Ship Controller Documentation

`ShipController` is a `SceneComponent` that manages other (child) ship `SceneComponents` such as `ShipEngine` and `ShipRudder`.

Input

Ship Controller requires following binding to be set up under *Project Settings ⇒ Input*:

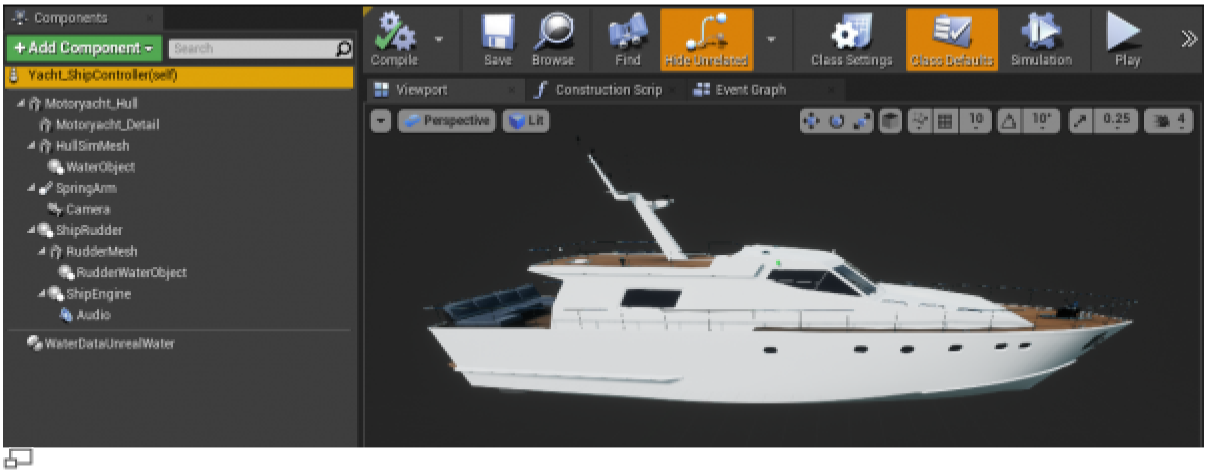
- *EngineStartStop* [action mapping]
- *Throttle* [axis]
- *Rudder* [axis]

Hierarchy

`ShipController` is intended to be used together with `WaterObject`, but this is not required.

Typical hierarchy would be:

- **MyShip** [`AShipController`] - ship controller pawn class.
  - **Hull** [`UStaticMeshComponent`] - a ship hull.
    - **HullWaterObject** [`UWaterObject`] - for simulating the ship hull/water interaction.
  - **ShipRudder** [`UShipRudder`] - hadles rotation of the rudder with input.
    - **ShipRudderWaterObject** [`UWaterObject`] - for simulating the rudder/water interaction.
  - **ShipEngine** [`UShipEngine`] - manages thrust, sound and propeller animation.
    - **Propeller** [`UStaticMeshComponent`][optional] - visual propeller rotation only, not used for simulation. It would be possible to use a `WaterObject` as a propeller, but not as flexible as a simple AddForce approach.
    - **Audio** [`UAudioComponent`][optional] - engine audio that gets modified through volume and pitch.
  - **Keel** [`UStaticMeshComponent`][optional]
    - **KeelWaterObject** [`UWaterObject`][optional]
  - … other `WaterObjects` …



Example ship setup from the demo yacht.

A few notes:

- `AShipController` should be placed as a root as it is a pawn class.
- Usage of `UShipRudder` and `UShipEngine` is optional.
- Multiple rudders and/or engines can be present at the same time.
- To create an outboard engine where the direction of thrust changes with the rudder direction, parent the `UShipEngine` to `UShipRudder` as the direction and position of the thrust of the `UShipEngine` is equal to its transform position and forward direction.

UShipController

A simple ship controller script containing no optional fields.

UShipRudder

Rotates the child objects around its transform position.

UShipEngine

Handles engine thrust, sound, and propeller rotation. The latter two are optional.

- Thrust will be applied to the root `UPrimitiveComponent`.
- To make the engine run on BeginPlay, tick `Is On` field.
- To use sound attach `UAudioComponent` as a child of `UShipEngine` and assign a looping audio file. Sound settings can be adjusted through the *Sound* section in the *Details* panel.
- To use propeller rotation (visual only) attach `UStaticMeshComponent` representing the propeller as a child of `UShipEngine`.
    - It is important that the mesh pivot of the propeller is at its center or otherwise the rotation will be offset.
    - Propeller rotation speed in relation to the engine can be adjusted through `Propeller RPM Ratio`.

From:
http://unreal.dynamicwaterphysics.com/ - **Documentation for Unreal Engine**

Permanent link:
http://unreal.dynamicwaterphysics.com/doku.php/ShipController

Last update: **2021/10/19 16:21**