

Agents Of Steer

1.5

Generated by Doxygen 1.8.13

Contents

1	Hierarchical Index	1
1.1	Class Hierarchy	1
2	Data Structure Index	3
2.1	Data Structures	3
3	Data Structure Documentation	5
3.1	AgentController Class Reference	5
3.1.1	Detailed Description	6
3.1.2	Member Function Documentation	6
3.1.2.1	GetPolicing()	6
3.1.2.2	GetVelocity()	7
3.1.2.3	IsAPolice()	7
3.1.2.4	isReversing()	7
3.1.2.5	SetNewTarget()	7
3.1.2.6	SetPolicing()	7
3.1.2.7	SomethingDetected()	8
3.1.2.8	SteerVector()	8
3.1.3	Field Documentation	8
3.1.3.1	limitersAndInfluencers	8
3.1.3.2	reverseControl	8
3.1.3.3	sensors	8
3.1.3.4	showDebugRays	9
3.1.3.5	strengths	9

3.1.3.6	target	9
3.1.3.7	useSensorsOn	9
3.1.3.8	weights	9
3.1.3.9	whatIsaWall	9
3.2	AgentHUDInfo Class Reference	9
3.2.1	Member Function Documentation	10
3.2.1.1	DisableHUD()	10
3.2.1.2	EnableHUD()	10
3.2.2	Field Documentation	10
3.2.2.1	agentID	11
3.2.2.2	lapDisplay	11
3.2.2.3	positionDisplay	11
3.2.2.4	speedOMeter	11
3.3	AgentPathCreator Class Reference	11
3.3.1	Detailed Description	12
3.3.2	Member Function Documentation	12
3.3.2.1	AddNewAgent()	12
3.3.2.2	GetPath()	12
3.3.2.3	PlayerSpawned()	13
3.3.2.4	StartTheRace()	13
3.3.3	Field Documentation	13
3.3.3.1	agents	13
3.3.3.2	circuit	13
3.3.3.3	LaunchAtStart	13
3.3.3.4	oneWay	13
3.3.3.5	oneWaySkipPoint	14
3.3.3.6	pathPredictionMultiplier	14
3.3.3.7	pathResolution	14
3.3.3.8	policeAgents	14
3.3.3.9	radius	14

3.3.3.10	resetFixThreshold	14
3.4	AgentPolice Class Reference	14
3.4.1	Member Function Documentation	15
3.4.1.1	AreWePolicing()	15
3.4.2	Field Documentation	15
3.4.2.1	audioPlayer	15
3.4.2.2	evadedDistance	15
3.4.2.3	getTriggeredDistace	16
3.4.2.4	hindrences	16
3.4.2.5	patrolSpeed	16
3.4.2.6	startPatrolling	16
3.5	AgentProgressTab Class Reference	16
3.5.1	Member Function Documentation	17
3.5.1.1	GiveName()	17
3.5.1.2	UpdateProgress()	17
3.5.2	Field Documentation	17
3.5.2.1	myName	17
3.5.2.2	myPos	18
3.6	AgentRaceFinisher Class Reference	18
3.6.1	Member Function Documentation	18
3.6.1.1	OnTriggerEnter()	18
3.6.2	Field Documentation	18
3.6.2.1	myRaceManager	19
3.7	AgentRaceManager Class Reference	19
3.7.1	Member Enumeration Documentation	20
3.7.1.1	TriggerPlacementOption	20
3.7.2	Member Function Documentation	20
3.7.2.1	AgentFinishedTheLap()	20
3.7.2.2	GetPositionInRaceHierarchy()	21
3.7.2.3	GetTheDriver()	21

3.7.2.4	GetTheLapNumber()	21
3.7.2.5	HowManyLaps()	22
3.7.2.6	HowManyRacing()	22
3.7.2.7	InitiateRaceManager()	22
3.7.2.8	IsRaceInitiated()	23
3.7.2.9	SortTheRacingAgents()	23
3.7.2.10	UpdateAgentsCurrentLapPosition()	23
3.7.3	Field Documentation	23
3.7.3.1	agentNames	23
3.7.3.2	currentMainCamera	23
3.7.3.3	endPanel	24
3.7.3.4	finishTrigger	24
3.7.3.5	headPosInfo	24
3.7.3.6	mobileControlsHolder	24
3.7.3.7	progressContent	24
3.7.3.8	progressTab	24
3.7.3.9	progressTabForHuman	24
3.7.3.10	reverseFinishLineCheetingThreshold	24
3.7.3.11	whereIsTriggerIsPlaced	25
3.7.3.12	wholeProgressBoard	25
3.8	AgentRaceStarter Class Reference	25
3.8.1	Member Function Documentation	26
3.8.1.1	InitiateTheRaceStarter()	26
3.8.2	Field Documentation	26
3.8.2.1	audioPlayer	26
3.8.2.2	cameraToBeAssigned	26
3.8.2.3	carTypes	27
3.8.2.4	countSound	27
3.8.2.5	delay	27
3.8.2.6	initiateAtStart	27

3.8.2.7	playerSpawnPoint	27
3.8.2.8	policeHolder	27
3.8.2.9	spawnAbove	27
3.8.2.10	spawnPoints	27
3.8.2.11	startCountElements	28
3.8.2.12	ullInput	28
3.9	AgentRaceStarterInitiator Class Reference	28
3.9.1	Member Function Documentation	28
3.9.1.1	AssignVars()	28
3.10	AgentRaceManager.AgentsInRaceInfos Class Reference	29
3.10.1	Detailed Description	30
3.10.2	Member Function Documentation	30
3.10.2.1	AssignAgentController()	30
3.10.2.2	FinishTheLap()	30
3.10.2.3	NewLapProgress()	30
3.10.3	Field Documentation	30
3.10.3.1	agentID	31
3.10.3.2	agentsCurrentLapProgress	31
3.10.3.3	agentsLapNumber	31
3.10.3.4	agentsName	31
3.10.3.5	canFinishTheLap	31
3.10.3.6	disqualified	31
3.10.3.7	finished	31
3.10.3.8	headPosInfo	31
3.10.3.9	longestLegalTravel	32
3.10.3.10	myAgentController	32
3.10.3.11	myProgressTab	32
3.10.3.12	skipedFirstLap	32
3.11	AICarDriver Class Reference	32
3.11.1	Detailed Description	34

3.11.2	Member Function Documentation	34
3.11.2.1	ApplyLocalPositionToVisuals()	34
3.11.2.2	CurrentCalculatedSteer()	35
3.11.2.3	GetClampedVelocity()	35
3.11.2.4	GetCurrentSpeed()	35
3.11.2.5	GetGround()	35
3.11.2.6	IsReversing()	36
3.11.2.7	SetEngine()	36
3.11.2.8	SetInputX()	36
3.11.2.9	SetInputY()	36
3.11.2.10	SetNewControls()	36
3.11.3	Field Documentation	36
3.11.3.1	brakeColor	37
3.11.3.2	brakeSensitivity	37
3.11.3.3	canFlip	37
3.11.3.4	centerOfMass	37
3.11.3.5	centerOfMassAdditiveY	37
3.11.3.6	controlledByPlayer	37
3.11.3.7	controlType	37
3.11.3.8	flipSteerOnReverse	37
3.11.3.9	flipTimeOut	38
3.11.3.10	gears	38
3.11.3.11	handBrakeTorque	38
3.11.3.12	idleColor	38
3.11.3.13	maxReverseSpeed	38
3.11.3.14	maxSkidVertices	38
3.11.3.15	maxSpeed	38
3.11.3.16	maxSteerAngle	38
3.11.3.17	maxTorque	39
3.11.3.18	pressureOnClimbs	39

3.11.3.19 rearLights	39
3.11.3.20 reverseColor	39
3.11.3.21 skidMaterial	39
3.11.3.22 skidSmoke	39
3.11.3.23 skidSound	39
3.11.3.24 skidVertexDistance	39
3.11.3.25 slideAfterSlip	40
3.11.3.26 spark	40
3.11.3.27 speedRelativeSteer	40
3.11.3.28 steerResponsiveness	40
3.11.3.29 steerStabilityThreshold	40
3.11.3.30 wheels	40
3.11.3.31 wheelsForSpeedCalculation	40
3.12 AntiRoll Class Reference	41
3.12.1 Member Function Documentation	41
3.12.1.1 FixedUpdate()	41
3.12.2 Field Documentation	41
3.12.2.1 antiRoll	41
3.12.2.2 WheelL	42
3.12.2.3 WheelR	42
3.13 BillboardController Class Reference	42
3.13.1 Member Function Documentation	42
3.13.1.1 GiveCam()	42
3.13.2 Field Documentation	43
3.13.2.1 maxfadeDist	43
3.13.2.2 minFadeDist	43
3.14 CameraScript Class Reference	43
3.14.1 Detailed Description	44
3.14.2 Field Documentation	44
3.14.2.1 distance	44

3.14.2.2	height	44
3.14.2.3	heightDamping	44
3.14.2.4	rotationDamping	44
3.14.2.5	target	44
3.15	CameraTargetChanger Class Reference	44
3.15.1	Member Function Documentation	45
3.15.1.1	NextTarget()	45
3.15.1.2	PrevTarget()	45
3.15.2	Field Documentation	45
3.15.2.1	targets	45
3.16	MenuController.ControlHandle Struct Reference	45
3.16.1	Detailed Description	46
3.17	Fader Class Reference	46
3.18	AlCarDriver.GearInfo Class Reference	46
3.18.1	Detailed Description	47
3.18.2	Field Documentation	47
3.18.2.1	maxTorqueMultiplier	47
3.18.2.2	tillWhatRPM	47
3.19	AgentController.LimitAndInfluence Class Reference	47
3.19.1	Detailed Description	48
3.19.2	Field Documentation	48
3.19.2.1	avoidForce	48
3.19.2.2	detectionLength	48
3.19.2.3	maximumContainmentDistance	48
3.19.2.4	maximumDistanceForPursuit	48
3.19.2.5	maximumObstacleAvoidanceDistance	48
3.19.2.6	velocitySensorMultiplier	48
3.20	LimitedSpeedZone Class Reference	49
3.20.1	Detailed Description	49
3.20.2	Field Documentation	49

3.20.2.1	maxSpeed	49
3.21	MenuController Class Reference	49
3.21.1	Member Function Documentation	50
3.21.1.1	InitiateNewRaceConfig()	50
3.21.1.2	UpdateAICount()	51
3.21.1.3	UpdateInputOptions()	51
3.21.1.4	UpdateLapCount()	51
3.21.2	Field Documentation	51
3.21.2.1	AICount	51
3.21.2.2	AICountSlider	51
3.21.2.3	AIType	51
3.21.2.4	circuitSceneName	51
3.21.2.5	circuitToggle	52
3.21.2.6	controlHandles	52
3.21.2.7	includePolice	52
3.21.2.8	IsAMobileApp	52
3.21.2.9	LapCount	52
3.21.2.10	LapCountSlider	52
3.21.2.11	musicPrefab	52
3.21.2.12	PlayerType	52
3.21.2.13	sprintSceneName	53
3.22	MusicController Class Reference	53
3.23	ParticleKiller Class Reference	53
3.23.1	Detailed Description	53
3.24	AgentController.ReversingVariables Class Reference	53
3.24.1	Detailed Description	54
3.24.2	Field Documentation	54
3.24.2.1	canReverse	54
3.24.2.2	minVelocityForReverseTimeOut	54
3.24.2.3	reverseTimeOutIn	54

