Aerodynamic Objects

Anyone can make things fly

Read Me

Last updated 14/07/2023 14:27:00

Welcome to Aerodynamic Objects – a tool to help you make things fly in Unity

What is it?

Aerodynamic Objects is a set of tools for real-time modelling of aerodynamic behaviour of physical objects.

Features include:

- Aerodynamics engine for calculating aerodynamic forces acting on objects
- Easy integration with Unity's physics engine
- Custom flow field models to create wind or wakes
- Visualisation tools to provide visual feedback for developers and players
- Expanding range of design tools to support development of complex assemblies such as aircraft, sail boats or flying animals

Aerodynamic Objects uses real physics with simplifying assumptions to make it both fast and easy to use. It will give an answer that is exactly right sometimes, but more importantly is guaranteed to give an answer that is plausible at all times. This means that whatever you ask the model to do, it will always give a usable answer.

Integration with the Unity physics engine is simple; the AeroObject component can be connected to a Rigidbody component to apply aerodynamic forces to that rigid body. This means you can extend existing models and game rigs in Unity to include aerodynamic behaviours as well as build new models from scratch.

Our design philosophy with Aerodynamic Objects is that users should be able to create interesting and complex models and behaviours driven by aerodynamics without having to have in-depth knowledge of the subject. Simple things should be easy to do immediately; complex things should be possible with experience. If you are used to working in traditional animation, there is a step up to physics-based animation (simulation). Physics-based animation is a hugely powerful creative tool, but you need to develop understanding to be able to direct it as you wish.

Online Resources

An Introduction to Aerodynamic Objects video tutorial series is available to help you learn the basics. It assumes the user has some background understanding of high-school physics and a working knowledge of the Unity interface. The tutorials start with the basics of simulating wind and simple objects, then move on to building up the components of a simple but fully flyable propeller-driven aircraft. Scripting is not covered in the tutorials – where simple scripts are needed these are supplied readymade. Completed versions of the tutorials are also provided as Unity scenes as part of the distributed Unity package.

Getting started **video tutorials**:

https://youtube.com/playlist?list=PLohJe_Z4v94c6IrDXtbejYX3Fkm29Gjan

Further **examples** can be found on:

www.AerodynamicObjects.com

Full code documentation can be found at:

https://conorzam.github.io/AerodynamicObjectsDocs/

Getting Started

This section provides a quick start guide to using Aerodynamic Objects to apply aerodynamic forces to objects, visualise the forces being applied, and to set up fluid zones to define specific fluid velocities within the scene.

Aero Objects

The core functionality of Aerodynamic Objects comes from the AeroObject class. The AeroObject class calculates the chosen aerodynamic forces for an object based on the transform scale and the dimensions provided in the class.

The aerodynamic forces include:

- **Drag** – also known as air resistance, this force slows down an object moving through a fluid. It always acts in the opposite direction to the object's velocity relative to the fluid
- **Lift** – objects which have a low thickness relative to their length, such as wings, can produce lift. Lift acts perpendicularly to the object's relative velocity.
- **Buoyancy** – this force arises due to differences in density between an object and a fluid. If an object has a lower density than the fluid it is in, it will float.
- **Rotational Damping** – this is a torque which slows down an object with angular velocity.
- **Rotational Lift** – also known as the Magnus effect, this is a force which arises when a spinning object moves through a fluid. It is advised to exclude this force unless the application specifically requires it.

Aerodynamic forces can be calculated based on the motion of a Rigidbody component or by tracking the transform's motion. An AeroObject will default to using a Rigidbody to get its velocities. To change this, select "Is Kinematic" and the object will use the attached transform to obtain its velocity.

An AeroObject will always apply forces to a Rigidbody if it is provided a reference.

Arrows

Each aerodynamic force has an associated arrow for visualisation purposes. For example, the DragArrow component will render an arrow which points in the direction of the object's drag force and will have a length relative to the magnitude of the drag force. Other arrow components include wind velocity and the weight of the object.

To use the arrow components, attach them to the same GameObject as the AeroObject you want to visualise forces for.

Fluid Zones

Fluid zones are used to set the properties of a fluid within a specific region of a scene. This could be for a particularly windy area or to model the wake behind an object.

A fluid zone is created by adding the FluidZone component to an object and a trigger collider. To have an AeroObject interact with a FluidZone, the AeroObject must have a collider or a trigger collider either attached to the same GameObject or to a parent of the AeroObject. The AeroObject will then automatically detect when it enters a fluid zone and use the fluid zone’s properties when calculating its aerodynamic forces.