# **The Journey of Princess Engine**

## **Making an entry point**

*Does the game run the engine or does the engine run the game?*  
I chose for the option that the game runs my engine and thus the game the entry point has.

## **Component system: ECS Lite (the best of both worlds)**

There will be a component manager that has object pools for every component(?)  
-> maybe this can be a vector but if you don’t reserve and the capacity is not enough this may become an expensive copy   
Game object has a vector of ids  
-> reserve to the max nr of different components we have in the engine)  
-> *How do we know which component it is and the place in the array?*

Bit Fields

* Main concern: The C standard leaves bitfield order to compiler implementation, this can become a problem when you want to see the ID as one solid number instead of 2
* Concern is not needed: we do simply not care about the order as we can access the 2 IDs separately and we will never think of this bitfield as one whole number.
* NOW (8/04)
  + 3 bits for component ID: 8 different components (can become more if need be)
  + 13 bits for array ID: should be more than enough (can become less if we need more bits for component ID but this shouldn’t cause problems as there still will be enough possible IDs for the array)

+ we do not need a base component -> no inheritance

- We need to take care of extra bookkeeping (concept of IDs)

It’s important that the components in the component manager lie in a contiguous array. This so we can loop over them very quickly in the critical update loop.

* Sort them on an “in use” bit
  + Extra bookkeeping: need to keep track of the “in use” part of the component
  + When you sort the array, the IDs in the game object get invalidated
* Let them point to the next one in line
  + Extra memory to point to the next
  + Looping over a linked list is as expensive in time complexity as looping over an array
  + We will be able to access them through the IDs (in an O(1) time fashion) as they are stored in a contiguous array, so pointer arithmetic is possible here.

# **References**

## **Entity component system**

* <https://wickedengine.net/2019/09/29/entity-component-system/>