

---

# **essence Documentation**

***Release 0.0.1***

**Hector Dearman**

October 31, 2015



<b>1</b>	<b>Example</b>	<b>3</b>
<b>2</b>	<b>What is an Entity-Component-System?</b>	<b>5</b>
<b>3</b>	<b>Where to find out more about ECSs</b>	<b>7</b>
<b>4</b>	<b>Indices and tables</b>	<b>9</b>



essence is an Entity-Component-System framework for Python.



---

**Example**

---

```
from essence import World, Component, System

class Position(Component):
    def __init__(self, x, y): self.x = x self.y = y

class Physics(System):
    def update(self, world):
        for e in world.entities_with(Position): e.get(Position).y -= 1

if __name__ == '__main__': world = World() world.systems.append(Physics()) player = world.create_entity()
    player.add(Position(1, 1))
    while True: world.update()
"""
```





---

## What is an Entity-Component-System?

---

An Entity-Component-System (or ECS) is an architectural pattern commonly used in games. Rather than model the world as a deep class hierarchy it instead divides the world into:

**Components** Which hold the data for particular aspect of a thing in the game world, for example a position or an animation or the ‘Health’ counter.

**Entities** Which collect a group of Components together and represent a concrete thing in the game world, for example the player character or an asteroid from asteroids.

**Systems** Which operate on a group of entities to implement a behavior, for example a ‘PhysicsSystem’ which updates the position component based on the velocity component and whether the entity has collided with any other entities.



---

## Where to find out more about ECSs

---

- [Wikipedia Article](#)

Contents:



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`