

sensorLengthSlider: Slider[0..1]

carSpeed: TextField [0..1]
colorSelector: Circle[0..1]
deadCars: ArrayList<Car> [0..*]
exit: Button [0..1]
fitnessList: ArrayList<Integer> [0..*]
gen: TextField [0..1]
greenSlider: Label [0..1]
intersections: ArrayList<Shape> [0..*]
lastCar: Car [0..1]
layers: int[] [0..*]
menuBar: MenuBar [0..1]
mutRate: TextField [0..1]
neuralDisplay: NeuralDisplay [0..1]
neuronsPerLayerLabel: Label [0..1]
neuronsPerLayerTextField: TextField [0..1]
noCars: TextField [0..1]
pause: boolean [1]
play: Button [0..1]
player: MediaPlayer [0..1]
recommended: Button [0..1]
redSlider: Label [0..1]
reset: Button [0..1]
root: BorderPane [0..1]
save: Button [0..1]
sliderBlue: Slider [0..1]
sliderGreen: Slider [0..1]
sliderRed: Slider [0..1]
start: Button [0..1]
started: BooleanProperty [0..1]
timer: AnimationTimer [0..1]
title: Label [0..1]
userInterface: GridPane [0..1]
volumeSlider: Slider [0..1]
volumeSliderImage: ImageView [0..1]

+ addFitness(): void
applySettings(): void
- calculateDistanceToPoint(sensor: Line,
x: double. v: double): double

- activations: double
- neuralNetwork: NeuralNetwork
- layers: HiddenLayer[]
- nbLayers: int
- lineWeights: ArrayList<double>
- neuronList: ArrayList<Neuron>
+ timer: AnimationTimer
+ NeuralDisplay
- generateNeurons()
- generateWeights()
+ getDisplayedCar()
- update(): void

WelcomeController

- welcome: Label

- start: Button

- exit: Button

- volumeSlider: Slider

- player: MediaPlayer

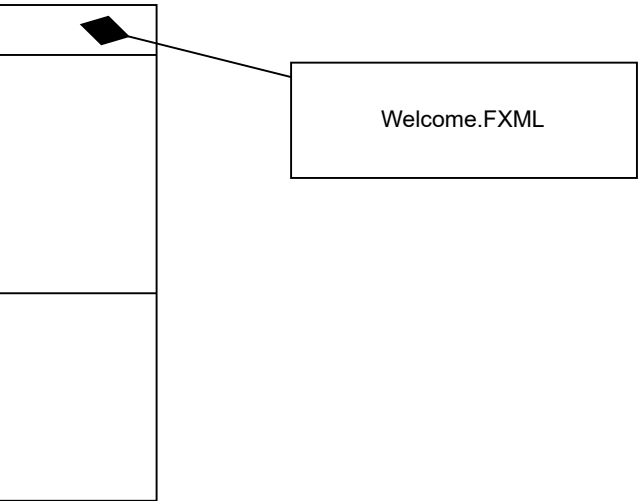
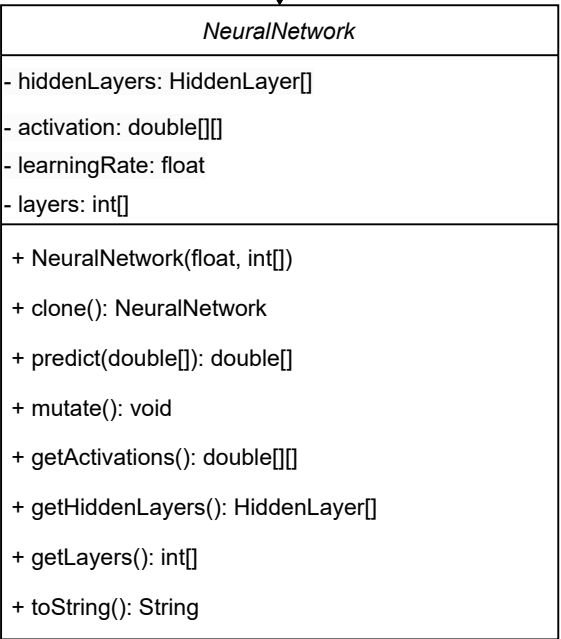
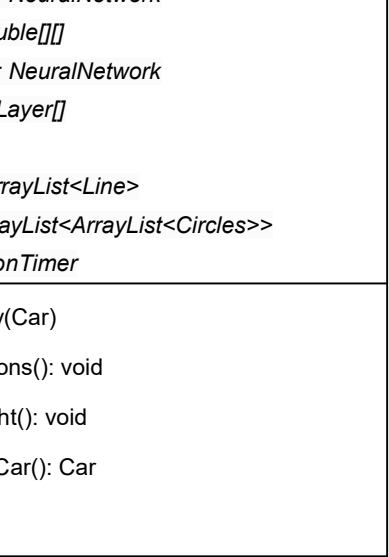
- simulation: Simulation

+ initialize(URL, ResourceBundle): void

+ getPlayer(): MediaPlayer

+ changeVolume(Slider, Media): void

+ highlightButton(Button): void



```

        -calculateSensorsLength(): void

        - checkCarCollision(): void
+ checkCollisionWithWall(order: int, car:
    Car): double

        -checkIfAllCarsDead(): void

        - checkMutateValue(): void

        - calculateSensorsLength(): void

        -checkTextInputs(e: KeyEvent, select:
        TextField, noCars: TextField, mutRate:
        TextField, carSpeed: TextField, ang
        Velocity: TextField, save: Button, start:
        Button): void

        -checkTextInputsComma(e: KeyEvent,
        select: TextField, no Cars: TextField,
        mutRate: TextField, carSpeed: TextField,
        ang Velocity: TextField, save: Button,
        start: Button): void

        -checkTextInputsDecimal(e: KeyEvent,
        select: TextField, no Cars: TextField,
        mutRate: TextField, carSpeed: TextField,
        ang Velocity: TextField, save: Button,
        start: Button)
            : void

        #draw(): void

        + getRoot(): BorderPane

        - getVolumeSlider(): void

+ getSimulationInstance(): Simulation

        + getBorderList(): ArrayList<Shape>

        +highlightButton(button: Button): void

        -moveCar(): void

        - removeDeadCars(): void

        - showAboutWindow(): void

        -showAlert(title: String, content: String):
        void

        -showEndWindow(): void

        - setLengthSlider():void

        - setPlace(): void

        - setupMenuBar(): void

        - setCheckInputs(): void

        - setTrack(): void

        - setBorders(): void

        -setTitle(): void

        - setButtons(): void

        - setGenerationsOfCars(): void

        - setTextFieldNbCars(): void

```



```
- setTextFieldNbCars(): void
    - setColorBox(): void
    - setTextFieldCarSpeed(): void
- setTextFieldAngularVelocity(): void
    - setTextFieldMutRate(): void
    - setColorSelector(): void
    - setPlayer(): void
    - setupSliders(): void
    - setUserInterface(): void
    - setCarPane(): void
    - setBackgroundBox()
    + start(): void
+ setupNextGeneration(): void
-showNeuralDisplay(car: Car)
```

