה המכילה את ייצוג הישות Allie בשרת והפעולות אותן הוא מבצע.
- **Agent**: מחלקה המכילה את ייצוג הישות Agent בשרת והפעולות אותן הוא מבצע.

:BruteForce package

- **Decryption Manager**: מחלקת האחראית על כלל פעולת ה-brute force. בתוכה מתנהלת חלוקת המשימות לסוכנים, הפעלת ה-thread pool ומנהלת תור של פלטים אותם יש להחזיר ל-UI.
- **Decryption Task**: מחלקה זו מגדירה משימת פענוח בודדת המבוצעת על ידי thread מה-thread pool. מוגדרת בה משימת פענוח: מעבר על קבוצת קודים, מציאת מועמדות לפענוח והעברתן אל תור דיווח ל-UI.

- **ABCCounter**: מחלקה שמאפשרת למצוא את מיקום הרוטורים העוכב. מבצעת ספירת בבסיס L (גודל ה-ABC של המכונה). מאפשרת לקדם את מיקום הרוטורים ולמצוא מיקום בהינתן אינדקס.
- **BruteForceDifficulty**: Enum המשמש להבחנה בין רמות הקושי השונות בברוט פורס. כל רמת קושי מכילה גם הגדרה של מספר תוצאות הפענוח שיש לשלוח ל-UI בכל פעימה.
- **BruteForceResult**: מחלקה המייצגת מועמדת לפענוח בודדת. מכילה את השדות: זמן הפענוח, מחרוזת וקוד המכונה בה היא נמצאה.
- **BruteForceSettings**: מחלקה המכילה את כל הגדרות הברוט פורס שנאספו על ידי המשתמש כמו מספר הסוכנים ורמת קושי, לצד פרמטרים נוספים שהקונטרולר היה מעוניין להעביר למנוע: כמו מה לבצע בעת מציאת מועמדת לפענוח.
- **ProgressInfo**: DTO בין המנוע לקונטרולר המשמש את ה-UI לצורך התעדכנות בסטטוס מלאכת הפענוח. מכיל שדות כמו סה"כ המשימות, כמות המשימות שבוצעו, הודעת סטטוס התקדמות ועוד.
- **FinishedTaskData**: מחלקה המכילה מידע לגבי המשימה שהסתיימה. כדוגמת גודל המשימה ומשך הזמן שלקח לבצע אותה.
- **Constants package**: מכיל מחלקת Constant ובה הקבועים המשמשים את המנוע.

מודול ה-DTO'S:

מודול המכיל את כל ה-DTOs הנחוצים לתקשורת בין הקליינטים השונים לשרת ולמנוע, כמו: מידע על תוצאת הצפנה, מידע על קוד המכונה, מידע אודות סטטוס התחרות ועוד.

מודול ה-Exceptions:

מודול המכיל את כל השגיאות המתקבלות בזמן ריצת התוכנית, כמו שגיאות העלאת קובץ על ידי ה-Uboat, שגיאות הצפנה ועוד.

מודול ה-UBoat:

ChatAreaUboat package: פקג' המכיל את הקונטרולר המשמש לאזור הציאט ורפרשר הפונה לשרת ומעדכן את ההודעות החדשות שנשלחו בציאט.

FirstPageUboat package:

- **FirstPageController**: קונטרולר ראשי למסך הראשון. מהווה "מנהל" של תת הקומפוננטות של אותו העמוד. מהווה מתווך בין הקונטרולרים של תתי הקומפוננטות שלו ובין ה-mainController.

CodeCalibration package:

- **CodeCalibrationController**: קונטרולר לקומפוננטה בה ניתן להגדיר את תצורת הקוד של המכונה. אוספת מהמשתמש את המידע על הקוד שרוצה לקבוע ומעבירה אותה במעלה שרשרת הדרגות אל ה-mainController המתקשר עם המנוע.

MachineDetails package:

- **MachineDetailsController**: קונטרולר לקומפוננטה המציגה את פרטי המכונה.
- LoginUboat package**: מכיל את הקונטרולר המשמש להתחברות שחקן מסוג UBoat למערכת. פונה לשרת ומבצע רישום של שחקן חדש.

MainControllerUboat package:

- **MainController**: הקונטרולר הראשי של המערכת. אחראי על הקשר שבין הקליינט לשרת.

