

Reflection of Final Project

Subject: COMP90041

Name: Shiyang Chen

ID: 931880

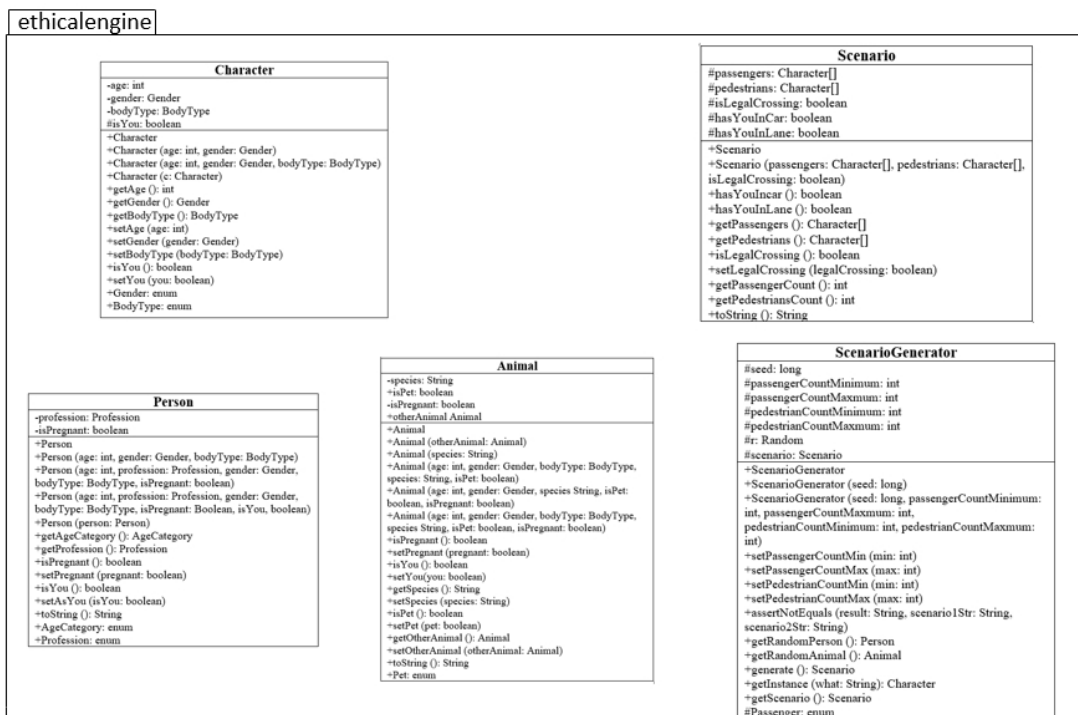
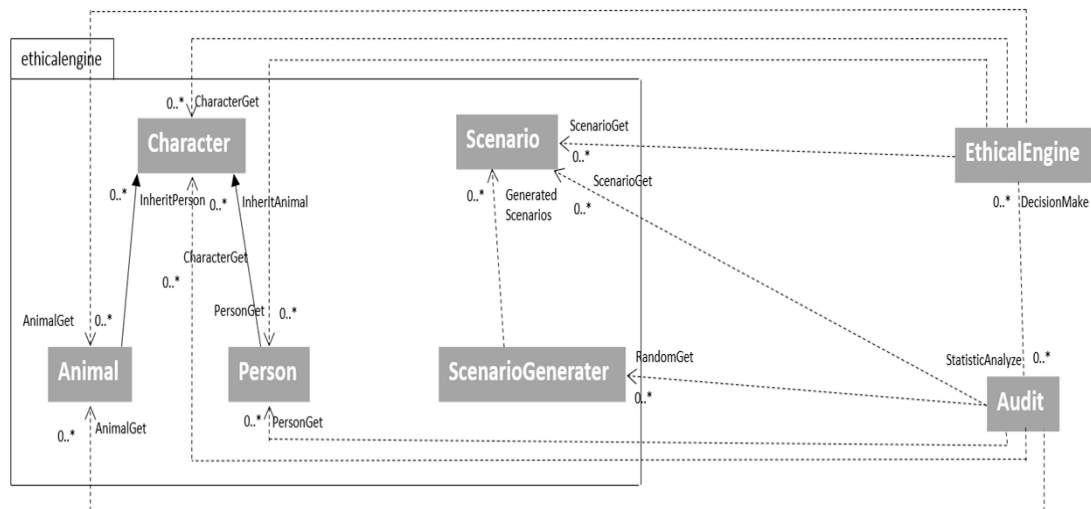
Through this final project, I applied the java knowledge I learned in class to practice. The purpose of this project is to establish a moral engine, let the system make sacrifices and judgments based on the number and characteristics of creatures, and save some but not all of them.

