

# Appendix C

## Classes

This chapter contains brief descriptions of all classes in the Sheep framework. The purpose of this appendix is to provide an overview of how the packages are structured, and as such, method and attribute details for each class has been omitted.

### C.1 `sheep.audio`

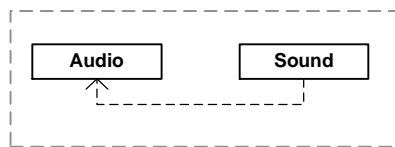


Figure C.1: Classes in the `sheep.audio` package.

This package contains the audio device singleton and a class for loading and playing sound.

- *Audio* – Represents the audio device on the system. This basically represents the instance of the Android SDK `SoundPool` class.
- *Sound* – Represents a playable sound.

### C.2 `sheep.collision`

This package contains classes for collision detection, spatial partitioning and interfaces for collision events.

- *Spatial* – Interface for all spatial partitioning structures.
- *SpatialVisitor* – In order to create objects which can visit sprites contained in spatial partitioners, their corresponding classes much implement this interface.
- *Flat* – Implements a "flat" partitioning as a reference.
- *QuadTree* – A QuadTree is a way of dividing the game scene into quadrants recursively.
- *Quad* – The node in the QuadTree.

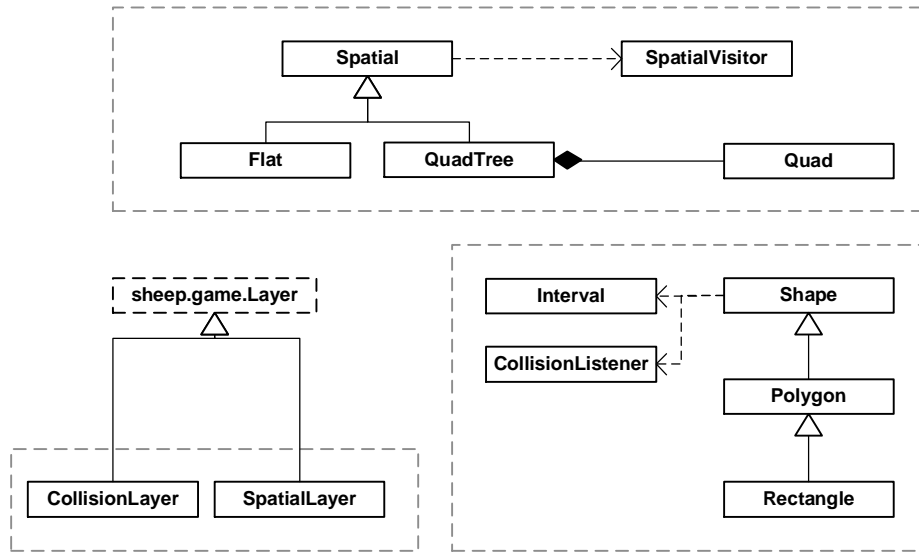


Figure C.2: Classes in the `sheep collision` package.

- *CollisionLayer* – A layer which contains Sprites and generates collisions between them, but without the overhead a spatial partitioning. Flat Implements a "flat" partitionin
- *SpatialLayer* – A layer which can partition sprites according to a Spatial implementor.
- *Shape* – An abstraction over all shapes in the collision detection system.
- *Interval* – Contains an interval used in the collision detection algortihm.
- *CollisionListener* – Classes that want to be able to listen for collision events on a certain sprite must implement this interface.
- *Polygon* – An implementation of the Shape class for general convex polygons.
- *Rectangle* – A convenience class for creating rectangular Polygons.

### C.3 `sheep.game`

This package contains contains classes related to game logic, that is classes which help form the model of the game.

- *GameThread* –
- *Game* – This class is subclass of `SurfaceView`, which makes it "viewable" in an Android application.
- *State* – You should inherit from this class to create the various states for your game.
- *SpriteContainer* – An interface for classes which contains Sprites.
- *Sprite* – The Sprite represents superclass of the "models" of the game objects.
- *Camera* – Can be attached to a World object and translate the entire scene according its position.
- *World* – A World is a simply a collection of Layers and a Camera.
- *Layer* – Abstract class for all Layers.