- **SkinsOptions**: Enum שמכיל את המידע הנחוץ לטעינת כל skin למערכת (כמו מיקומי קבצי ה-css).
:Header package
- **HeaderController**: הקונטרולר של רכיב ה-header. מקבל בקשות מהמשתמש באשר להתמצאות בין העמודים וטעינת קובץ ומעביר את הבקשות ל-mainController. בעלת מטודות המאפשרות להציג הודעות שגיאה/הצלחה למשתמש.
:Intro package
- **IntroController**: הקונטרולר של העמוד הפותח של התוכנה. תפקידו להציג את התמונה המוצגת בעמוד.
:MainUboat package
- **Main**: מחלקת ה-main המכילה את המטודה main המפעילה את המערכת.
- **DTOParser**: מחלקה שתפקידה לקבל DTO שונים שהקונטרולר מקבל מהמנוע ולהחזירו בפורמט המתאים לתצוגה ע"י ה-UI (רשימה של string, map של string או string בודד).
:PropertiesUboat package
- **MachineConfigProperty**: מחלקה המכילה שדות property המכילים מידע על הקוד של המכונה. ה-properties השונים קשורים ל-labels השונים ב-UI, ובפרט בקומפוננטה של code config, המציגים את פרטי הקוד של המכונה.
- **MachineInfoProperty**: מחלקה המכילה שדות property המכילים מידע על המכונה. ה-properties השונים קשורים ל-labels השונים ב-UI המציגים את פרטי המכונה.
- **MachineInfoSupplier**: interface המחזיק מטודות לקבלת מידע אודות המכונה, ללא אפשרות לשנות אותו ובצורת משתנים פרימיטיביים (ולא properties).
:resource package
:SecondPage package
- **SecondPageController**: קונטרולר ראשי למסך השני. מהווה "מנהל" של תת הקומפוננטות של אותו העמוד. מהווה מתווך בין הקונטרולרים של תתי הקומפוננטות שלו ובין ה-mainController.
- **מכיל מספר רפרשרים המעדכנים מידע אודות התחרות כמו מועמדים וסטטוס התחרות.**
:encryptArea package
- **EncryptBruteForceController**: קונטרולר לתהליך ההצפנה של הברוט פורס. יכול לקבל מילה להצפנה בתיבת הטקסט או על ידי קבלתה מהמילון ומעביר הלאה במעלה ההיררכיה להצפנה במנוע.
:Dictionary package
- **DictionaryController**: קונטרולר לקומפוננטת המילון. בעל פונקציונליות לחיפוש מילה במילון.
- **DictionaryTrie**: מימוש למבנה הנתונים Trie המשמש לצורך המילון. בעל פונקציונליות של הוספת מילה וחיפוש מילה. מאפשר למצוא את כל המילים במילון המתחילות בתחילית מסוימת.
- **TrieNode**: מחלקת המשמשת למימוש ה-DictionaryTrie, מייצגת איבר יחיד בעץ.
:resultsArea package
- **ResultsController**: קונטרולר לטבלה המציגה את תוצאות ההצפנה של כל הצוותים המשתתפים בתחרות.
- **ResultsRow**: מחלקה המייצגת שורה בודדת בטבלת תוצאות ההצפנה.

TeamsData package: מכיל מחלקות השולטות על המידע בטבלת הצוותים המשתתפים בתחרות (כמו קונטרולר לטבלה ופרפרשר הפונה לשרת ומביא את המידע העדכני).
UtilsUboat package: מכיל את כל המטודות והעזרים הנחוצים לתפקוד האפליקציה, כמו constants, cookieManager ועוד.

מודול ה-Allies:

ChatAreaAllies package: פקג' המכיל את הקונטרולר המשמש לאזור הציאט ופרפרשר הפונה לשרת ומעדכן את ההודעות החדשות שנשלחו בציאט.

BruteForceAllies package: מכיל מחלקת **BruteForceManager** האחראי על התנעת פעולת ה-bruteForce בשרת ומעקב אחריה.

FirstPageAllies package:

- **FirstPageController**: קונטרולר ראשי למסך הראשון. מהווה "מנהל" של תת הקומפוננטות של אותו העמוד. מהווה מתווך בין הקונטרולרים של תתי הקומפוננטות שלו ובין ה-mainController.

AgentsTable package: מכיל מחלקות השולטות על המידע המוצג בטבלת הסוכנים הרשומים ל-allie (כמו קונטרולר לטבלה ופרפרשר הפונה לשרת ומביא את המידע העדכני).

ContestTable package: מכיל מחלקות השולטות על המידע המוצג בטבלת כלל התחרויות ומאפשר רישום לתחרות חדשה ע"י יצירת פניה לשרת. (מכיל בין היתר קונטרולר לטבלה ופרפרשר הפונה לשרת ומביא את המידע העדכני).

LoginAllies package: מכיל את הקונטרולר המשמש להתחברות שחקן מסוג Allie למערכת. פונה לשרת ומבצע רישום של שחקן חדש.

MainAllies package:

- **Main**: מחלקת ה-main המכילה את המטודה main המפעילה את המערכת.

- **DTOParser**: מחלקה שתפקידה לקבל DTO שונים שהקונטרולר מקבל מהמנוע ולהחזירו בפורמט המתאים לתצוגה ע"י ה-UI (רשימה של string, map של string או string בודד).

MainControllerAllies package:

- **MainController**: הקונטרולר הראשי של המערכת. אחראי על הקשר שבין הקליינט לשרת.

- **SkinsOptions**: Enum שמכיל את המידע הנחוץ לטעינת כל skin למערכת (כמו מיקומי קבצי ה-css).