3.24.2.4 reversingSensorMultiplier	54
3.25 SendUIInput Class Reference	55
3.25.1 Member Function Documentation	55
3.25.1.1 RefreshControls()	55
3.25.1.2 SetInputY()	55
3.25.2 Field Documentation	56
3.25.2.1 steerWheel	56
3.25.2.2 UI_Steer_BTNS	56
3.25.2.3 UI_Tap_Acc	56
3.26 SimpleSceneController Class Reference	56
3.26.1 Member Function Documentation	57
3.26.1.1 Exit()	57
3.26.1.2 Link()	57
3.26.1.3 LoadThis()	57
3.27 SirenAnimator Class Reference	57
3.27.1 Field Documentation	58
3.27.1.1 maxBrightness	58
3.27.1.2 myFlares	58
3.27.1.3 myPoliceAgent	58
3.27.1.4 sirenFrequency	58
3.28 SkidmarkDestroyer Class Reference	58
3.28.1 Detailed Description	59
3.28.2 Field Documentation	59
3.28.2.1 destroyAfter	59
3.28.2.2 fadeIn	59
3.29 AgentHUDInfo.speedDisplay Struct Reference	59
3.29.1 Detailed Description	59
3.30 SteerWheelUI Class Reference	60
3.30.1 Field Documentation	60
3.30.1.1 maximumSteeringAngle	60

3.30.1.2	wheelAngle	60
3.30.1.3	wheelReleasedSpeed	61
3.31	AgentController.StrengthClass Class Reference	61
3.31.1	Detailed Description	61
3.31.2	Field Documentation	61
3.31.2.1	containmentAhead	61
3.31.2.2	pridictPath	61
3.31.2.3	pursuitAhead	62
3.31.2.4	queuingAhead	62
3.31.2.5	seperationDistance	62
3.31.2.6	unalignedAvoidanceAhead	62
3.32	TransformRotator Class Reference	62
3.33	AgentController.WeightsClass Class Reference	63
3.33.1	Detailed Description	63
3.33.2	Field Documentation	63
3.33.2.1	avoidObstacle	63
3.33.2.2	containment	63
3.33.2.3	pathFollowing	63
3.33.2.4	pursuit	64
3.33.2.5	queuing	64
3.33.2.6	seperation	64
3.33.2.7	unalignedCollisionAvoidence	64
3.34	AICarDriver.WheelInfo Class Reference	64
3.34.1	Detailed Description	65
3.34.2	Field Documentation	65
3.34.2.1	isAMotorWheel	65
3.34.2.2	isASteeringWheel	65
3.34.2.3	skidMarkWidth	65
3.34.2.4	skidRenderer	65
3.34.2.5	skidVertexHolder	65
3.34.2.6	wheelCol	65
3.34.2.7	wheelTrans	65

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

AgentRaceManager.AgentsInRaceInfos	29
MenuController.ControlHandle	45
AICarDriver.GearInfo	46
AgentController.LimitAndInfluence	47
MonoBehaviour	
AgentController	5
AgentHUDInfo	9
AgentPathCreator	11
AgentPolice	14
AgentProgressTab	16
AgentRaceFinisher	18
AgentRaceManager	19
AgentRaceStarter	25
AgentRaceStarterInitiator	28
AICarDriver	32
AntiRoll	41
BillboardController	42
CameraScript	43
CameraTargetChanger	44
Fader	46
LimitedSpeedZone	49
MenuController	49
MusicController	53
ParticleKiller	53
SendUIInput	55
SimpleSceneController	56
SirenAnimator	57
SkidmarkDestroyer	58
SteerWheelUI	60
TransformRotator	62
AgentController.ReversingVariables	53
AgentHUDInfo.speedDisplay	59
AgentController.StrengthClass	61
AgentController.WeightsClass	63
AICarDriver.WheelInfo	64

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

AgentController	
Calculates the best path for the agent to move on but dosen't apply any type of movement . . .	5
AgentHUDInfo	9
AgentPathCreator	
Creates path for the agents and controls their progress	11
AgentPolice	14
AgentProgressTab	16
AgentRaceFinisher	18
AgentRaceManager	19
AgentRaceStarter	25
AgentRaceStarterInitiator	28
AgentRaceManager.AgentsInRaceInfos	
Agent in race info	29
AICarDriver	
Controls the AI car	32
AntiRoll	41
BillboardController	42
CameraScript	
Simple smooth follow of the camera	43
CameraTargetChanger	44
MenuController.ControlHandle	
Toggle controls for input type selection	45
Fader	46
AICarDriver.GearInfo	
Holds gearinfo	46
AgentController.LimitAndInfluence	
This class holds the variables that either limit or influence the vehicle behaviour for every Behaviour	47
LimitedSpeedZone	
Specific area where you want to set the maximum speed of an agent	49
MenuController	49
MusicController	53
ParticleKiller	
Destroys the particle system after it stops playing	53
AgentController.ReversingVariables	
This class holds Reversing variables	53

SendUIInput	55
SimpleSceneController	56
SirenAnimator	57
SkidmarkDestroyer	
Destroys skidmarks based on some values	58
AgentHUDInfo.speedDisplay	
Hold speedOmeter Information	59
SteerWheelUI	60
AgentController.StrengthClass	
This class holds strength variables for every Behaviour	61
TransformRotator	62
AgentController.WeightsClass	
This describes which behaviour will have more or less priority than other	63
AICarDriver.WheelInfo	
Holds wheel Info	64

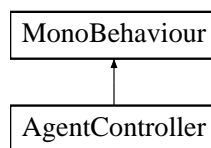
Chapter 3

Data Structure Documentation

3.1 AgentController Class Reference

Calculates the best path for the agent to move on but doesn't apply any type of movement

Inheritance diagram for AgentController:



Data Structures

- class [LimitAndInfluence](#)
This class holds the variables that either limit or influence the vehicle behaviour for every Behaviour
- class [ReversingVariables](#)
This class holds Reversing variables
- class [StrengthClass](#)
This class holds strength variables for every Behaviour
- class [WeightsClass](#)
This describes which behaviour will have more or less priority than other

Public Member Functions

- void [SetPolicing](#) (bool status)
Are we on a police job?
- bool [GetPolicing](#) ()
return true is we are policing
- bool [IsAPolice](#) ()
is This Agent a police
- Vector3 [GetVelocity](#) ()
Returns velocity based on transforms position

- Vector3 [SteerVector](#) ()
Returns current steer vector
- void [SetNewTarget](#) (Transform newTar)
Set the new target for the agent passed as newTar
- bool [isReversing](#) ()
Are we reversing?
- bool [SomethingDetected](#) ()
Is there something that is blocking the path? includes all the behaviours

Data Fields

- Transform [target](#)
Current Target
- [WeightsClass](#) [weights](#)
Object of weights class
- [StrengthClass](#) [strengths](#)
An object of strength class
- [LimitAndInfluence](#) [limitersAndInfluencers](#)
An object of [LimitAndInfluence](#) class
- [ReversingVariables](#) [reverseControl](#)
An object of [ReversingVariables](#) class
- Transform [] [sensors](#)
Avoidance and Pursuit Sensor made up of empty Transforms that can be scaled on forward axis to adjust the sensitivity
- LayerMask [whatIsaWall](#)
Define wall which the agent will use containment logic on
- LayerMask [useSensorsOn](#)
Uncheck The Floor
- bool [showDebugRays](#) = true
Show debug rays

3.1.1 Detailed Description

3.1.2 Member Function Documentation

3.1.2.1 GetPolicing()

```
bool AgentController.GetPolicing ( ) [inline]
```

Returns

3.1.2.2 GetVelocity()

```
Vector3 AgentController.GetVelocity ( ) [inline]
```

Returns

3.1.2.3 IsAPolice()

```
bool AgentController.IsAPolice ( ) [inline]
```

Returns

3.1.2.4 isReversing()

```
bool AgentController.isReversing ( ) [inline]
```

Returns

3.1.2.5 SetNewTarget()

```
void AgentController.SetNewTarget (
    Transform newTar ) [inline]
```

Parameters

<i>newTar</i>	
---------------	--

3.1.2.6 SetPolicing()

```
void AgentController.SetPolicing (
    bool status ) [inline]
```

Parameters

<i>status</i>	
---------------	--

3.1.2.7 SomethingDetected()

```
bool AgentController.SomethingDetected ( ) [inline]
```

Returns**3.1.2.8 SteerVector()**

```
Vector3 AgentController.SteerVector ( ) [inline]
```

Returns**3.1.3 Field Documentation****3.1.3.1 limitersAndInfluencers**

```
LimitAndInfluence AgentController.limitersAndInfluencers
```

3.1.3.2 reverseControl

```
ReversingVariables AgentController.reverseControl
```

3.1.3.3 sensors

```
Transform [ ] AgentController.sensors
```

3.1.3.4 showDebugRays

```
bool AgentController.showDebugRays = true
```

3.1.3.5 strengths

```
StrengthClass AgentController.strengths
```

3.1.3.6 target

```
Transform AgentController.target
```

3.1.3.7 useSensorsOn

```
LayerMask AgentController.useSensorsOn
```

3.1.3.8 weights

```
WeightsClass AgentController.weights
```

3.1.3.9 whatIsaWall

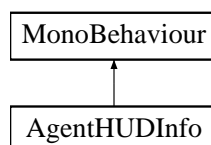
```
LayerMask AgentController.whatIsaWall
```

The documentation for this class was generated from the following file:

- AgentController.cs

3.2 AgentHUDInfo Class Reference

Inheritance diagram for AgentHUDInfo:



Data Structures

- struct `speedDisplay`
Hold speedOmeter Information

Public Member Functions

- void `DisableHUD` ()
Disable the hud
- void `EnableHUD` ()
Enable the hud
- void `GetDriver` ()

Data Fields

- int `agentID` = 0
Id of the agent whose HUD it is
- Text `lapDisplay`
Display the lap
- Text `positionDisplay`
Display the position
- `speedDisplay` `speedOMeter`
Object of speed display

3.2.1 Member Function Documentation

3.2.1.1 DisableHUD()

```
void AgentHUDInfo.DisableHUD ( ) [inline]
```

3.2.1.2 EnableHUD()

```
void AgentHUDInfo.EnableHUD ( ) [inline]
```

3.2.2 Field Documentation

3.2.2.1 agentID

```
int AgentHUDInfo.agentID = 0
```

3.2.2.2 lapDisplay

```
Text AgentHUDInfo.lapDisplay
```

3.2.2.3 positionDisplay

```
Text AgentHUDInfo.positionDisplay
```

3.2.2.4 speedOMeter

```
speedDisplay AgentHUDInfo.speedOMeter
```

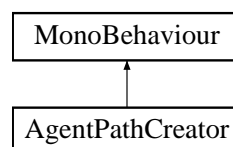
The documentation for this class was generated from the following file:

- AgentHUDInfo.cs

3.3 AgentPathCreator Class Reference

Creates path for the agents and controls their progress

Inheritance diagram for AgentPathCreator:



Public Member Functions

- void [StartTheRace](#) (int laps)
Sets everything and starts the race
- Vector3 [] [GetPath](#) ()
Returns a vector 3 array with path points
- void [AddNewAgent](#) ([AgentController](#) newAgent)
Add new agent for the race
- void [PlayerSpawned](#) (int at)
Did we spawn a human Controlled agent

Data Fields

- bool `LaunchAtStart` = true
Starts all the agents at the start of the scene
- float `radius` = 5.0f
Radius of the path
- bool `circuit` = false
is this path a circuit
- bool `oneWay` = false
make all the agents go one way
- int `oneWaySkipPoint` = 3
a constant value to fix oneWayLogic
- int `resetFixThreshold` = 5
Reset fix - play with this value if you vehicle loses control at the end of the path
- float `pathPredictionMultiplier` = 0.5f
velocity based prediction for every agent to make them stay on the path
- `AgentController [] agents`
Agents that you want to be controlled by this path
- `AgentController [] policeAgents`
These are the police agents who will be patrolling
- float `pathResolution` = 0.1f
Resolution for catmull-rom based smoothness of the path, make sure it adds up to 1

