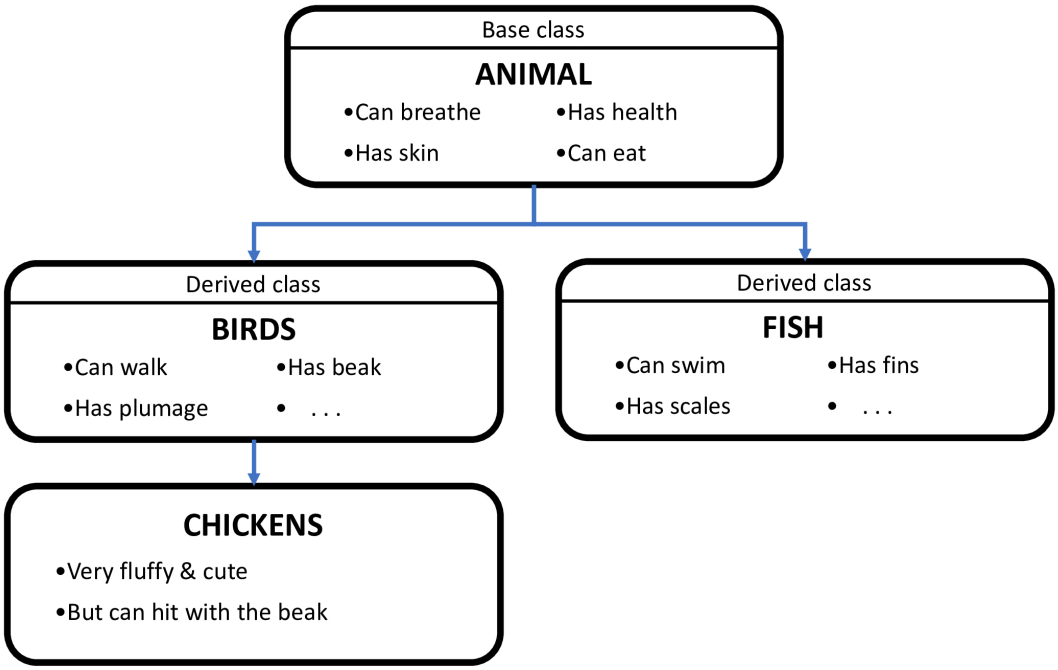
Introduction to ECS+

ECS+ is a new programming paradigm based on ECS and implemented in Apparatus.

Talking about OOP, we consider our practical task as multiplicity of special abstract things. In terms of UE4 these abstractions are named UObjects and over them we can apply such a principle like an inheritance. We create some main abstraction called „Base class“, define properties (i.e. variables and functions) and by deriving from that class we can create other abstractions which will get all the parent properties and define their own additional or distinctive ones. The most popular example in that way is the tree of animals inheritance, where the base class represent any animal exist, but the others (derived from the base one) represent separate animal sorts:



By using base class properties we can change current state of objects of derived class. But the problem is the game logic can become too scattered across the classes.

In game development there is such an approach named ECS. Apparatus“ new ECS+ paradigm may remind you an ECS approach with additions to common OOP technics provided with UE4. Apparatus uses some ECS features, but it has its own peculiarities. To understand all the theme, just view the next table:

ECS term	Apparatus term
Entity	Subject
Component	Detail
System	Mechanic
A group of Systems	Mechanism
Archetype	Fingerprint
Chunk	Belt

Source' explanation

At the lower level, „Subjective“ is an UE4 interface and any class that implements it have an array of enabled/disabled details. If the detail is enabled, then it“s „bIsAvtive“ bool variable is set to true. In the Plugin there are several implemented subjects which are:

- „SubjectiveActorComponent“ (based on „ActorComponent“),
- „SubjectiveUserWidget“ (based on „UserWidget“),
- „SubjectiveActor“ (based on „Actor“).

If you open the source, you can see they actually implement an „ISubjective“ interface. So it means you can use these classes as base classes for your own Blueprints and thus easily add to them necessary details.

Mechanism class is also an interface, and there are also some of them done:

- „MechanicalActor“ (inherited from „Actor“),
- „MechanicalGameModeBase“ (- „GameModeBase“),
- „MechanicalGameMode“ (- „GameMode“).

From:
http://turbanov.ru/wiki/ - **Turbopedia**

Permanent link:
http://turbanov.ru/wiki/en/toolworks/docs/apparatus/ecs?rev=1617393475

Last update: **2021/04/02 19:57**