# Catalyst Engine Documentation

## Defines

- **CATALYST_ENABLE_OCEAN** – Enables ocean rendering.
- **CATALYST_ENABLE_VOLUMETRIC_FOG** – Enables volumetric fog rendering.

## Naming

- Member variables start with an underscore. Every subsequent word begins with a capital letter, the rest lower case. Example: `_MyMemberVariable`
- All other variables is written in camel case, i.e. the first word starts with a lower case letter and subsequent letters are all lower case. The subsequent words starts with upper case. Example: m`yVariable`
- Constant variables are all in upper case, with underscores separating words. Example: `MY_CONSTANT_VARIABLE`

# Update Phases

The Catalyst Engine is updated in a number of distinct phases. Updates for game-specific logic can be injected during these phases via the **UpdateSystem**, both in synchronous and asynchronous passes. Below follows a summary of what happens during these phases on the engine side. The results of what is done during each of these phases can only be assumed to be available in subsequent phases.

## *Pre-update*

Can also be though of as the *"preparation phase"*. In the following order, during this phase;

- The update system is run, executing all registered synchronous and asynchronous pre-update functions.
- The culling system is run, updating the world space axis-aligned bounding boxes of relevant entities.
- Input is updated. This means that any updates requiring input must be placed in a subsequent phase.

## *Update*

Can also be though of as the *"core update* phase". In the following order, during this phase;

- The update system is run, executing all registered synchronous and asynchronous update functions.
- The entity system initializes and terminates entities.
- The physics system applies physics to all entities.
- The culling and level of detail system calculates which entities should be culled and their level of detail.
- Rendering matrices is updated. This means that any updates requiring the use of world-to-screen conversions must be placed in a subsequent phase.

## *Post-update*

Can also be though of as the *"application* phase". In the following order, during this phase;

- The update system is run, executing all registered synchronous and asynchronous post-update functions.
- The rendering system is updated, rendering all non-culled entities.
- The sound system is updated, playing all sounds for this frame.

## *Restrictions*

Certain restrictions is applied to what can be done in what phase and is checked in runtime in non-final builds. These are;

- Input states can not be retrieved in the pre-update phase.
- Moving, rotating or scaling entities can not be done in the post-update phase.