3.3.1 Detailed Description

3.3.2 Member Function Documentation

3.3.2.1 AddNewAgent()

```
void AgentPathCreator.AddNewAgent (
    AgentController newAgent ) [inline]
```

Parameters

<code>newAgent</code>	
-----------------------	--

3.3.2.2 GetPath()

```
Vector3 [] AgentPathCreator.GetPath ( ) [inline]
```

Returns

3.3.2.3 PlayerSpawned()

```
void AgentPathCreator.PlayerSpawned (
    int at ) [inline]
```

Parameters

<i>at</i>	
-----------	--

3.3.2.4 StartTheRace()

```
void AgentPathCreator.StartTheRace (
    int laps ) [inline]
```

3.3.3 Field Documentation

3.3.3.1 agents

```
AgentController [ ] AgentPathCreator.agents
```

3.3.3.2 circuit

```
bool AgentPathCreator.circuit = false
```

3.3.3.3 LaunchAtStart

```
bool AgentPathCreator.LaunchAtStart = true
```

3.3.3.4 oneWay

```
bool AgentPathCreator.oneWay = false
```

3.3.3.5 oneWaySkipPoint

```
int AgentPathCreator.oneWaySkipPoint = 3
```

3.3.3.6 pathPredictionMultiplier

```
float AgentPathCreator.pathPredictionMultiplier = 0.5f
```

3.3.3.7 pathResolution

```
float AgentPathCreator.pathResolution = 0.1f
```

3.3.3.8 policeAgents

```
AgentController [ ] AgentPathCreator.policeAgents
```

3.3.3.9 radius

```
float AgentPathCreator.radius = 5.0f
```

3.3.3.10 resetFixThreshold

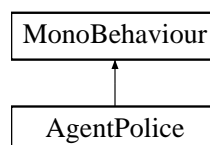
```
int AgentPathCreator.resetFixThreshold = 5
```

The documentation for this class was generated from the following file:

- AgentPathCreator.cs

3.4 AgentPolice Class Reference

Inheritance diagram for AgentPolice:



Public Member Functions

- bool [AreWePolicing](#) ()
are we policing

Data Fields

- bool [startPatroling](#)
How close should the speeder be for policing
- float [getTriggeredDistace](#) = 30.0f
How close should the speeder be for policing
- float [evadedDistance](#) = 100.0f
How far the speeder should be to be considered evaded
- LayerMask [hindrences](#)
Hindrence Layer
- float [patrolSpeed](#) = 70.0f
Speed when Patrolling
- AudioSource [audioPlayer](#)
SFX for sirens

3.4.1 Member Function Documentation

3.4.1.1 AreWePolicing()

```
bool AgentPolice.AreWePolicing ( ) [inline]
```

3.4.2 Field Documentation

3.4.2.1 audioPlayer

```
AudioSource AgentPolice.audioPlayer
```

3.4.2.2 evadedDistance

```
float AgentPolice.evadedDistance = 100.0f
```

3.4.2.3 getTriggeredDistace

```
float AgentPolice.getTriggeredDistace = 30.0f
```

3.4.2.4 hindrences

```
LayerMask AgentPolice.hindrences
```

3.4.2.5 patrolSpeed

```
float AgentPolice.patrolSpeed = 70.0f
```

3.4.2.6 startPatroling

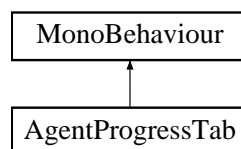
```
bool AgentPolice.startPatroling
```

The documentation for this class was generated from the following file:

- AgentPolice.cs

3.5 AgentProgressTab Class Reference

Inheritance diagram for AgentProgressTab:



Public Member Functions

- void [GiveName](#) (string newName)
Give this agent a name UI
- void [UpdateProgress](#) (int posInHierarchy, bool isFinished)
Update the progress

Data Fields

- Text [myPos](#)
Whats this agents position in race UI
- Text [myName](#)
Name of the agent UI

3.5.1 Member Function Documentation

3.5.1.1 GiveName()

```
void AgentProgressTab.GiveName (
    string newName ) [inline]
```

Parameters

<i>newName</i>	
----------------	--

3.5.1.2 UpdateProgress()

```
void AgentProgressTab.UpdateProgress (
    int posInHierarchy,
    bool isFinished ) [inline]
```

Parameters

<i>posInHierarchy</i>	whats the pos
<i>isFinished</i>	has this agent finished

3.5.2 Field Documentation

3.5.2.1 myName

```
Text AgentProgressTab.myName
```

3.5.2.2 myPos

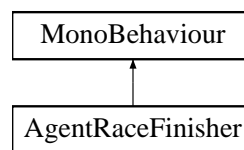
Text AgentProgressTab.myPos

The documentation for this class was generated from the following file:

- AgentProgressTab.cs

3.6 AgentRaceFinisher Class Reference

Inheritance diagram for AgentRaceFinisher:



Public Member Functions

- void [OnTriggerEnter](#) (Collider other)
trigger to finish the race

Data Fields

- [AgentRaceManager myRaceManager](#)
The race manager to finish the race

3.6.1 Member Function Documentation

3.6.1.1 OnTriggerEnter()

```
void AgentRaceFinisher.OnTriggerEnter (
    Collider other ) [inline]
```

Parameters

<i>other</i>	
--------------	--

3.6.2 Field Documentation

3.6.2.1 myRaceManager

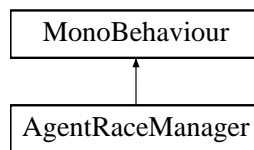
[AgentRaceManager](#) `AgentRaceFinisher.myRaceManager`

The documentation for this class was generated from the following file:

- `AgentRaceFinisher.cs`

3.7 AgentRaceManager Class Reference

Inheritance diagram for AgentRaceManager:



Data Structures

- class [AgentsInRaceInfos](#)
Agent in race info

Public Types

- enum [TriggerPlacementOption](#) { **Ahead**, **Behind** }
where is finish trigger is placed

Public Member Functions

- void [InitiateRaceManager](#) ([AgentController](#)[] racingAgents, Vector3[] path, bool humanRacing, int humanAt, int noOfLaps, bool isACircuit)
Initiates the race manager
- void [UpdateAgentsCurrentLapPosition](#) (int agentIndex, int pathPoint, Vector3 currentPos)
Update the progress the an agent in the race
- void [SortTheRacingAgents](#) ()
Reorder the racing agents based on their progress
- void [AgentFinishedTheLap](#) ([AgentController](#) thisAgent)
This agent finished the lap
- float [GetTheLapNumber](#) (int id)
Get the lap no.
- int [GetPositionInRaceHierarchy](#) (int id)
At what position
- [AICarDriver](#) [GetTheDriver](#) (int id)
Give me the driver
- int [HowManyLaps](#) ()
How many laps in this race
- int [HowManyRacing](#) ()
How many agents are racing
- bool [IsRaceInitiated](#) ()
is race manager initiated

Data Fields

- [TriggerPlacementOption](#) *whereIsTriggerIsPlaced*
Object
- Collider [finishTrigger](#)
Finishing trigger
- float [reverseFinishLineCheatingThreshold](#) = 0.2f
Reverse cheating threshold fixes the winning of race by just revesing back to finish line
- Transform [currentMainCamera](#)
Whats the main camera // will be assigned to billboard heads
- Transform [progressContent](#)
Content of leaderboard scroll
- GameObject [progressTab](#)
Prefab of progress tab for leaderboard
- GameObject [progressTabForHuman](#)
Prefab of progress tab for leaderboard and this one is for human
- GameObject [wholeProgressBoard](#)
Enable disable this progress board
- bool **autoHideProgressBoard** = true
- GameObject [endPanel](#)
End panel to show when you finish the race
- GameObject [headPosInfo](#)
Prefab for head info display
- String [] [agentNames](#)
Names to be randomly assigned
- GameObject [mobileControlsHolder](#)
Add the object that hold the UI based controls

3.7.1 Member Enumeration Documentation

3.7.1.1 TriggerPlacementOption

```
enum AgentRaceManager.TriggerPlacementOption [strong]
```

3.7.2 Member Function Documentation

3.7.2.1 AgentFinishedTheLap()

```
void AgentRaceManager.AgentFinishedTheLap (
    AgentController thisAgent ) [inline]
```

Parameters

<i>thisAgent</i>	pass the agent
------------------	----------------

3.7.2.2 GetPositionInRaceHierarchy()

```
int AgentRaceManager.GetPositionInRaceHierarchy (  
    int id ) [inline]
```

Parameters

<i>id</i>	this agent is at?
-----------	-------------------

Returns

3.7.2.3 GetTheDriver()

```
AICarDriver AgentRaceManager.GetTheDriver (  
    int id ) [inline]
```

Parameters

<i>id</i>	of this agent
-----------	---------------

Returns

3.7.2.4 GetTheLapNumber()

```
float AgentRaceManager.GetTheLapNumber (  
    int id ) [inline]
```

Parameters

<i>id</i>	of this agent
-----------	---------------

Returns

3.7.2.5 HowManyLaps()

```
int AgentRaceManager.HowManyLaps ( ) [inline]
```

Returns

3.7.2.6 HowManyRacing()

```
int AgentRaceManager.HowManyRacing ( ) [inline]
```

Returns

3.7.2.7 InitiateRaceManager()

```
void AgentRaceManager.InitiateRaceManager (
    AgentController [] racingAgents,
    Vector3 [] path,
    bool humanRacing,
    int humanAt,
    int noOfLaps,
    bool isACircuit ) [inline]
```

Parameters

<i>racingAgents</i>	who are all the agents that are racing
<i>path</i>	path array
<i>humanRacing</i>	is a human racing
<i>humanAt</i>	index of human agent
<i>noOfLaps</i>	how many laps , if any?
<i>isACircuit</i>	is this a circuit

3.7.2.8 IsRaceInitiated()

```
bool AgentRaceManager.IsRaceInitiated ( ) [inline]
```

Returns

3.7.2.9 SortTheRacingAgents()

```
void AgentRaceManager.SortTheRacingAgents ( ) [inline]
```

3.7.2.10 UpdateAgentsCurrentLapPosition()

```
void AgentRaceManager.UpdateAgentsCurrentLapPosition (
    int agentIndex,
    int pathPoint,
    Vector3 currentPos ) [inline]
```

Parameters

<i>agentIndex</i>	at what index this agent is at?
<i>pathPoint</i>	nearest path point
<i>currentPos</i>	agent's position

3.7.3 Field Documentation

3.7.3.1 agentNames

```
String [ ] AgentRaceManager.agentNames
```

3.7.3.2 currentMainCamera

```
Transform AgentRaceManager.currentMainCamera
```

3.7.3.3 endPanel

`GameObject AgentRaceManager.endPanel`

3.7.3.4 finishTrigger

`Collider AgentRaceManager.finishTrigger`

3.7.3.5 headPosInfo

`GameObject AgentRaceManager.headPosInfo`

3.7.3.6 mobileControlsHolder

`GameObject AgentRaceManager.mobileControlsHolder`

3.7.3.7 progressContent

`Transform AgentRaceManager.progressContent`

3.7.3.8 progressTab

`GameObject AgentRaceManager.progressTab`

3.7.3.9 progressTabForHuman

`GameObject AgentRaceManager.progressTabForHuman`

3.7.3.10 reverseFinishLineCheatingThreshold

`float AgentRaceManager.reverseFinishLineCheatingThreshold = 0.2f`

3.7.3.11 whereIsTriggerIsPlaced

`TriggerPlacementOption AgentRaceManager.whereIsTriggerIsPlaced`

3.7.3.12 wholeProgressBoard

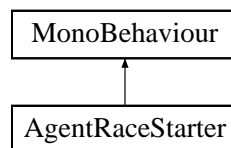
`GameObject AgentRaceManager.wholeProgressBoard`

The documentation for this class was generated from the following file:

- AgentRaceManager.cs

3.8 AgentRaceStarter Class Reference

Inheritance diagram for AgentRaceStarter:



Public Member Functions

- void `InitiateTheRaceStarter` (int reqAIType, int noOfAIs, int reqPlayerType, int noOfLaps, bool hasPolice, bool isMobile)
Initiate the race , spawn players and starts the countdown

Data Fields

- bool `initiateAtStart` = false
Initiate the race with default values
- Image [] `startCountElements`
Start 321 count elements
- AudioSource `audioPlayer`
Audio player
- AudioClip `countSound`
Count SFX
- float `delay` = 1.0f
Delay in count
- GameObject [] `carTypes`
Types of cars
- GameObject [] `spawnPoints`
Spawn points for the cars
- GameObject `playerSpawnPoint`