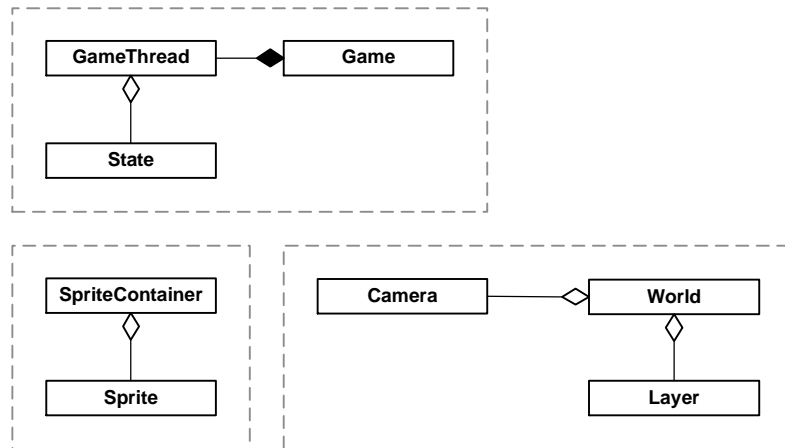


Figure C.3: Classes in the `sheep.game` package.

## C.4 `sheep.graphics`

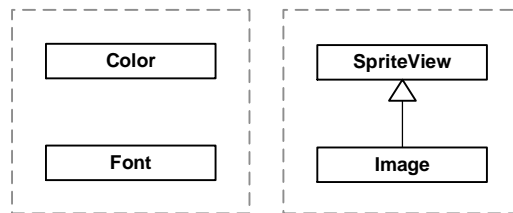


Figure C.4: Classes in the `sheep.graphics` package.

This package contains contains classes related to visual representations, including the "view".

- *Color* – This class exists to simplify the creation of predefined pure-color Paint.
- *Font* – This class exists primarily as a convenience when drawing text.
- *SpriteView* – This class is an abstraction over various visual representations for Sprites.
- *Image* – A Sprite representation which draws a transformed image onto the Canvas.

## C.5 `sheep.gui`

This package contains the graphical user interface system.

- *Widget* – The superclass of all components in the GUI system.
- *Container* – A Container is a Widget which contains other Widgets.
- *TextButton* – A TextButton is a simple borderless button which trigger WidgetEvents when clicked.
- *WidgetListener* – Observers of widgets must implement this interface.
- *WidgetAction* – Superclass for WidgetActions.

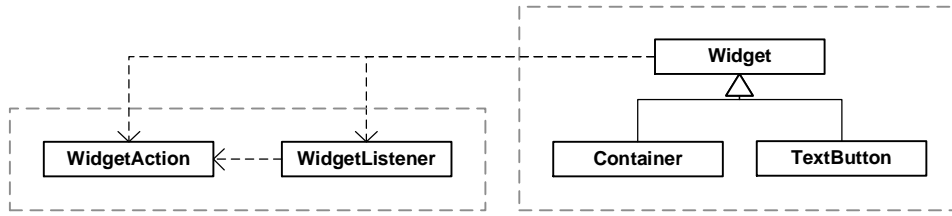


Figure C.5: Classes in the `sheep.gui` package.

## C.6 `sheep.input`

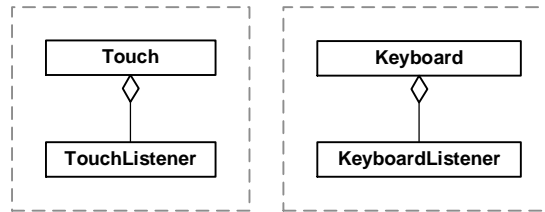


Figure C.6: Classes in the `sheep.input` package.

This package contains the input singletons and interfaces for listening to events generated by these singletons.

- *Keyboard* – This singleton class tracks the state of the keyboard.
- *KeyboardListener* – An interface for classes which want to subscribe for events on the keyboard.
- *Touch* – This singleton class tracks the state of the touchscreen.
- *TouchListener* – Objects can subscribe to touch events by implementing this interface.

## C.7 `sheep.math`

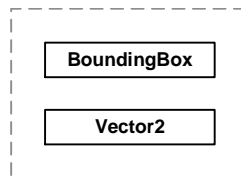


Figure C.7: Classes in the `sheep.math` package.

This package contains math components not directly related to collision detection.

- *BoundingBox* – A bounding box defined by two pairs min/max limits along the x and y axes.
- *Vector2* – A 2D Vector with basic operations like add, sub, length, normalize, and so forth.

## C.8 `sheep.util`

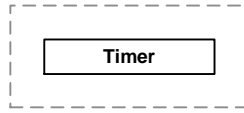


Figure C.8: Classes in the `sheep.util` package.

This package contains only the `Timer` device.

- *Timer* – A class which keeps track of time from one call of `getDelta` to the next.