The ethical engine, which is the core part, in this system obtains weighted coefficients of pedestrians and passengers by judging whether the pedestrians is legal, whether the creatures in the scenario is pregnant, age, etc. should be selected to be saved.

However, there is still a gap between the judgment made by this engine and the actual situation in life, or the optimal solution. For example, in the real life, a car accident is not just a problem for both parties to survive. In a car accident, the correct handling may save more lives than just the lives of pedestrians or passengers. In this system, due to the limited data types (characters) of the scenario, the system cannot cover information such as family relations and personal insurance plans. In practical applications, such as autonomous driving, a larger database can be referenced to allow the moral engine to make less sacrificed judgments.

In the different test, there is an “incorrect” in the output. After careful comparison, I found no difference between the actual output and the expected output. By consulting the information, I found that the cause of incorrect may be the difference between Unix (LF) and Windows (CR LF) which will cause different file lengths. So making the system keep consistent in different environments is also a problem to be considered in future development programs. However, at the final test, I finally found there are two version of welcome.ascii.

UML:



Next page ctd..

| Audit | |
|---|--|
| #auditType: String #auditType!: AuditType #num: int #personNum: double #personNumSurvive: double #animalNum: double #animalNumSurvive: double #doctorNum: double #doctorNumSurvive: double #scoutNum: double #scoutNumSurvive: double #criminalNum: double #criminalNumSurvive: double #homelessNum: double #homelessNumSurvive: double #unemployedNum: double #unemployedNumSurvive: double #unknownNum: double #unknownNumSurvive: double #mouseNum: double #mouseNumSurvive: double #recreantNum: double #recreantNumSurvive: double #playerNum: double #playerNumSurvive: double #babyNum: double #babyNumSurvive: double #childNum: double #childNumSurvive: double #adultNum: double #adultNumSurvive: double #seniorNum: double #seniorNumSurvive: double #maleNum: double #maleNumSurvive: double #femaleNum: double #femaleNumSurvive: double #allNum: double #ageNumSurvive: double #genderNum: double #genderNumSurvive: double #bodyNum: double #bodyNumSurvive: double #pregnantNum: double #pregnantNumSurvive: double #speciesNum: double #speciesNumSurvive: double #isYou: boolean #isYouSurvive: boolean #setAge (age: int) #setGender (gender: Gender) #setBodyType (bodyType: BodyType) #pregnant (scenarioTemp: Scenario) #species (species: String) #isYou (isYou: boolean) #setYou (you: boolean) #Gender: enum #BodyType: enum | #recreantNum: double #recreantNumSurvive: double #scoutNum: double #scoutNumSurvive: double #animalNum: double #animalNumSurvive: double #doctorNum: double #doctorNumSurvive: double #algorithmOrUser: String #algorithmOrUser!: String #audit -Audit (scenario: Scenario) -run (scenario: Scenario, survivorDecision: EthicalEngineDecision) -run -run (num: int) -setAuditType (auditType: String) -getAuditType (): String -toString (): String -printStatistics() -printToFile (filepath: String) -printToFileForLog (filepath: String) -savePredictionsOrPassenger (scenario: Scenario, filepath: String, flag: char) -setAttributes (characters: Character[]) -addProfession (profession: Person.Profession) -addProfessionSurvive (profession: Person.Profession) -addAgeCategory (ageCategory: Person.AgeCategory) -addAgeCategorySurvive (ageCategory: Person.AgeCategory) -addGender (gender: Character.Gender) -addGenderSurvive (gender: Character.Gender) -addBodyType (bodyType: Character.BodyType) -addBodyTypeSurvive (bodyType: Character.BodyType) -addPregnant (person: Character) -addPregnantSurvive (person: Character) -addSpecies (species: String) -addSpeciesSurvive (species: String) -addIsYou (isYou: boolean) -addSetSurvive (isYou: boolean) -greenOrRed (scenarioTemp: Scenario) -greenOrRedSurvive (scenarioTemp: Scenario, num: int) -isYou (isYou: boolean) -isYouSurvive (isYou: boolean) -addNum (num: Double): double -AuditType: enum |

| EthicalEngine |
|---|
| #input: Scanner +main (args: String[]) +weatherToSave (isResults: boolean): boolean +interactiveWithUsers() +isInteractive (argument: String): boolean +isHelp (argument: String): boolean +isConfig (argument: String): boolean +isResults (argument: String): boolean +getInteractiveAuto () +decide (scenario: Scenario): Decision +getDifferentstatic (characters: Character[]): double[] +getScenarios (filepath: String, scenarioFlag: int[]): Scenario[] +getCount (path: String): int +getTempPerson (temp: String[], line: int): Person +getTempAnimal (temp: String[], line: int): Animal +getScenarioLineCount (path: String): int[] +getHelp +Decision: enum #InvalidDataFormatException #NumberFormatException #InvalidCharacteristicException |