- Spawn point for player*
 - [CameraScript cameraToBeAssigned](#)
 - Assign this camera to player*
 - float [spawnAbove](#) = 5.0f
 - Spawn height above spawn point*
 - GameObject [policeHolder](#)
 - holds police cars*
 - [SendUIInput uIInput](#)
 - UI input Objecy*

3.8.1 Member Function Documentation

3.8.1.1 InitiateTheRaceStarter()

```
void AgentRaceStarter.InitiateTheRaceStarter (
    int reqAIType,
    int noOfAIs,
    int reqPlayerType,
    int noOfLaps,
    bool hasPolice,
    bool isMobile ) [inline]
```

Parameters

<i>reqAIType</i>	type of AI
<i>noOfAIs</i>	No. of AI's
<i>reqPlayerType</i>	player car type
<i>noOfLaps</i>	no. of laps

3.8.2 Field Documentation

3.8.2.1 audioPlayer

```
AudioSource AgentRaceStarter.audioPlayer
```

3.8.2.2 cameraToBeAssigned

```
CameraScript AgentRaceStarter.cameraToBeAssigned
```


3.8.2.3 carTypes

```
GameObject [] AgentRaceStarter.carTypes
```

3.8.2.4 countSound

```
AudioClip AgentRaceStarter.countSound
```

3.8.2.5 delay

```
float AgentRaceStarter.delay = 1.0f
```

3.8.2.6 initiateAtStart

```
bool AgentRaceStarter.initiateAtStart = false
```

3.8.2.7 playerSpawnPoint

```
GameObject AgentRaceStarter.playerSpawnPoint
```

3.8.2.8 policeHolder

```
GameObject AgentRaceStarter.policeHolder
```

3.8.2.9 spawnAbove

```
float AgentRaceStarter.spawnAbove = 5.0f
```

3.8.2.10 spawnPoints

```
GameObject [] AgentRaceStarter.spawnPoints
```

3.8.2.11 startCountElements

```
Image [ ] AgentRaceStarter.startCountElements
```

3.8.2.12 ullInput

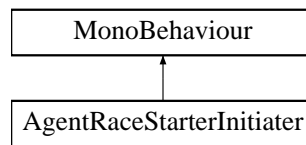
```
SendUIInput AgentRaceStarter.uIInput
```

The documentation for this class was generated from the following file:

- AgentRaceStarter.cs

3.9 AgentRaceStarterInitiator Class Reference

Inheritance diagram for AgentRaceStarterInitiator:



Public Member Functions

- void [AssignVars](#) (int AIType, int PlayerType, int Lap, int AICount, bool includePolice, bool areWeOnAMobile)
Assing the variables for new race config

3.9.1 Member Function Documentation

3.9.1.1 AssignVars()

```
void AgentRaceStarterInitiator.AssignVars (
    int AIType,
    int PlayerType,
    int Lap,
    int AICount,
    bool includePolice,
    bool areWeOnAMobile ) [inline]
```

Parameters

<i>AIType</i>	AI type
<i>PlayerType</i>	Player type
<i>Lap</i>	laps
<i>AICount</i>	AI count
<i>includePolice</i>	map will contain police agents

The documentation for this class was generated from the following file:

- AgentRaceStarterInitiater.cs

3.10 AgentRaceManager.AgentsInRaceInfos Class Reference

Agent in race info

Public Member Functions

- void [FinishTheLap](#) (int maxLaps, ref float stillRacing, bool itsACircuit)
Finish the lap and the race
- void [NewLapProgress](#) (float newProgress, float threshold)
Current lap progress
- void [AssignAgentController](#) ([AgentController](#) thisAgent)
The agent this block belongs to

Data Fields

- string [agentsName](#) = ""
Agent's name
- int [agentID](#) = 0
Agent's ID
- float [agentsCurrentLapProgress](#) = 0.0f
Current laps progress
- float [agentsLapNumber](#) = 1f
Current Lap no.
- float [longestLegalTravel](#) = 0.0f
Max lap progress covered legally
- bool [finished](#) = false
Has this agent finished the race?
- bool [disqualified](#) = false
Has this agent been disqualified
- bool [canFinishTheLap](#) = false
Can this agent finish the current lap?
- [AgentController](#) [myAgentController](#)
Agent controller of this agent
- [AgentProgressTab](#) [myProgressTab](#)
progress tab for leaderbordard
- Text [headPosInfo](#)
head pos info to be displayed at the head if the agent
- bool [skippedFirstLap](#) = false
have we skipped the first lap // depends on where the finish trigger is placed

3.10.1 Detailed Description

3.10.2 Member Function Documentation

3.10.2.1 AssignAgentController()

```
void AgentRaceManager.AgentsInRaceInfos.AssignAgentController (  
    AgentController thisAgent ) [inline]
```

Parameters

<i>thisAgent</i>	AI
------------------	----

3.10.2.2 FinishTheLap()

```
void AgentRaceManager.AgentsInRaceInfos.FinishTheLap (  
    int maxLaps,  
    ref float stillRacing,  
    bool itsACircuit ) [inline]
```

Parameters

<i>maxLaps</i>	no of laps
<i>stillRacing</i>	how many are still racing
<i>itsACircuit</i>	is this map a circuit

3.10.2.3 NewLapProgress()

```
void AgentRaceManager.AgentsInRaceInfos.NewLapProgress (  
    float newProgress,  
    float threshold ) [inline]
```

Parameters

<i>newProgress</i>	new calculated progress
<i>threshold</i>	reverse cheating threshold

3.10.3 Field Documentation

3.10.3.1 agentID

```
int AgentRaceManager.AgentsInRaceInfos.agentID = 0
```

3.10.3.2 agentsCurrentLapProgress

```
float AgentRaceManager.AgentsInRaceInfos.agentsCurrentLapProgress = 0.0f
```

3.10.3.3 agentsLapNumber

```
float AgentRaceManager.AgentsInRaceInfos.agentsLapNumber = 1f
```

3.10.3.4 agentsName

```
string AgentRaceManager.AgentsInRaceInfos.agentsName = ""
```

3.10.3.5 canFinishTheLap

```
bool AgentRaceManager.AgentsInRaceInfos.canFinishTheLap = false
```

3.10.3.6 disqualified

```
bool AgentRaceManager.AgentsInRaceInfos.disqualified = false
```

3.10.3.7 finished

```
bool AgentRaceManager.AgentsInRaceInfos.finished = false
```

3.10.3.8 headPosInfo

```
Text AgentRaceManager.AgentsInRaceInfos.headPosInfo
```

3.10.3.9 longestLegalTravel

```
float AgentRaceManager.AgentsInRaceInfos.longestLegalTravel = 0.0f
```

3.10.3.10 myAgentController

```
AgentController AgentRaceManager.AgentsInRaceInfos.myAgentController
```

3.10.3.11 myProgressTab

```
AgentProgressTab AgentRaceManager.AgentsInRaceInfos.myProgressTab
```

3.10.3.12 skippedFirstLap

```
bool AgentRaceManager.AgentsInRaceInfos.skippedFirstLap = false
```

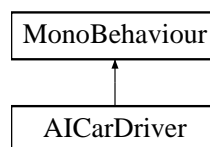
The documentation for this class was generated from the following file:

- AgentRaceManager.cs

3.11 AICarDriver Class Reference

Controls the AI car

Inheritance diagram for AICarDriver:



Data Structures

- class [GearInfo](#)
Holds gearinfo
- class [WheelInfo](#)
Holds wheel Info

Public Types

- enum **ControlTypes** { **Keyboard**, **UI_Tap_and_Acc_Control**, **UI_SteerWheel_BTNS** }

Public Member Functions

- void **SetNewControls** ()
Call When On mobile
- float **CurrentCalculatedSteer** ()
Gives current calculated steer angle
- void **SetInputY** (float to)
Set Y input
- void **SetInputX** (float to)
Set X input
- bool **GetGround** ()
Are we grounded?
- void **SetEngine** (bool recVal)
Set engine On or Off
- float **GetCurrentSpeed** ()
Returns car's current speed
- void **ApplyLocalPositionToVisuals** (WheelCollider col, Transform visualTransform)
position the transform based on wheel colliders calculated position
- bool **IsReversing** ()
Are we reversing?
- float **GetClampedVelocity** ()
Get the speed value clamped between 0 and 1

Data Fields

- List< **WheelInfo** > **wheels**
All the wheels of this vehical
- bool **controlledByPlayer** = false
Gives controls of this car to player
- ControlTypes **controlType**
Control types to be chosen
- WheelCollider [] **wheelsForSpeedCalculation**
Wheels to calculate the speed of car
- Vector3 **centerOfMass** = Vector3.zero
Center of mass at start
- float **centerOfMassAdditiveY** = -0.99f
lowered center of mass when at maimum speed
- float **maxTorque** = 200
Maximum torque
- float **handBrakeTorque** = 1000.0f
Maximum brake torque
- float **brakeSensitivity** = 0.75f
Brake sensitivity
- float **maxSteerAngle** = 30.0f
Maximum steer angle
- bool **speedRelativeSteer** = false

- Restrict steer based on speed*
 - float [steerStabilityThreshold](#) = 0.15f
 - Set a value between 0 and 1 it will try to stop vehicle wobbling*
- bool [flipSteerOnReverse](#) = false
 - Flip steer on reverse*
- float [steerResponsiveness](#) = 20.0f
 - How responsive the steering will be*
- float [pressureOnClimbs](#) = 0.15f
 - Pressure applied to car downwards when climbing*
- [GearInfo](#) [] [gears](#)
 - All the gears*
- float [maxSpeed](#) = 120.0f
 - Maximum speed of the car in KMPH*
- float [maxReverseSpeed](#) = 50.0f
 - Max reverse speed*
- float [skidVertexDistance](#) = 0.15f
 - maximum distance between two skidmark points*
- int [maxSkidVertices](#) = 10
 - maximum skid vertices for one geomatry*
- float [slideAfterSlip](#) = 0.002f
 - emnable skid effects after this much slip*
- Material [skidMaterial](#)
 - Material applied to the skid mark*
- GameObject [skidSmoke](#)
 - Skidsmoke for the wheels*
- GameObject [spark](#)
 - Spark effect on collisions*
- AudioClip [skidSound](#)
 - Skidding sound*
- bool [canFlip](#) = true
 - Can the car reset itself when stuck*
- float [flipTimeOut](#) = 1.0f
 - Reset after this much time*
- Renderer [rearLights](#)
 - Rear brake lights*
- Color [brakeColor](#) = Color.red
 - Brake color*
- Color [idleColor](#) = Color.black
 - Idle color*
- Color [reverseColor](#) = Color.white
 - Reverse color*

3.11.1 Detailed Description

3.11.2 Member Function Documentation

3.11.2.1 ApplyLocalPositionToVisuals()

```
void AICarDriver.ApplyLocalPositionToVisuals (
    WheelCollider col,
    Transform visualTransform ) [inline]
```


Parameters

<i>col</i>	passed wheel collider
<i>visualTransform</i>	passed wheel geomatry

3.11.2.2 CurrentCalculatedSteer()

```
float AICarDriver.CurrentCalculatedSteer ( ) [inline]
```

Returns

3.11.2.3 GetClampedVelocity()

```
float AICarDriver.GetClampedVelocity ( ) [inline]
```

Returns

3.11.2.4 GetCurrentSpeed()

```
float AICarDriver.GetCurrentSpeed ( ) [inline]
```

Returns

3.11.2.5 GetGround()

```
bool AICarDriver.GetGround ( ) [inline]
```

Returns

3.11.2.6 IsReversing()

```
bool AICarDriver.IsReversing ( ) [inline]
```

Returns

3.11.2.7 SetEngine()

```
void AICarDriver.SetEngine (
    bool recVal ) [inline]
```

3.11.2.8 SetInputX()

```
void AICarDriver.SetInputX (
    float to ) [inline]
```

Parameters

<i>to</i>	
-----------	--

3.11.2.9 SetInputY()

```
void AICarDriver.SetInputY (
    float to ) [inline]
```

Parameters

<i>to</i>	
-----------	--

3.11.2.10 SetNewControls()

```
void AICarDriver.SetNewControls ( ) [inline]
```

3.11.3 Field Documentation

3.11.3.1 brakeColor

```
Color AICarDriver.brakeColor = Color.red
```

3.11.3.2 brakeSensitivity

```
float AICarDriver.brakeSensitivity = 0.75f
```

3.11.3.3 canFlip

```
bool AICarDriver.canFlip = true
```

3.11.3.4 centerOfMass

```
Vector3 AICarDriver.centerOfMass = Vector3.zero
```

3.11.3.5 centerOfMassAdditiveY

```
float AICarDriver.centerOfMassAdditiveY = -0.99f
```

