DES

Generated by Doxygen 1.8.14

Contents

1 DES - Dorstijn Entity System **1**

2 Data Structure Index **3**

2.1 Data Structures 3

3 Data Structure Documentation **5**

3.1 AssemblagePool Struct Reference 5

3.2 EntityPool Struct Reference 6

3.2.1 Field Documentation 6

3.2.1.1 component_index 6

3.2.1.2 component_pool 6

3.2.1.3 guid 6

3.3 MetaComponentPool Struct Reference 7

3.3.1 Detailed Description 7

3.3.2 Field Documentation 7

3.3.2.1 component_pool 7

3.3.2.2 size 7

3.4 PositionComponentPool Struct Reference 7

3.5 VelocityComponentPool Struct Reference 7

Index **9**

Chapter 1

DES - Dorstijn Entity System

This is a minimalistic entity component system created by Robin A. Dorstijn, based off the T=Machine blog. The purpose of this ECS is to provide a backend for games.

TODOS

- Make all ECS tables (structs) dynamic memory
- Remove globals and starting passing around variables.
- Add SQL compatibility
- Rename components (to something with pools)and metaComponents

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

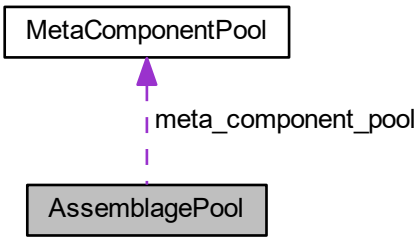
AssemblagePool 5
EntityPool 6
MetaComponentPool 7
PositionComponentPool 7
VelocityComponentPool 7

Chapter 3

Data Structure Documentation

3.1 AssemblagePool Struct Reference

Collaboration diagram for AssemblagePool:



Data Fields

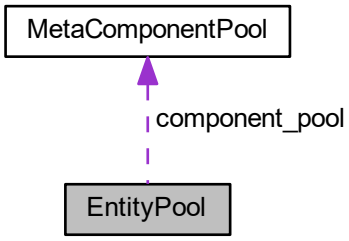
- int **assemblage_id** [ASSEMBLAGE_POOL_SIZE]
- MetaComponentPool * **meta_component_pool** [ASSEMBLAGE_POOL_SIZE]

The documentation for this struct was generated from the following file:

- assemblage.c

3.2 EntityPool Struct Reference

Collaboration diagram for EntityPool:



Data Fields

- int **size**
- int * guid
- MetaComponentPool ** component_pool
- int * component_index

3.2.1 Field Documentation

3.2.1.1 component_index

```
int* component_index
```

List of pointers to the relevant component pool

3.2.1.2 component_pool

```
MetaComponentPool** component_pool
```

Globally Unique IDentifier of an identity. `guid[i]` represents the i-th entity.

3.2.1.3 guid

```
int* guid
```

Amount of rows that the pool will be able to hold

The documentation for this struct was generated from the following file:

- des.h

3.3 MetaComponentPool Struct Reference

`#include <des.h>`

Data Fields

- u64 * **mask**
- int size
- void * component_pool

3.3.1 Detailed Description

Struct that creates a layer of abstraction between DES and the user's component pools.

3.3.2 Field Documentation

3.3.2.1 component_pool

`void* component_pool`

Size in rows that the component pool should be able to hold. This information is supplied by the user.

3.3.2.2 size

`int size`

Bitwise representation of the table. 1 means the slot/row is in use, 0 implies the row can be filled with a new entity.

The documentation for this struct was generated from the following file:

- des.h

3.4 PositionComponentPool Struct Reference

Data Fields

- float * **x_position**
- float * **y_position**

The documentation for this struct was generated from the following file:

- components/components.h

3.5 VelocityComponentPool Struct Reference

Data Fields

- float * **x_velocity**
- float * **y_velocity**

The documentation for this struct was generated from the following file:

- components/components.h

Index

AssemblagePool, 5

component_index
    EntityPool, 6
component_pool
    EntityPool, 6
    MetaComponentPool, 7

EntityPool, 6
    component_index, 6
    component_pool, 6
    guid, 6

guid
    EntityPool, 6

MetaComponentPool, 7
    component_pool, 7
    size, 7

PositionComponentPool, 7

size
    MetaComponentPool, 7

VelocityComponentPool, 7