Header package:

- **HeaderController**: הקונטרולר של רכיב ה-header. בעל מטודות המאפשרות להציג הודעות שגיהא/הצלחה למשתמש.

resource package: מכיל את כל המשאבים הנחוצים לטעינת ה-skins השונים.

SecondPage package:

- **SecondPageController**: קונטרולר ראשי למסך השני. מהווה "מנהל" של תת הקומפוננטות של אותו העמוד. מהווה מתווך בין הקונטרולרים של תתי הקומפוננטות שלו ובין ה-mainController.

- **מכיל מספר פרפרשרים המעדכנים מידע אודות התחרות כמו מועמדים וסטטוס התחרות.**

ContestData package: מכיל קונטרולר עבור הקומפוננטה המציגה את המידע אודות התחרות אליה רשום המשתמש.

TeamsCandidates package:

- **TeamsCandidatesController**: קונטרולר לטבלה המציגה את תוצאות ההצפנה של כל הצוותים המשתתפים בתחרות.
- **CandidatesRow**: מחלקה המייצגת שורה בודדת בטבלת תוצאות ההצפנה.
- TeamsData package: מכיל מחלקות השולטות על המידע בטבלת הצוותים המשתתפים בתחרות (כמו קונטרולר לטבלה ופרפרשר הפונה לשרת ומביא את המידע העדכני).
- TeamsProgress package: מכיל מחלקות השולטות על המידע אודות התקדמות הסוכנים השונים במלאכת ההצפנה. (כמו קונטרולר לקומפוננטות ופרפרשר המביא את המידע העדכני מהשרת).
- UtilsAllies package: מכיל את כל המטודות והעזרים הנחוצים לתפקוד האפליקציה, כמו constants, cookieManager ועוד.

מודול ה-Agent:

- ChatAreaAgent package: פקג' המכיל את הקונטרולר המשמש לאזור הציאט ופרפרשר הפונה לשרת ומעדכן את ההודעות החדשות שנשלחו בציאט.
- AgentBruteForce package
- **ThreadPoolManager**: מחלקת המנהלת את חלוקת המשימות לטרדים, הבאת משימות מהשרת ועדכון על מועמדים שנמצאו.
- **Decryption Task**: מחלקה זו מגדירה משימת פענוח בודדת המבוצעת על ידי thread מה-thread pool. מוגדרת בה משימת פענוח: מעבר על קבוצת קודים, מציאת מועמדות לפענוח ודיווח לשרת.
- **ABCCounter**: מחלקה שמאפשרת למצוא את מיקום הרוטורים העוכב. מבצעת ספירת בבסיס L (גודל ה-ABC של המכונה). מאפשרת לקדם את מיקום הרוטורים ולמצוא מיקום בהינתן אינדקס.
- **AgentProgressInfo**: מחלקה המכילה מידע אודות התקדמות מלאכת הפענוח. מכיל שדות כמו סה"כ המשימות, כמות המשימות שבוצעו ועוד.
- **FinishedTaskData**: מחלקה המכילה מידע לגבי המשימה שהסתיימה.
- EnigmaMachineAgent package: מופע של מכונת האניגמה. פקג' זהה לזה המופיע במודול ה-engine.
- MainAgents package
- **Main**: מחלקת ה-main המכילה את המטודה main המפעילה את המערכת.
- **DTOParser**: מחלקה שתפקידה לקבל DTO שונים שהקונטרולר מקבל מהמנוע ולהחזירו בפורמט המתאים לתצוגה ע"י ה-UI (רשימה של string, map של string או string בודד).
- MainControllerAgents package
- **MainController**: הקונטרולר הראשי של המערכת. אחראי על הקשר שבין הקליינט לשרת.
- **SkinsOptions**: Enum שמכיל את המידע הנחוץ לטעינת כל skin למערכת (כמו מיקומי קבצי ה-css).
- **מכיל מספר רפרשרים המעדכנים מידע אודות התחרות כמו מועמדים וסטטוס התחרות.**
- ContestData package: מכיל קונטרולר ופרפרשר עבור הקומפוננטה המציגה את המידע אודות התחרות אליה רשום המשתמש.
- Candidates package
- **CandidatesTableController**: קונטרולר לטבלה המציגה את תוצאות ההצפנה.

- **CandidatesRow**: מחלקה המייצגת שורה בודדת בטבלת תוצאות הצפנה.
Progress package: מכיל את הקונטרולר לקומפוננטת ה-progress.
resource package: מכיל את כל המשאבים הנחוצים לטעינת ה-skins השונים.
UtilsAgents package: מכיל את כל המטודות והעזרים הנחוצים לתפקוד האפליקציה, כמו constants, cookieManager ועוד.

בחירות שקיבלנו בעת מימוש המערכת

- בחרנו להציג את כל השגיאות, הודעות ההצלחה והודעות הקשורות לסטטוס התחרות ב-header.