3.11.3.6 controlledByPlayer

```
bool AICarDriver.controlledByPlayer = false
```

3.11.3.7 controlType

```
ControlTypes AICarDriver.controlType
```

3.11.3.8 flipSteerOnReverse

```
bool AICarDriver.flipSteerOnReverse = false
```

3.11.3.9 flipTimeOut

```
float AICarDriver.flipTimeOut = 1.0f
```

3.11.3.10 gears

```
GearInfo [] AICarDriver.gears
```

3.11.3.11 handBrakeTorque

```
float AICarDriver.handBrakeTorque = 1000.0f
```

3.11.3.12 idleColor

```
Color AICarDriver.idleColor = Color.black
```

3.11.3.13 maxReverseSpeed

```
float AICarDriver.maxReverseSpeed = 50.0f
```

3.11.3.14 maxSkidVerticies

```
int AICarDriver.maxSkidVerticies = 10
```

3.11.3.15 maxSpeed

```
float AICarDriver.maxSpeed = 120.0f
```

3.11.3.16 maxSteerAngle

```
float AICarDriver.maxSteerAngle = 30.0f
```

3.11.3.17 maxTorque

```
float AICarDriver.maxTorque = 200
```

3.11.3.18 pressureOnClimbs

```
float AICarDriver.pressureOnClimbs = 0.15f
```

3.11.3.19 rearLights

```
Renderer AICarDriver.rearLights
```

3.11.3.20 reverseColor

```
Color AICarDriver.reverseColor = Color.white
```

3.11.3.21 skidMaterial

```
Material AICarDriver.skidMaterial
```

3.11.3.22 skidSmoke

```
GameObject AICarDriver.skidSmoke
```

3.11.3.23 skidSound

```
AudioClip AICarDriver.skidSound
```

3.11.3.24 skidVertexDistance

```
float AICarDriver.skidVertexDistance = 0.15f
```

3.11.3.25 `slideAfterSlip`

```
float AICarDriver.slideAfterSlip = 0.002f
```

3.11.3.26 `spark`

```
GameObject AICarDriver.spark
```

3.11.3.27 `speedRelativeSteer`

```
bool AICarDriver.speedRelativeSteer = false
```

3.11.3.28 `steerResponsiveness`

```
float AICarDriver.steerResponsiveness = 20.0f
```

3.11.3.29 `steerStabilityThreshold`

```
float AICarDriver.steerStabilityThreshold = 0.15f
```

3.11.3.30 `wheels`

```
List<WheelInfo> AICarDriver.wheels
```

3.11.3.31 `wheelsForSpeedCalculation`

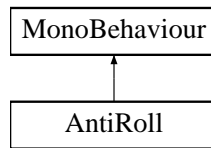
```
WheelCollider [] AICarDriver.wheelsForSpeedCalculation
```

The documentation for this class was generated from the following file:

- `AICarDriver.cs`

3.12 AntiRoll Class Reference

Inheritance diagram for AntiRoll:



Public Member Functions

- void [FixedUpdate](#) ()
Apply the antiroll

Data Fields

- WheelCollider [WheelL](#)
Left wheel on the axle
- WheelCollider [WheelR](#)
Right wheel on axle
- float [antiRoll](#) = 5000.0f
Anti roll force multiplier

3.12.1 Member Function Documentation

3.12.1.1 FixedUpdate()

```
void AntiRoll.FixedUpdate ( ) [inline]
```

3.12.2 Field Documentation

3.12.2.1 antiRoll

```
float AntiRoll.antiRoll = 5000.0f
```

3.12.2.2 WheelL

```
WheelCollider AntiRoll.WheelL
```

3.12.2.3 WheelR

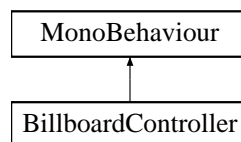
```
WheelCollider AntiRoll.WheelR
```

The documentation for this class was generated from the following file:

- AntiRoll.cs

3.13 BillboardController Class Reference

Inheritance diagram for BillboardController:



Public Member Functions

- void [GiveCam](#) (Transform suppCam)
Assign the camera

Data Fields

- float [minFadeDist](#) = 5.0f
Minimum fade distance
- float [maxfadeDist](#) = 10f
Maximum fade distance

3.13.1 Member Function Documentation

3.13.1.1 GiveCam()

```
void BillboardController.GiveCam (
    Transform suppCam ) [inline]
```


Parameters

<i>suppCam</i>	supplied the camera
----------------	---------------------

3.13.2 Field Documentation

3.13.2.1 maxfadeDist

```
float BillboardController.maxfadeDist = 10f
```

3.13.2.2 minFadeDist

```
float BillboardController.minFadeDist = 5.0f
```

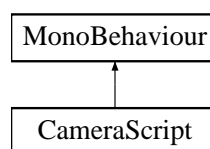
The documentation for this class was generated from the following file:

- BillboardController.cs

3.14 CameraScript Class Reference

Simple smooth follow of the camera

Inheritance diagram for CameraScript:



Data Fields

- Transform [target](#)
Current target of the camera
- float [distance](#) = 10.0f
Distance between target ad camera
- float [height](#) = 3.0f
Height difference between camera and target
- float [heightDamping](#) = 2.0f
Height damping
- float [rotationDamping](#) = 3.0f
Rotation Damping

3.14.1 Detailed Description

3.14.2 Field Documentation

3.14.2.1 distance

```
float CameraScript.distance = 10.0f
```

3.14.2.2 height

```
float CameraScript.height = 3.0f
```

3.14.2.3 heightDamping

```
float CameraScript.heightDamping = 2.0f
```

3.14.2.4 rotationDamping

```
float CameraScript.rotationDamping = 3.0f
```

3.14.2.5 target

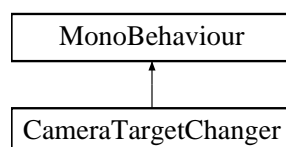
```
Transform CameraScript.target
```

The documentation for this class was generated from the following file:

- CameraScript.cs

3.15 CameraTargetChanger Class Reference

Inheritance diagram for CameraTargetChanger:



Public Member Functions

- void [PrevTarget](#) ()
Switch to previous target
- void [NextTarget](#) ()
Switch to next target

Data Fields

- Transform [] [targets](#)
Targets to be followed by the camera

3.15.1 Member Function Documentation

3.15.1.1 NextTarget()

```
void CameraTargetChanger.NextTarget ( ) [inline]
```

3.15.1.2 PrevTarget()

```
void CameraTargetChanger.PrevTarget ( ) [inline]
```

3.15.2 Field Documentation

3.15.2.1 targets

```
Transform [] CameraTargetChanger.targets
```

The documentation for this class was generated from the following file:

- CameraTargetChanger.cs

3.16 MenuController.ControlHandle Struct Reference

Toggle controls for input type selection

Data Fields

- Toggle **tapAndAcc**
- Toggle **steerAndButtons**

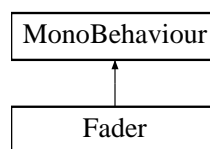
3.16.1 Detailed Description

The documentation for this struct was generated from the following file:

- MenuController.cs

3.17 Fader Class Reference

Inheritance diagram for Fader:



Public Member Functions

- void **InitiateFader** ()

Data Fields

- bool **start** = false
- float **fadeDamp** = 0.0f
- string **fadeScene**
- float **alpha** = 0.0f
- Color **fadeColor**
- bool **isFadeln** = false

The documentation for this class was generated from the following file:

- Fader.cs

3.18 AICarDriver.GearInfo Class Reference

Holds gearinfo

Data Fields

- float [tillWhatRPM](#)
till what rpm of the wheel we will use this gear
- float [maxTorqueMultiplier](#)
what the motor torque gonna be on this gear

3.18.1 Detailed Description

3.18.2 Field Documentation

3.18.2.1 maxTorqueMultiplier

```
float AICarDriver.GearInfo.maxTorqueMultiplier
```

3.18.2.2 tillWhatRPM

```
float AICarDriver.GearInfo.tillWhatRPM
```

The documentation for this class was generated from the following file:

- AICarDriver.cs

3.19 AgentController.LimitAndInfluence Class Reference

This class holds the variables that either limit or influence the vehicle behaviour for every Behaviour

Data Fields

- float [maximumDistanceForPursuit](#) = 30.0f
Minimum distance to start overtaking another agent
- float [maximumContainmentDistance](#) = 25.0f
Maximum Distance to enable the containment behaviour
- float [maximumObstacleAvoidanceDistance](#) = 20.0f
Maximum Distance to enable the avoidance behaviour
- float [detectionLength](#) = 5.0f
Minimum Detection length for every sensor (each of sensor can be scaled to minimize or maximize the minimum detection length)
- float [avoidForce](#) = 1.0f
how much influence behaviours have on the agent
- float [velocitySensorMultiplier](#) = 0.25f
How long ahead can the agent see based on velocity

3.19.1 Detailed Description

3.19.2 Field Documentation

3.19.2.1 avoidForce

```
float AgentController.LimitAndInfluence.avoidForce = 1.0f
```

3.19.2.2 detectionLength

```
float AgentController.LimitAndInfluence.detectionLength = 5.0f
```

3.19.2.3 maximumContainmentDistance

```
float AgentController.LimitAndInfluence.maximumContainmentDistance = 25.0f
```

3.19.2.4 maximumDistanceForPursuit

```
float AgentController.LimitAndInfluence.maximumDistanceForPursuit = 30.0f
```

3.19.2.5 maximumObstacleAvoidanceDistance

```
float AgentController.LimitAndInfluence.maximumObstacleAvoidanceDistance = 20.0f
```

3.19.2.6 velocitySensorMultiplier

```
float AgentController.LimitAndInfluence.velocitySensorMultiplier = 0.25f
```

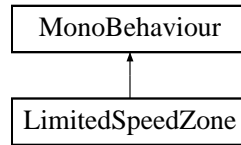
The documentation for this class was generated from the following file:

- AgentController.cs

3.20 LimitedSpeedZone Class Reference

Specific area where you want to set the maximum speed of an agent

Inheritance diagram for LimitedSpeedZone:



Data Fields

- float `maxSpeed` = 100
Maximum speed of the agent in this zone

3.20.1 Detailed Description

3.20.2 Field Documentation

3.20.2.1 maxSpeed

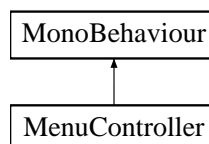
```
float LimitedSpeedZone.maxSpeed = 100
```

The documentation for this class was generated from the following file:

- LimitedSpeedZone.cs

3.21 MenuController Class Reference

Inheritance diagram for MenuController:



Data Structures

- struct `ControlHandle`
Toggle controls for input type selection

Public Member Functions

- void [UpdateInputOptions](#) ()
Updating the input
- void **SetInputType** (int newType)
- void [UpdateAICount](#) ()
Update the AI count
- void [UpdateLapCount](#) ()
Update the Lap count
- void [InitiateNewRaceConfig](#) ()
Initialize new AI race

