Enigma Ex2 By :

Ben Cohen 304965643 Bencohen3@gmail.com

Eden Hartman 313526105 [edenha@mta.ac.il](mailto:edenha@mta.ac.il)

Special Notes :   
We create all missions at the start of the decryption process so in case of more than 1,000,000+ tasks it can take some time (few seconds).

**Module : EnigmaUI**

**Package : UI**

* UIManager :   
  Responsible for displaying the user menu, receiving data, and displaying the outputs. contains UIEnigmaProfile object.
* UIEnigmaProfile :   
  Object of the UI. Responsible for keeping all machine details. Contains InitialCodeConfiguration.
* InitialCodeConfiguration :  
   Object of the UI. Responsible for keeping all machine Configuration.
* RomanDigit (enum) :  
  Object of the UI. Allows a simple transition between the rotors ID and their representation in Roman numerals.

**Module : EnigmaEnigne Dependency(EnigmaMachineFactory)**

**Package : Logic**

* EngineManager :  
   The App manager in charge of all logic operations and validation. Holds an EnigmaMachine for processing strings, EnigmaComponenetFactory for creating the machine and a StatsManager for holding all stats.
* StatsManager:  
  In charge of holding all of the current machine stats  
  Using the following objects :
* CodeFormat which represent a machine's "secret".
* CodedStrings which represent a machine's output and input history.

**Module : EnigmaMachineFactory**

**Package : EnigmaMachineFactory - Contains all off the Machine objects.**

* EnigmaComponentFactory (Interface):  
  An interface for creating a machine (Facotry Design Pattern). Its implementation – EnigmaComponentFactoryImpl
* EnigmaMachine (Interface):  
   An interface for using a machine. Its implementation – EnigmaMachineImpl
* EnigmaMachineBuilder:  
   A bulider for the machine with its parts (Builder Design Pattern)
* JAXBToActual (Singleton):  
  In charge of transforming a machine created by the java JAXB feature to our Actual machine.
* Secret (Interface):  
   Represent the machine "secret" which is the chosen rotors, their positions and the chosen reflector. With its implementation - SecretImpl
* SecretBuilder:
* A builder for the Secret object (Builder Design Pattern).

**Package : JAXBGenerated**

Contains the auto generated JAXB classes from the Enigma.xsd

**Package : Actual**

Contains our actual inner machine classes.  
 Objects such as : Enigma, Machine, Mapping, Reflect, Reflector and Rotor

**Module : DecryptionManager Dependency(EnigmaMachineFactory, AgentModule)**

**Package : DecryptionManager -** Manages all of the deciphering mission

* DecipherManager:  
  The deciphering manager, in charge of the mission and all of the agents
* DecipherMission:Represent a mission for deciphering
* DifficultyLevel(Enum):  
  4 levels of Difficulty

**Module : AgentModule**

**Package : AgentModule - Contains all off the Agent related objects**.

* Agent:  
  Extends thread, gets a block of AgentTasks, deciphers them, check them with the dictonary and return results to DM
* AgentFactory:  
  Resposnble for creating new agents
* AgentResponse:  
  An object for representing an agent response to the DM
* AgentTask:  
  Represent an agent task i.e a starting secret and a size to run on.
* CandidateForDecoding:  
  1 code after deciphering it with its decipheing agent and it's secret
* DecipheringStatus:  
  Object to sync the agents with the DM. i.e controls the pause\play\stop options from the DM