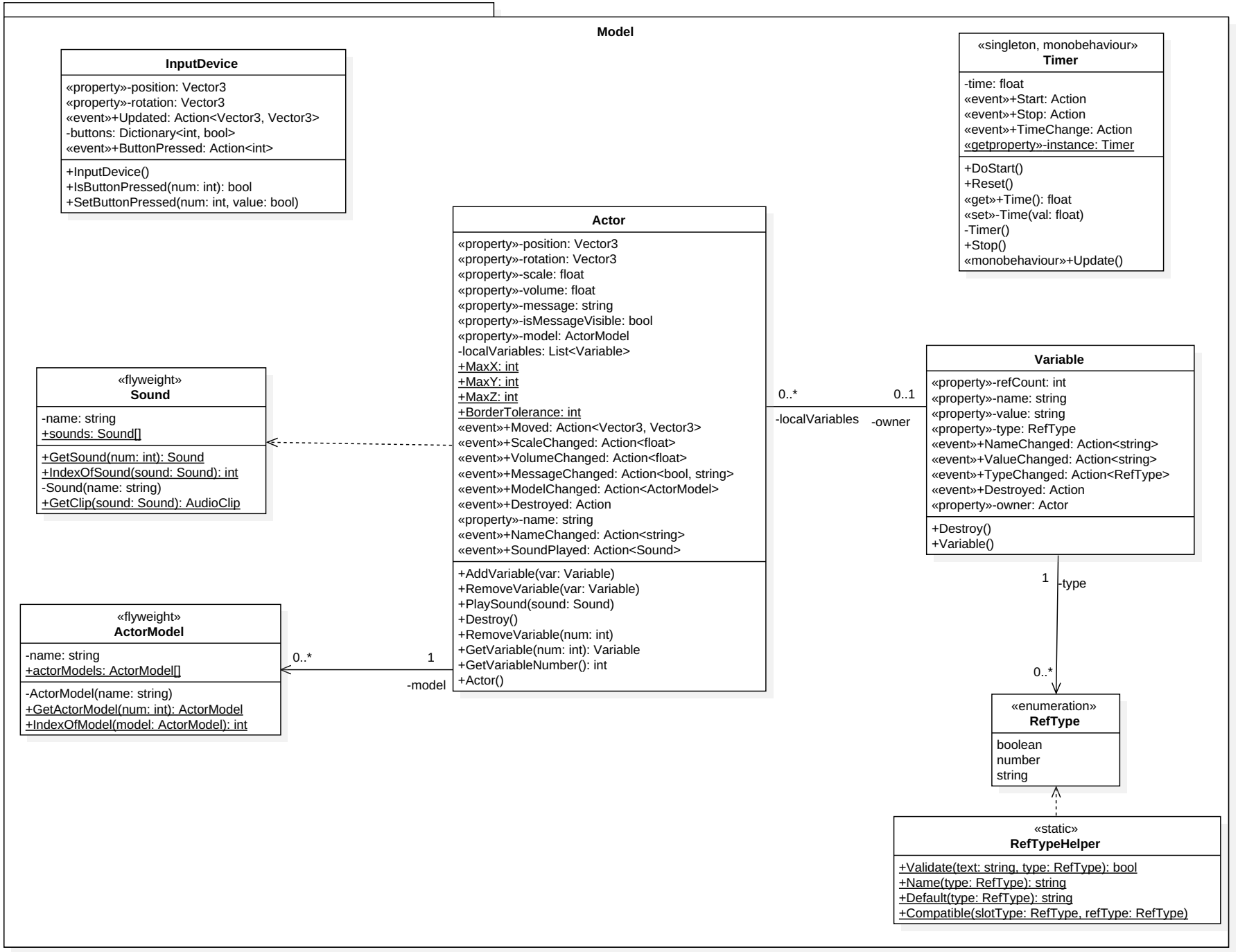
Model
InputDevice
«property»-position: Vector3
«property»-rotation: Vector3
«event»+Updated: Action<Vector3, Vector3>
-buttons: Dictionary<int, bool>
«event»+ButtonPressed: Action<int>
+InputDevice()
+IsButtonPressed(num: int): bool
+SetButtonPressed(num: int, value: bool)
«singleton, monobehaviour»
Timer
-time: float
«event»+Start: Action
«event»+Stop: Action
«event»+TimeChange: Action
«getproperty»-instance: Timer
+DoStart()
+Reset()
«get»+Time(): float
«set»-Time(val: float)
-Timer()
+Stop()
«monobehaviour»+Update()
Actor
«property»-position: Vector3
«property»-rotation: Vector3
«property»-scale: float
«property»-volume: float
«property»-message: string
«property»-isMessageVisible: bool
«property»-model: ActorModel
-localVariables: List<Variable>
+MaxX: int
+MaxY: int
+MaxZ: int
+BorderTolerance: int
«event»+Moved: Action<Vector3, Vector3>
«event»+ScaleChanged: Action<float>
«event»+VolumeChanged: Action<float>
«event»+MessageChanged: Action<bool, string>
«event»+ModelChanged: Action<ActorModel>
«event»+Destroyed: Action
«property»-name: string
«event»+NameChanged: Action<string>
«event»+SoundPlayed: Action<Sound>
+AddVariable(var: Variable)
+RemoveVariable(var: Variable)
+PlaySound(sound: Sound)
+Destroy()
+RemoveVariable(num: int)
+GetVariable(num: int): Variable
+GetVariableNumber(): int
+Actor()
«flyweight»
Sound
-name: string
+sounds: Sound[]
+GetSound(num: int): Sound
+IndexOfSound(sound: Sound): int
-Sound(name: string)
+GetClip(sound: Sound): AudioClip
0..*
-localVariables
0..1
-owner
Variable
«property»-refCount: int
«property»-name: string
«property»-value: string
«property»-type: RefType
«event»+NameChanged: Action<string>
«event»+ValueChanged: Action<string>
«event»+TypeChanged: Action<RefType>
«event»+Destroyed: Action
«property»-owner: Actor
+Destroy()
+Variable()
1
-type
0..*
«flyweight»
ActorModel
-name: string
+actorModels: ActorModel[]
-ActorModel(name: string)
+GetActorModel(num: int): ActorModel
+IndexOfModel(model: ActorModel): int
0..*
1
-model
«enumeration»
RefType
boolean
number
string
«static»
RefTypeHelper
+Validate(text: string, type: RefType): bool
+Name(type: RefType): string
+Default(type: RefType): string
+Compatible(slotType: RefType, refType: RefType)