Data Fields

- Text [AICount](#)
AI Count text display UI
- Slider [AICountSlider](#)
AI Count Slider AI
- Text [LapCount](#)
Lap Count display UI
- Slider [LapCountSlider](#)
Lap Count Slider
- Toggle [circuitToggle](#)
Is it a circuit UI
- Toggle [] [AIType](#)
Car type for AI
- Toggle [] [PlayerType](#)
Car type for Player
- Toggle [includePolice](#)
Car type for Player
- GameObject [musicPrefab](#)
Music prefab for background music
- string [circuitSceneName](#) = "CircuitRace"
scene to load when circuit is selected
- string [sprintSceneName](#) = "SprintRace"
scene to load when sprint is selected
- bool [IsAMobileApp](#) = false
Is this a mobile app
- [ControlHandle](#) [controlHandles](#)
Object for [ControlHandle](#)

3.21.1 Member Function Documentation

3.21.1.1 InitiateNewRaceConfig()

```
void MenuController.InitiateNewRaceConfig ( ) [inline]
```


3.21.1.2 UpdateAICount()

```
void MenuController.UpdateAICount ( ) [inline]
```

3.21.1.3 UpdateInputOptions()

```
void MenuController.UpdateInputOptions ( ) [inline]
```

3.21.1.4 UpdateLapCount()

```
void MenuController.UpdateLapCount ( ) [inline]
```

3.21.2 Field Documentation

3.21.2.1 AICount

```
Text MenuController.AICount
```

3.21.2.2 AICountSlider

```
Slider MenuController.AICountSlider
```

3.21.2.3 AIType

```
Toggle [ ] MenuController.AIType
```

3.21.2.4 circuitSceneName

```
string MenuController.circuitSceneName = "CircuitRace"
```

3.21.2.5 circuitToggle

Toggle MenuController.circuitToggle

3.21.2.6 controlHandles

ControlHandle MenuController.controlHandles

3.21.2.7 includePolice

Toggle MenuController.includePolice

3.21.2.8 IsAMobileApp

bool MenuController.IsAMobileApp = false

3.21.2.9 LapCount

Text MenuController.LapCount

3.21.2.10 LapCountSlider

Slider MenuController.LapCountSlider

3.21.2.11 musicPrefab

GameObject MenuController.musicPrefab

3.21.2.12 PlayerType

Toggle [] MenuController.PlayerType

3.21.2.13 sprintSceneName

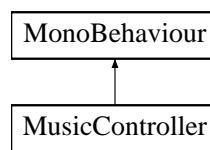
```
string MenuController.sprintSceneName = "SprintRace"
```

The documentation for this class was generated from the following file:

- MenuController.cs

3.22 MusicController Class Reference

Inheritance diagram for MusicController:



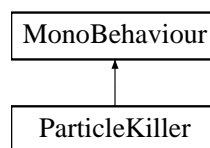
The documentation for this class was generated from the following file:

- MusicController.cs

3.23 ParticleKiller Class Reference

Destroys the particle system after it stops playing

Inheritance diagram for ParticleKiller:



3.23.1 Detailed Description

The documentation for this class was generated from the following file:

- ParticleKiller.cs

3.24 AgentController.ReversingVariables Class Reference

This class holds Reversing variables

Data Fields

- bool `canReverse` = false
can use reversing logic
- float `minVelocityForReverseTimeOut` = 0.25f
Minimum velocity ti check if we are really stuck
- float `reverseTimeOutCounter` = 0.0f
- float `reverseTimeOutIn` = .25f
Reverse in "amount of time"
- float `reversingSensorMultiplier` = .35f
Sensor detection while reversing
- bool `reversing` = false

3.24.1 Detailed Description

3.24.2 Field Documentation

3.24.2.1 `canReverse`

```
bool AgentController.ReversingVariables.canReverse = false
```

3.24.2.2 `minVelocityForReverseTimeOut`

```
float AgentController.ReversingVariables.minVelocityForReverseTimeOut = 0.25f
```

3.24.2.3 `reverseTimeOutIn`

```
float AgentController.ReversingVariables.reverseTimeOutIn = .25f
```

3.24.2.4 `reversingSensorMultiplier`

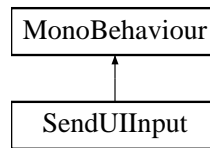
```
float AgentController.ReversingVariables.reversingSensorMultiplier = .35f
```

The documentation for this class was generated from the following file:

- `AgentController.cs`

3.25 SendUIInput Class Reference

Inheritance diagram for SendUIInput:



Public Member Functions

- void **GiveDriver** ([AICarDriver](#) driver)
- void [RefreshControls](#) ()
Get the new Input values
- void [SetInputY](#) (float newY)
Set new Y value

Data Fields

- [SteerWheelUI](#) [steerWheel](#)
Steering wheel for UI based input
- GameObject [UI_Tap_Acc](#)
Assign the holder of this type of controls
- GameObject [UI_Steer_BTNS](#)
Assign the holder of this type of controls

3.25.1 Member Function Documentation

3.25.1.1 RefreshControls()

```
void SendUIInput.RefreshControls ( ) [inline]
```

3.25.1.2 SetInputY()

```
void SendUIInput.SetInputY (
    float newY ) [inline]
```

Parameters

<i>newY</i>	pass this as new Y
-------------	--------------------

3.25.2 Field Documentation

3.25.2.1 steerWheel

`SteerWheelUI SendUIInput.steerWheel`

3.25.2.2 UI_Steer_BTNS

`GameObject SendUIInput.UI_Steer_BTNS`

3.25.2.3 UI_Tap_Acc

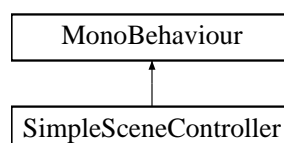
`GameObject SendUIInput.UI_Tap_Acc`

The documentation for this class was generated from the following file:

- SendUIInput.cs

3.26 SimpleSceneController Class Reference

Inheritance diagram for SimpleSceneController:



Public Member Functions

- void `LoadThis` (string scene)
Load tis scene
- void `Link` ()
open this link
- void `Exit` ()
Quit the application

3.26.1 Member Function Documentation

3.26.1.1 Exit()

```
void SimpleSceneController.Exit ( ) [inline]
```

3.26.1.2 Link()

```
void SimpleSceneController.Link ( ) [inline]
```

3.26.1.3 LoadThis()

```
void SimpleSceneController.LoadThis (
    string scene ) [inline]
```

Parameters

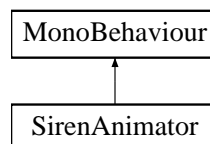
<i>scene</i>	
--------------	--

The documentation for this class was generated from the following file:

- SimpleSceneController.cs

3.27 SirenAnimator Class Reference

Inheritance diagram for SirenAnimator:



Data Fields

- [AgentPolice myPoliceAgent](#)
My police agent
- [LensFlare \[\] myFlares](#)
Siren Flares
- float [sirenFrequency](#) = 10.0f
Siren Frequency
- float [maxBrightness](#) = 20.0f
Max Siren brightness

3.27.1 Field Documentation

3.27.1.1 maxBrightness

```
float SirenAnimator.maxBrightness = 20.0f
```

3.27.1.2 myFlares

```
LensFlare [] SirenAnimator.myFlares
```

3.27.1.3 myPoliceAgent

```
AgentPolice SirenAnimator.myPoliceAgent
```

3.27.1.4 sirenFrequency

```
float SirenAnimator.sirenFrequency = 10.0f
```

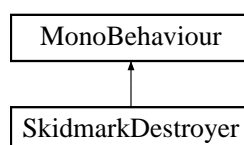
The documentation for this class was generated from the following file:

- SirenAnimator.cs

3.28 SkidmarkDestroyer Class Reference

Destroys skidmarks based on some values

Inheritance diagram for SkidmarkDestroyer:



Data Fields

- float `destroyAfter` = 5.0f
Start fading the skidmark in
- float `fadeIn` = 5.0f
Fade out in this many seconds

3.28.1 Detailed Description

3.28.2 Field Documentation

3.28.2.1 `destroyAfter`

```
float SkidmarkDestroyer.destroyAfter = 5.0f
```

3.28.2.2 `fadeIn`

```
float SkidmarkDestroyer.fadeIn = 5.0f
```

The documentation for this class was generated from the following file:

- SkidmarkDestroyer.cs

3.29 AgentHUDInfo.speedDisplay Struct Reference

Hold speedOmeter Information

Data Fields

- Image `needle`
- Text `digitalDisplay`
- float `minAngle`
- float `maxAngle`
- `AICarDriver` `myDriver`

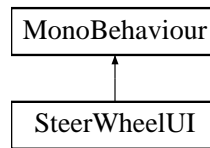
3.29.1 Detailed Description

The documentation for this struct was generated from the following file:

- AgentHUDInfo.cs

3.30 SteerWheelUI Class Reference

Inheritance diagram for SteerWheelUI:



Public Member Functions

- float **GetClampedValue** ()
- float **GetAngle** ()
- void **PressEvent** (BaseEventData eventData)
- void **DragEvent** (BaseEventData eventData)
- void **ReleaseEvent** (BaseEventData eventData)

Data Fields

- Graphic **UI_Element**
- float **maximumSteeringAngle** = 270f
Maximum angle this can be rotated at?
- float **wheelReleasedSpeed** = 1000f
How fast the weel should snap back
- float **wheelAngle** = 0f
Current wheel angle

3.30.1 Field Documentation

3.30.1.1 maximumSteeringAngle

```
float SteerWheelUI.maximumSteeringAngle = 270f
```

3.30.1.2 wheelAngle

```
float SteerWheelUI.wheelAngle = 0f
```

3.30.1.3 wheelReleasedSpeed

```
float SteerWheelUI.wheelReleasedSpeed = 1000f
```

The documentation for this class was generated from the following file:

- SteerWheelUI.cs

3.31 AgentController.StrengthClass Class Reference

This class holds strength variables for every Behaviour

Data Fields

- float [pursuitAhead](#)
Pursuit look ahead
- float [queuingAhead](#)
Queuing if ahead
- float [seperationDistance](#)
Speration Distance
- float [unalignedAvoidanceAhead](#)
UnalignedAvoidance how much furter in future
- float [containmentAhead](#)
Avoidance for containment behaviours
- float [pridictPath](#) = 1.0f
Pridict path ahead

3.31.1 Detailed Description

3.31.2 Field Documentation

3.31.2.1 containmentAhead

```
float AgentController.StrengthClass.containmentAhead
```

3.31.2.2 pridictPath

```
float AgentController.StrengthClass.pridictPath = 1.0f
```

3.31.2.3 pursuitAhead

```
float AgentController.StrengthClass.pursuitAhead
```

3.31.2.4 queuingAhead

```
float AgentController.StrengthClass.queuingAhead
```

3.31.2.5 seperationDistance

```
float AgentController.StrengthClass.seperationDistance
```

3.31.2.6 unalignedAvoidanceAhead

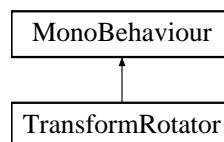
```
float AgentController.StrengthClass.unalignedAvoidanceAhead
```

The documentation for this class was generated from the following file:

- AgentController.cs

3.32 TransformRotator Class Reference

Inheritance diagram for TransformRotator:



Data Fields

- float **rotateSpeed** = 0.2f

The documentation for this class was generated from the following file:

- TransformRotator.cs

3.33 AgentController.WeightsClass Class Reference

This describes which behaviour will have more or less priority than other

Data Fields

- float [pursuit](#)
Defines priority and power of persuit behaviour
- float [avoidObstacle](#)
Defines priority and power of obstacle avoidance behaviour
- float [containment](#)
Defines priority and power of wall containment behaviour
- float [pathFollowing](#)
Defines priority and power of path following behaviour
- float [seperation](#)
Defines priority and power of Sepration behaviour
- float [unalignedCollisionAvoidence](#)
Defines priority and power of Unaligned Collision Avoidence behaviour
- float [queuing](#)
Defines priority and power of Queuing behaviour