Appendix

| Character |
|--|
| -age: int -gender: Gender -bodyType: BodyType #isYou: boolean |
| +Character +Character (age: int, gender: Gender) +Character (age: int, gender: Gender, bodyType: BodyType) +Character (c: Character) +getAge (): int +getGender (): Gender +getBodyType (): BodyType +setAge (age: int) +setGender (gender: Gender) +setBodyType (bodyType: BodyType) +isYou (): boolean +setYou (you: boolean) +Gender: enum +BodyType: enum |

| Person |
|---|
| -profession: Profession -isPregnant: boolean |
| +Person +Person (age: int, gender: Gender, bodyType: BodyType) +Person (age: int, profession: Profession, gender: Gender, bodyType: BodyType, isPregnant: boolean) +Person (age: int, profession: Profession, gender: Gender, bodyType: BodyType, isPregnant: Boolean, isYou, boolean) +Person (person: Person) +getAgeCategory (): AgeCategory +getProfession (): Profession +isPregnant (): boolean +setPregnant (pregnant: boolean) +isYou (): boolean +setAsYou (isYou: boolean) +toString (): String +AgeCategory: enum +Profession: enum |

| Animal |
|--|
| -species: String +isPet: boolean -isPregnant: boolean +otherAnimal Animal |
| +Animal +Animal (otherAnimal: Animal) +Animal (species: String) +Animal (age: int, gender: Gender, bodyType: BodyType, species: String, isPet: boolean) +Animal (age: int, gender: Gender, species String, isPet: boolean, isPregnant: boolean) +Animal (age: int, gender: Gender, bodyType: BodyType, species String, isPet: boolean, isPregnant: boolean) +isPregnant (): boolean +setPregnant (pregnant: boolean) +isYou (): boolean +setYou(you: boolean) +getSpecies (): String +setSpecies (species: String) +isPet (): boolean +setPet (pet: boolean) +getOtherAnimal (): Animal +setOtherAnimal (otherAnimal: Animal) +toString (): String +Pet: enum |

| Scenario |
|--|
| #passengers: Character[] #pedestrians: Character[] #isLegalCrossing: boolean #hasYouInCar: boolean #hasYouInLane: boolean |
| +Scenario +Scenario (passengers: Character[], pedestrians: Character[], isLegalCrossing: boolean) +hasYouIncar (): boolean +hasYouInLane (): boolean +getPassengers (): Character[] +getPedestrians (): Character[] +isLegalCrossing (): boolean +setLegalCrossing (legalCrossing: boolean) +getPassengerCount (): int +getPedestriansCount (): int +toString (): String |

| ScenarioGenerator |
|--|
| #seed: long #passengerCountMinimum: int #passengerCountMaxmum: int #pedestrianCountMinimum: int #pedestrianCountMaxmum: int #r: Random #scenario: Scenario |
| +ScenarioGenerator +ScenarioGenerator (seed: long) +ScenarioGenerator (seed: long, passengerCountMinimum: int, passengerCountMaxmum: int, pedestrianCountMinimum: int, pedestrianCountMaxmum: int) +setPassengerCountMin (min: int) +setPassengerCountMax (max: int) +setPedestrianCountMin (min: int) +setPedestrianCountMax (max: int) +assertNotEquals (result: String, scenario1Str: String, scenario2Str: String) +getRandomPerson (): Person +getRandomAnimal (): Animal +generate (): Scenario +getInstance (what: String): Character +getScenario (): Scenario #Passenger: enum |

