COMP90041 Programming and Software Development
Final Project
UML Class Diagram

Ebru Soezbir 1135390 nsoezbir | June 25, 2020

package ethicalengine

<<enumeration>> Gender

FEMALE
MALE
UNKNOWN

<<enumeration>> BodyType

AVERAGE
ATHLETIC
OVERWEIGHT
UNSPECIFIED

Character

- age: int
- gender: Gender
- bodyType: BodyType
...

+ Character()
+ Character(age: int, gender: Gender, bodyType: BodyType)
+ Character(otherCharacter: Character)
+ getAge(): int
+ getGender(): Gender
+ getBodyType(): BodyType
+ setAge(age: int): void
+ setGender(gender: Gender): void
+ setBodyType(bodyType: BodyType): void

Person

- isPregnant: boolean
- isYou: boolean
- profession: Profession
- ageCategory: AgeCategory
...

+ Person()
+ Person(age: int, gender: Gender, bodyType: BodyType)
+ Person(age: int, profession: Profession, gender: Gender, bodyType: BodyType, isPregnant: boolean)
+ Person(otherPerson: Person)
+ isPregnant(): boolean
+ setPregnant(pregnant: boolean): void
+ isYou(): boolean
+ setAsYou(isYou: boolean): void
+ getAgeCategory(age: int): AgeCategory
+ getAgeCategory(): AgeCategory
+ getProfession(): Profession
+ toString(): String

<<enumeration>> Profession

DOCTOR
CEO
CRIMINAL
HOMELESS
UNEMPLOYED
TEACHER
STUDENT
UNKNOWN
NONE
PROFESSOR

<<enumeration>> AgeCategory

BABY
CHILD
ADULT
SENIOR

Animal

- species: String
- isPet: boolean
...

+ Animal()
+ Animal(species: String)
+ Animal(species: String, isPet: boolean)
+ Animal(age: int, gender: Gender, bodyType: BodyType, species: String, isPet: boolean)
+ Animal(otherAnimal: Animal)
+ getSpecies(): String
+ setSpecies(species: String): void
+ isPet(): boolean
+ setPet(isPet: boolean): void
+ toString(): String

contains 0..*

Scenario

- passengers: ArrayList<Character>
- pedestrians: ArrayList<Character>
- isLegalCrossing: boolean

+ hasYouInCar(): boolean
+ hasYouInLane(): boolean
+ getPassengers(): Character[]
+ getPedestrians(): Character[]
+ isLegalCrossing(): boolean
+ setLegalCrossing(isLegalCrossing: boolean): void
+ getPassengerCount(): int
+ getPedestrianCount(): int
+ toString(): String

contains 0..*

generates 0..*

ScenarioGenerator

- random: Random
- passengerCountMinimum: int
- passengerCountMaximum: int
- pedestrianCountMinimum: int
- pedestrianCountMaximum: int
...

+ ScenarioGenerator()
+ ScenarioGenerator(seed: long)
+ ScenarioGenerator(seed: long, passengerCountMinimum: int, passengerCountMaximum: int, pedestrianCountMinimum: int, pedestrianCountMaximum: int)
+ setPassengerCountMin(min: int): void
+ setPassengerCountMax(max: int): void
+ setPedestrianCountMin(min: int): void
+ setPedestrianCountMax(max: int): void
+ getRandomPerson(): Person
+ getRandomAnimal(): Animal
+ generate(): Scenario

MyPair

- name: String
- value: double

+ MyPair(name: String, value: double)
+ getName(): String
+ getValue(): double
+ compareTo(otherPair: MyPair)

contains 1..*

decides 0..*

runs 1..*

EthicalEngine

...

+ decide(scenario: Scenario): Decision
+ extractPerson(dataRow: String[], lineCounter: int): Person
+ extractAnimal(dataRow: String[], lineCounter: int): Animal
+ readConfigFile(filepath: String): ArrayList<Scenario>
...

generates 1..*

<<enumeration>> Decision

PEDESTRIANS
PASSANGERS

Audit

- auditType: String
- runs: int
- scenarioList: ArrayList<Scenario>
- survivorsReport: HashMap<String, Integer>
- deathsReport: HashMap<String, Integer>
...

+ Audit()
+ Audit(scenarioList: ArrayList<Scenario>)
+ setAuditType(auditTypeName: String): void
+ getAuditType(): String
+ run(newRuns: int): void
+ run(): void
+ toString(): String
+ printStatistic(): void
+ printToFile(filepath: String): void

COMP90041 Programming and Software Development
Final Project (Bonus Task)
Ebru Soezbir 1135390 nsoezbir

The trolley problem is a thought experiment which forces us to think about how to choose when there are no good choices. Here is the problem: Imagine a trolley which is out of control running down a track straight towards five workers who can't escape. You have the chance to flip a switch to divert the trolley on to a second track which has a worker on it too but just one. What do you do? Do you sacrifice one person to save five? For most people it might be reasonable to flip the switch letting one worker die to save five. This decision is consistent with the philosophical principal Utilitarianism which argues that the morally correct decision is the one that maximizes the well-being for the greatest number of people. However, people don't always take the utiliarity in view which we can seen by changing the trolley problem a bit. How would you choose if your life was at stake? Would you rather sacrifice two adults to save one child?
If I buy an autonomous vehicle the most important point would be to save my life at any cost. Hence, isYou has the highest priority in my decision-making algorithm. Moreover, life of babies and children are weighted more than adults and seniors. The algorithm also takes the ratio of passengers and pedestrians into account. Animals have less influence on the decision as well as if crossing the lane is legal or not.