3.33.1 Detailed Description

3.33.2 Field Documentation

3.33.2.1 avoidObstacle

```
float AgentController.WeightsClass.avoidObstacle
```

3.33.2.2 containment

```
float AgentController.WeightsClass.containment
```

3.33.2.3 pathFollowing

```
float AgentController.WeightsClass.pathFollowing
```

3.33.2.4 pursuit

```
float AgentController.WeightsClass.pursuit
```

3.33.2.5 queuing

```
float AgentController.WeightsClass.queuing
```

3.33.2.6 seperation

```
float AgentController.WeightsClass.seperation
```

3.33.2.7 unalignedCollisionAvoidance

```
float AgentController.WeightsClass.unalignedCollisionAvoidance
```

The documentation for this class was generated from the following file:

- AgentController.cs

3.34 AICarDriver.WheelInfo Class Reference

Holds wheel Info

Data Fields

- WheelCollider [wheelCol](#)
Wheel Collider of this wheel
- Transform [wheelTrans](#)
the wheel with geomatry
- bool [isAMotorWheel](#)
Is this a motor wheel
- bool [isASteeringWheel](#)
is this a steering wheel
- LineRenderer [skidRenderer](#)
Not visible in inspector holds current skid mark
- float [skidMarkWidth](#)
Width of the skid mark
- List< Vector3 > [skidVertexHolder](#)
Not visible in inspector holds current skid mark positions

3.34.1 Detailed Description

3.34.2 Field Documentation

3.34.2.1 isAMotorWheel

`bool AICarDriver.WheelInfo.isAMotorWheel`

3.34.2.2 isASteeringWheel

`bool AICarDriver.WheelInfo.isASteeringWheel`

3.34.2.3 skidMarkWidth

`float AICarDriver.WheelInfo.skidMarkWidth`

3.34.2.4 skidRenderer

`LineRenderer AICarDriver.WheelInfo.skidRenderer`

3.34.2.5 skidVertexHolder

`List<Vector3> AICarDriver.WheelInfo.skidVertexHolder`

3.34.2.6 wheelCol

`WheelCollider AICarDriver.WheelInfo.wheelCol`

3.34.2.7 wheelTrans

`Transform AICarDriver.WheelInfo.wheelTrans`

The documentation for this class was generated from the following file:

- AICarDriver.cs

Index

- AICarDriver, 32
 - ApplyLocalPositionToVisuals, 34
 - brakeColor, 36
 - brakeSensitivity, 37
 - canFlip, 37
 - centerOfMass, 37
 - centerOfMassAdditiveY, 37
 - controlType, 37
 - controlledByPlayer, 37
 - CurrentCalculatedSteer, 35
 - flipSteerOnReverse, 37
 - flipTimeOut, 37
 - gears, 38
 - GetClampedVelocity, 35
 - GetCurrentSpeed, 35
 - GetGround, 35
 - handBrakeTorque, 38
 - idleColor, 38
 - IsReversing, 35
 - maxReverseSpeed, 38
 - maxSkidVerticies, 38
 - maxSpeed, 38
 - maxSteerAngle, 38
 - maxTorque, 38
 - pressureOnClimbs, 39
 - rearLights, 39
 - reverseColor, 39
 - SetEngine, 36
 - SetInputX, 36
 - SetInputY, 36
 - SetNewControls, 36
 - skidMaterial, 39
 - skidSmoke, 39
 - skidSound, 39
 - skidVertexDistance, 39
 - slideAfterSlip, 39
 - spark, 40
 - speedRelativeSteer, 40
 - steerResponsiveness, 40
 - steerStabilityThreshold, 40
 - wheels, 40
 - wheelsForSpeedCalculation, 40
- AICarDriver.GearInfo, 46
- AICarDriver.WheelInfo, 64
- AICarDriver::GearInfo
 - maxTorqueMultiplier, 47
 - tillWhatRPM, 47
- AICarDriver::WheelInfo
 - isAMotorWheel, 65
 - isASteeringWheel, 65
 - skidMarkWidth, 65
 - skidRenderer, 65
 - skidVertexHolder, 65
 - wheelCol, 65
 - wheelTrans, 65
- AICount
 - MenuController, 51
- AICountSlider
 - MenuController, 51
- AIType
 - MenuController, 51
- AddNewAgent
 - AgentPathCreator, 12
- AgentController, 5
 - GetPolicing, 6
 - GetVelocity, 6
 - IsAPolice, 7
 - isReversing, 7
 - limitersAndInfluencers, 8
 - reverseControl, 8
 - sensors, 8
 - SetNewTarget, 7
 - SetPolicing, 7
 - showDebugRays, 8
 - SomethingDetected, 8
 - SteerVector, 8
 - strengths, 9
 - target, 9
 - useSensorsOn, 9
 - weights, 9
 - whatIsaWall, 9
- AgentController.LimitAndInfluence, 47
- AgentController.ReversingVariables, 53
- AgentController.StrengthClass, 61
- AgentController.WeightsClass, 63
- AgentController::LimitAndInfluence
 - avoidForce, 48
 - detectionLength, 48
 - maximumContainmentDistance, 48
 - maximumDistanceForPursuit, 48
 - maximumObstacleAvoidanceDistance, 48
 - velocitySensorMultiplier, 48
- AgentController::ReversingVariables
 - canReverse, 54
 - minVelocityForReverseTimeOut, 54
 - reverseTimeOutIn, 54
 - reversingSensorMultiplier, 54
- AgentController::StrengthClass

- containmentAhead, 61
- pridictPath, 61
- pursuitAhead, 61
- queuingAhead, 62
- seperationDistance, 62
- unalignedAvoidanceAhead, 62
- AgentController::WeightsClass
 - avoidObstacle, 63
 - containment, 63
 - pathFollowing, 63
 - pursuit, 63
 - queuing, 64
 - seperation, 64
 - unalignedCollisionAvoidence, 64
- AgentFinishedTheLap
 - AgentRaceManager, 20
- AgentHUDInfo, 9
 - agentID, 10
 - DisableHUD, 10
 - EnableHUD, 10
 - lapDisplay, 11
 - positionDisplay, 11
 - speedOMeter, 11
- AgentHUDInfo.speedDisplay, 59
- agentID
 - AgentHUDInfo, 10
 - AgentRaceManager::AgentsInRaceInfos, 30
- agentNames
 - AgentRaceManager, 23
- AgentPathCreator, 11
 - AddNewAgent, 12
 - agents, 13
 - circuit, 13
 - GetPath, 12
 - LaunchAtStart, 13
 - oneWay, 13
 - oneWaySkipPoint, 13
 - pathPridictionMultiplier, 14
 - pathResolution, 14
 - PlayerSpawned, 12
 - policeAgents, 14
 - radius, 14
 - resetFixThreshold, 14
 - StartTheRace, 13
- AgentPolice, 14
 - AreWePolicing, 15
 - audioPlayer, 15
 - evadedDistance, 15
 - getTriggeredDistace, 15
 - hindrences, 16
 - patrolSpeed, 16
 - startPatrolling, 16
- AgentProgressTab, 16
 - GiveName, 17
 - myName, 17
 - myPos, 17
 - UpdateProgress, 17
- AgentRaceFinisher, 18
 - myRaceManager, 18
 - OnTriggerEnter, 18
- AgentRaceManager, 19
 - AgentFinishedTheLap, 20
 - agentNames, 23
 - currentMainCamera, 23
 - endPanel, 23
 - finishTrigger, 24
 - GetPositionInRaceHierarchy, 21
 - GetTheDriver, 21
 - GetTheLapNumber, 21
 - headPosInfo, 24
 - HowManyLaps, 22
 - HowManyRacing, 22
 - InitiateRaceManager, 22
 - IsRaceInitiated, 22
 - mobileControlsHolder, 24
 - progressContent, 24
 - progressTab, 24
 - progressTabForHuman, 24
 - reverseFinishLineCheetingThreshold, 24
 - SortTheRacingAgents, 23
 - TriggerPlacementOption, 20
 - UpdateAgentsCurrentLapPosition, 23
 - whereIsTriggerIsPlaced, 24
 - wholeProgressBoard, 25
- AgentRaceManager.AgentsInRaceInfos, 29
- AgentRaceManager::AgentsInRaceInfos
 - agentID, 30
 - agentsCurrentLapProgress, 31
 - agentsLapNumber, 31
 - agentsName, 31
 - AssignAgentController, 30
 - canFinishTheLap, 31
 - disqualified, 31
 - FinishTheLap, 30
 - finished, 31
 - headPosInfo, 31
 - longestLegalTravel, 31
 - myAgentController, 32
 - myProgressTab, 32
 - NewLapProgress, 30
 - skippedFirstLap, 32
- AgentRaceStarter, 25
 - audioPlayer, 26
 - cameraToBeAssigned, 26
 - carTypes, 26
 - countSound, 27
 - delay, 27
 - initiateAtStart, 27
 - InitiateTheRaceStarter, 26
 - playerSpawnPoint, 27
 - policeHolder, 27
 - spawnAbove, 27
 - spawnPoints, 27
 - startCountElements, 27
 - ullInput, 28
- AgentRaceStarterInitiator, 28

- AssignVars, 28
- agents
 - AgentPathCreator, 13
- agentsCurrentLapProgress
 - AgentRaceManager::AgentsInRaceInfos, 31
- agentsLapNumber
 - AgentRaceManager::AgentsInRaceInfos, 31
- agentsName
 - AgentRaceManager::AgentsInRaceInfos, 31
- AntiRoll, 41
 - antiRoll, 41
 - FixedUpdate, 41
 - WheelL, 41
 - WheelR, 42
- antiRoll
 - AntiRoll, 41
- ApplyLocalPositionToVisuals
 - AICarDriver, 34
- AreWePolicing
 - AgentPolice, 15
- AssignAgentController
 - AgentRaceManager::AgentsInRaceInfos, 30
- AssignVars
 - AgentRaceStarterInitiator, 28
- audioPlayer
 - AgentPolice, 15
 - AgentRaceStarter, 26
- avoidForce
 - AgentController::LimitAndInfluence, 48
- avoidObstacle
 - AgentController::WeightsClass, 63
- BillboardController, 42
 - GiveCam, 42
 - maxfadeDist, 43
 - minFadeDist, 43
- brakeColor
 - AICarDriver, 36
- brakeSensitivity
 - AICarDriver, 37
- CameraScript, 43
 - distance, 44
 - height, 44
 - heightDamping, 44
 - rotationDamping, 44
 - target, 44
- CameraTargetChanger, 44
 - NextTarget, 45
 - PrevTarget, 45
 - targets, 45
- cameraToBeAssigned
 - AgentRaceStarter, 26
- canFinishTheLap
 - AgentRaceManager::AgentsInRaceInfos, 31
- canFlip
 - AICarDriver, 37
- canReverse
 - AgentController::ReversingVariables, 54
- carTypes
 - AgentRaceStarter, 26
- centerOfMass
 - AICarDriver, 37
- centerOfMassAdditiveY
 - AICarDriver, 37
- circuit
 - AgentPathCreator, 13
- circuitSceneName
 - MenuController, 51
- circuitToggle
 - MenuController, 51
- containment
 - AgentController::WeightsClass, 63
- containmentAhead
 - AgentController::StrengthClass, 61
- controlHandles
 - MenuController, 52
- controlType
 - AICarDriver, 37
- controlledByPlayer
 - AICarDriver, 37
- countSound
 - AgentRaceStarter, 27
- CurrentCalculatedSteer
 - AICarDriver, 35
- currentMainCamera
 - AgentRaceManager, 23
- delay
 - AgentRaceStarter, 27
- destroyAfter
 - SkidmarkDestroyer, 59
- detectionLength
 - AgentController::LimitAndInfluence, 48
- DisableHUD
 - AgentHUDInfo, 10
- disqualified
 - AgentRaceManager::AgentsInRaceInfos, 31
- distance
 - CameraScript, 44
- EnableHUD
 - AgentHUDInfo, 10
- endPanel
 - AgentRaceManager, 23
- evadedDistance
 - AgentPolice, 15
- Exit
 - SimpleSceneController, 57
- fadeIn
 - SkidmarkDestroyer, 59
- Fader, 46
- FinishTheLap
 - AgentRaceManager::AgentsInRaceInfos, 30
- finishTrigger
 - AgentRaceManager, 24
- finished