| Audit |
|--|
| #auditType: String #auditType1: AuditType #runs: int #personNum: double #personNumSurvive: double #animalNum: double #animalNumSurvive: double #doctorNum: double #doctorNumSurvive: double #ceoNum: double #ceoNumSurvive: double #criminalNum: double #criminalNumSurvive: double #homelessNum: double #homelessNumSurvive: double #unemployedNum: double #unemployedNumSurvive: double #unknownNum: double #unknownNumSurvive: double #noneNum: double #noneNumSurvive: double #retirementNum: double #retirementNumSurvive: double #layerNum: double #layerNumSurvive: double #babyNum: double #babyNumSurvive: double #childNum: double #childNumSurvive: double #adultNum: double #adultNumSurvive: double #seniorNum: double #seniorNumSurvive: double #maleNum: double #maleNumSurvive: double #femaleNum: double #femaleNumSurvive: double #allNum: double #ageNumSurvive: double #greenNum: double #greenNumSurvive: double #redNum: double #redNumSurvive: double #averageNum: double #averageNumSurvive: double #athleticNum: double #athleticNumSurvive: double #overweightNum: double #overweightNumSurvive: double #unspecifiedNum: double #unspecifiedNumSurvive: double #pregnantNum: double #pregnantNumSurvive: double #dogNum: double #dogNumSurvive: double #catNum: double #catNumSurvive: double #birdNum: double #birdNumSurvive: double |

```

#ferretNum: double
#ferretNumSurvive: double
#petNum: double
#petNumSurvive: double
#youNum: double
#youNumSurvive: double
#scenarios: Scenario[]
#algorithmOrUser: String

+getAlgorithmOrUser (): String
+setAlgorithmOrUser (algorithmOrUser: String)
+Audit
+Audit (scenarios: Scenario[])
+run (scenario: Scenario, survivorDecision:
EthicalEngine.Decision)
+run
+run (runs: int)
+setAuditType (auditType: String)
+getAuditType (): String
+toString (): String
+printStatistic()
+printToFile (filepath: String)
+printToFileForlogs (filepath: String)
+savePedestriansOrPassengers (scenario: Scenario, filePath: String,
flag: char)
+setAttributesSurvive (characters: Character[])
+setAttributes (characters: Character[])
+addProfession (profession: Person.Profession)
+addProfessionSurvive (profession: Person.Profession)
+addAgeCategory (ageCategory: Person.AgeCategory)
+addAgeCategorySurvive (ageCategory: Person.AgeCategory)
+addGender (gender: Character.Gender)
+addGenderSurvive (gender: Character.Gender)
+addBodyType (bodyType: Character.BodyType)
+addBodyTypeSurvive (bodyType: Character.BodyType)
+addPregnant (person: Character)
+addPregnantSurvive (person: Character)
+addSpecies (species: String)
+addSpeciesSurvive (species: String)
+addPet (ispet: boolean)
+addPetSurvive (ispet: boolean)
+greenOrRed (scenarioTemp: Scenario)
+greenOrRedSurvive (scenarioTemp: Scenario, num: int)
+isYou (isYou: boolean)
+isYouSurvive (isYou: boolean)
+addNum (num: Double): double
+AuditType: enum

```

| EthicalEngine |
|---|
| <div data-bbox="240 230 999 264">#input: Scanner</div> <div data-bbox="240 264 999 875"> <div data-bbox="240 264 999 297">+main (args: String[])</div> <div data-bbox="240 297 999 331">+weatherToSave (isResults: boolean): boolean</div> <div data-bbox="240 331 999 365">+interactiveWithUsers()</div> <div data-bbox="240 365 999 398">+isInteractive (argument: String): boolean</div> <div data-bbox="240 398 999 432">+isHelp (argument: String): boolean</div> <div data-bbox="240 432 999 465">+isConfig (argument: String): boolean</div> <div data-bbox="240 465 999 499">+isResults (argument: String): boolean</div> <div data-bbox="240 499 999 533">+getInteractiveAuto ()</div> <div data-bbox="240 533 999 566">+decide (scenario: Scenario): Decision</div> <div data-bbox="240 566 999 600">+getDifferentstatic (characters: Character[]): double[]</div> <div data-bbox="240 600 999 633">+getScenarios (filepath: String, scenarioFlag: int[]): Scenario[]</div> <div data-bbox="240 633 999 667">+getCount (path: String): int</div> <div data-bbox="240 667 999 701">+getTempPerson (temp: String[], line: int): Person</div> <div data-bbox="240 701 999 734">+getTempAnimal (temp: String[], line: int): Animal</div> <div data-bbox="240 734 999 768">+getScenarioLineCount (path: String): int[]</div> <div data-bbox="240 768 999 801">+getHelp</div> <div data-bbox="240 801 999 835">+Decision: enum</div> <div data-bbox="240 835 999 869">#InvalidDataFormatException</div> <div data-bbox="240 869 999 902">#NumberFormatException</div> <div data-bbox="240 902 999 936">#InvalidCharacteristicException</div> </div> |