- AgentRaceManager::AgentsInRaceInfos, 31
- FixedUpdate
 - AntiRoll, 41
- flipSteerOnReverse
 - AICarDriver, 37
- flipTimeOut
 - AICarDriver, 37
- gears
 - AICarDriver, 38
- GetClampedVelocity
 - AICarDriver, 35
- GetCurrentSpeed
 - AICarDriver, 35
- GetGround
 - AICarDriver, 35
- GetPath
 - AgentPathCreator, 12
- GetPolicing
 - AgentController, 6
- GetPositionInRaceHierarchy
 - AgentRaceManager, 21
- GetTheDriver
 - AgentRaceManager, 21
- GetTheLapNumber
 - AgentRaceManager, 21
- getTriggeredDistace
 - AgentPolice, 15
- GetVelocity
 - AgentController, 6
- GiveCam
 - BillboardController, 42
- GiveName
 - AgentProgressTab, 17
- handBrakeTorque
 - AICarDriver, 38
- headPosInfo
 - AgentRaceManager, 24
 - AgentRaceManager::AgentsInRaceInfos, 31
- height
 - CameraScript, 44
- heightDamping
 - CameraScript, 44
- hindrences
 - AgentPolice, 16
- HowManyLaps
 - AgentRaceManager, 22
- HowManyRacing
 - AgentRaceManager, 22
- idleColor
 - AICarDriver, 38
- includePolice
 - MenuController, 52
- initiateAtStart
 - AgentRaceStarter, 27
- InitiateNewRaceConfig
 - MenuController, 50
- InitiateRaceManager
 - AgentRaceManager, 22
- InitiateTheRaceStarter
 - AgentRaceStarter, 26
- IsAMobileApp
 - MenuController, 52
- isAMotorWheel
 - AICarDriver::WheelInfo, 65
- IsAPolice
 - AgentController, 7
- isASteeringWheel
 - AICarDriver::WheelInfo, 65
- IsRaceInitiated
 - AgentRaceManager, 22
- IsReversing
 - AICarDriver, 35
- isReversing
 - AgentController, 7
- LapCount
 - MenuController, 52
- LapCountSlider
 - MenuController, 52
- lapDisplay
 - AgentHUDInfo, 11
- LaunchAtStart
 - AgentPathCreator, 13
- LimitedSpeedZone, 49
 - maxSpeed, 49
- limitersAndInfluencers
 - AgentController, 8
- Link
 - SimpleSceneController, 57
- LoadThis
 - SimpleSceneController, 57
- longestLegalTravel
 - AgentRaceManager::AgentsInRaceInfos, 31
- maxBrightness
 - SirenAnimator, 58
- maxReverseSpeed
 - AICarDriver, 38
- maxSkidVerticies
 - AICarDriver, 38
- maxSpeed
 - AICarDriver, 38
 - LimitedSpeedZone, 49
- maxSteerAngle
 - AICarDriver, 38
- maxTorque
 - AICarDriver, 38
- maxTorqueMultiplier
 - AICarDriver::GearInfo, 47
- maxfadeDist
 - BillboardController, 43
- maximumContainmentDistance
 - AgentController::LimitAndInfluence, 48
- maximumDistanceForPursuit
 - AgentController::LimitAndInfluence, 48

- maximumObstacleAvoidanceDistance
 - AgentController::LimitAndInfluence, [48](#)
- maximumSteeringAngle
 - SteerWheelUI, [60](#)
- MenuController, [49](#)
 - AICount, [51](#)
 - AICountSlider, [51](#)
 - AIType, [51](#)
 - circuitSceneName, [51](#)
 - circuitToggle, [51](#)
 - controlHandles, [52](#)
 - includePolice, [52](#)
 - InitiateNewRaceConfig, [50](#)
 - IsAMobileApp, [52](#)
 - LapCount, [52](#)
 - LapCountSlider, [52](#)
 - musicPrefab, [52](#)
 - PlayerType, [52](#)
 - sprintSceneName, [52](#)
 - UpdateAICount, [50](#)
 - UpdateInputOptions, [51](#)
 - UpdateLapCount, [51](#)
- MenuController.ControlHandle, [45](#)
- minFadeDist
 - BillboardController, [43](#)
- minVelocityForReverseTimeOut
 - AgentController::ReversingVariables, [54](#)
- mobileControlsHolder
 - AgentRaceManager, [24](#)
- MusicController, [53](#)
- musicPrefab
 - MenuController, [52](#)
- myAgentController
 - AgentRaceManager::AgentsInRaceInfos, [32](#)
- myFlares
 - SirenAnimator, [58](#)
- myName
 - AgentProgressTab, [17](#)
- myPoliceAgent
 - SirenAnimator, [58](#)
- myPos
 - AgentProgressTab, [17](#)
- myProgressTab
 - AgentRaceManager::AgentsInRaceInfos, [32](#)
- myRaceManager
 - AgentRaceFinisher, [18](#)
- NewLapProgress
 - AgentRaceManager::AgentsInRaceInfos, [30](#)
- NextTarget
 - CameraTargetChanger, [45](#)
- OnTriggerEnter
 - AgentRaceFinisher, [18](#)
- oneWay
 - AgentPathCreator, [13](#)
- oneWaySkipPoint
 - AgentPathCreator, [13](#)
- ParticleKiller, [53](#)
- pathFollowing
 - AgentController::WeightsClass, [63](#)
- pathPredictionMultiplier
 - AgentPathCreator, [14](#)
- pathResolution
 - AgentPathCreator, [14](#)
- patrolSpeed
 - AgentPolice, [16](#)
- playerSpawnPoint
 - AgentRaceStarter, [27](#)
- PlayerSpawned
 - AgentPathCreator, [12](#)
- PlayerType
 - MenuController, [52](#)
- policeAgents
 - AgentPathCreator, [14](#)
- policeHolder
 - AgentRaceStarter, [27](#)
- positionDisplay
 - AgentHUDInfo, [11](#)
- pressureOnClimbs
 - AICarDriver, [39](#)
- PrevTarget
 - CameraTargetChanger, [45](#)
- pridictPath
 - AgentController::StrengthClass, [61](#)
- progressContent
 - AgentRaceManager, [24](#)
- progressTab
 - AgentRaceManager, [24](#)
- progressTabForHuman
 - AgentRaceManager, [24](#)
- pursuit
 - AgentController::WeightsClass, [63](#)
- pursuitAhead
 - AgentController::StrengthClass, [61](#)
- queuing
 - AgentController::WeightsClass, [64](#)
- queuingAhead
 - AgentController::StrengthClass, [62](#)
- radius
 - AgentPathCreator, [14](#)
- rearLights
 - AICarDriver, [39](#)
- RefreshControls
 - SendUIInput, [55](#)
- resetFixThreshold
 - AgentPathCreator, [14](#)
- reverseColor
 - AICarDriver, [39](#)
- reverseControl
 - AgentController, [8](#)
- reverseFinishLineCheetingThreshold
 - AgentRaceManager, [24](#)
- reverseTimeOutIn
 - AgentController::ReversingVariables, [54](#)

- reversingSensorMultiplier
 - AgentController::ReversingVariables, [54](#)
- rotationDamping
 - CameraScript, [44](#)
- SendUIInput, [55](#)
 - RefreshControls, [55](#)
 - SetInputY, [55](#)
 - steerWheel, [56](#)
 - UI_Steer_BTNS, [56](#)
 - UI_Tap_Acc, [56](#)
- sensors
 - AgentController, [8](#)
- seperation
 - AgentController::WeightsClass, [64](#)
- seperationDistance
 - AgentController::StrengthClass, [62](#)
- SetEngine
 - AICarDriver, [36](#)
- SetInputX
 - AICarDriver, [36](#)
- SetInputY
 - AICarDriver, [36](#)
 - SendUIInput, [55](#)
- SetNewControls
 - AICarDriver, [36](#)
- SetNewTarget
 - AgentController, [7](#)
- SetPolicing
 - AgentController, [7](#)
- showDebugRays
 - AgentController, [8](#)
- SimpleSceneController, [56](#)
 - Exit, [57](#)
 - Link, [57](#)
 - LoadThis, [57](#)
- SirenAnimator, [57](#)
 - maxBrightness, [58](#)
 - myFlares, [58](#)
 - myPoliceAgent, [58](#)
 - sirenFrequency, [58](#)
- sirenFrequency
 - SirenAnimator, [58](#)
- skidMarkWidth
 - AICarDriver::WheelInfo, [65](#)
- skidMaterial
 - AICarDriver, [39](#)
- skidRenderer
 - AICarDriver::WheelInfo, [65](#)
- skidSmoke
 - AICarDriver, [39](#)
- skidSound
 - AICarDriver, [39](#)
- skidVertexDistance
 - AICarDriver, [39](#)
- skidVertexHolder
 - AICarDriver::WheelInfo, [65](#)
- SkidmarkDestroyer, [58](#)
 - destroyAfter, [59](#)
- fadeOut, [59](#)
- skippedFirstLap
 - AgentRaceManager::AgentsInRaceInfos, [32](#)
- slideAfterSlip
 - AICarDriver, [39](#)
- SomethingDetected
 - AgentController, [8](#)
- SortTheRacingAgents
 - AgentRaceManager, [23](#)
- spark
 - AICarDriver, [40](#)
- spawnAbove
 - AgentRaceStarter, [27](#)
- spawnPoints
 - AgentRaceStarter, [27](#)
- speedOMeter
 - AgentHUDInfo, [11](#)
- speedRelativeSteer
 - AICarDriver, [40](#)
- sprintSceneName
 - MenuController, [52](#)
- startCountElements
 - AgentRaceStarter, [27](#)
- startPatrolling
 - AgentPolice, [16](#)
- StartTheRace
 - AgentPathCreator, [13](#)
- steerResponsiveness
 - AICarDriver, [40](#)
- steerStabilityThreshold
 - AICarDriver, [40](#)
- SteerVector
 - AgentController, [8](#)
- steerWheel
 - SendUIInput, [56](#)
- SteerWheelUI, [60](#)
 - maximumSteeringAngle, [60](#)
 - wheelAngle, [60](#)
 - wheelReleasedSpeed, [60](#)
- strengths
 - AgentController, [9](#)
- target
 - AgentController, [9](#)
 - CameraScript, [44](#)
- targets
 - CameraTargetChanger, [45](#)
- tillWhatRPM
 - AICarDriver::GearInfo, [47](#)
- TransformRotator, [62](#)
- TriggerPlacementOption
 - AgentRaceManager, [20](#)
- UI_Steer_BTNS
 - SendUIInput, [56](#)
- UI_Tap_Acc
 - SendUIInput, [56](#)
- uInput
 - AgentRaceStarter, [28](#)

- unalignedAvoidanceAhead
 - AgentController::StrengthClass, [62](#)
- unalignedCollisionAvoidance
 - AgentController::WeightsClass, [64](#)
- UpdateAICount
 - MenuController, [50](#)
- UpdateAgentsCurrentLapPosition
 - AgentRaceManager, [23](#)
- UpdateInputOptions
 - MenuController, [51](#)
- UpdateLapCount
 - MenuController, [51](#)
- UpdateProgress
 - AgentProgressTab, [17](#)
- useSensorsOn
 - AgentController, [9](#)
- velocitySensorMultiplier
 - AgentController::LimitAndInfluence, [48](#)
- weights
 - AgentController, [9](#)
- whatIsaWall
 - AgentController, [9](#)
- wheelAngle
 - SteerWheelUI, [60](#)
- wheelCol
 - AICarDriver::WheelInfo, [65](#)
- wheelReleasedSpeed
 - SteerWheelUI, [60](#)
- wheelTrans
 - AICarDriver::WheelInfo, [65](#)
- WheelL
 - AntiRoll, [41](#)
- WheelR
 - AntiRoll, [42](#)
- wheels
 - AICarDriver, [40](#)
- wheelsForSpeedCalculation
 - AICarDriver, [40](#)
- whereIsTriggerIsPlaced
 - AgentRaceManager, [24](#)
- wholeProgressBoard
 - AgentRaceManager, [25](#)