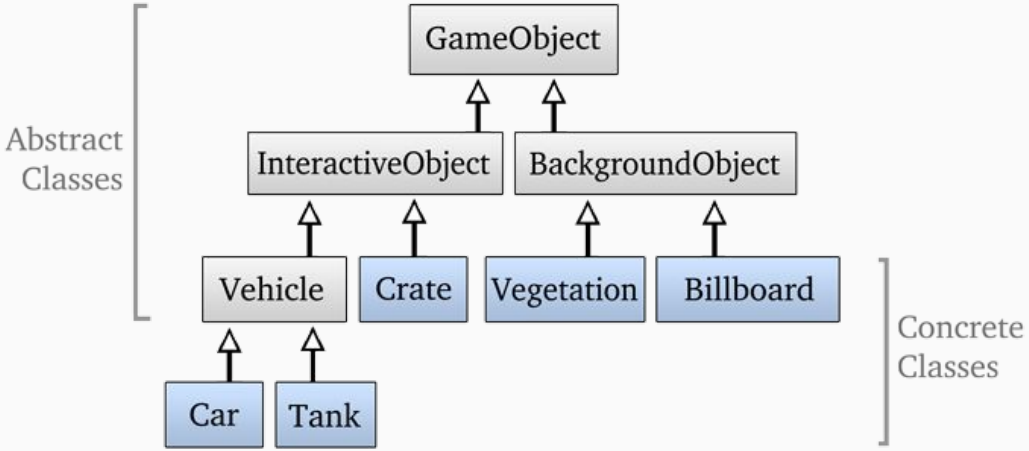
Game Dev: Entity Systems

Ricard Pillosu - UPC

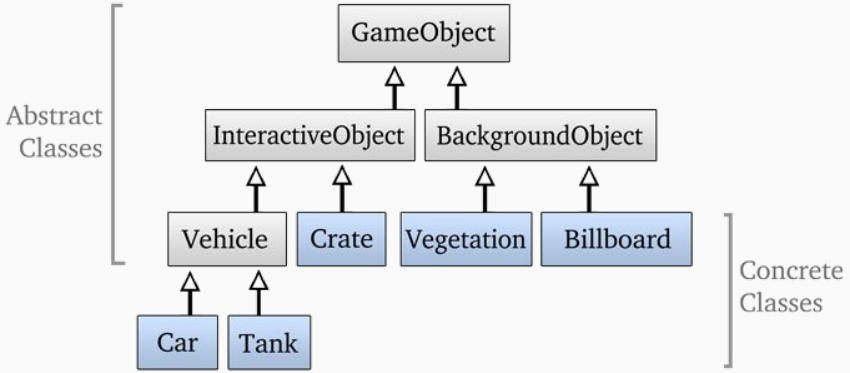
Entity Systems

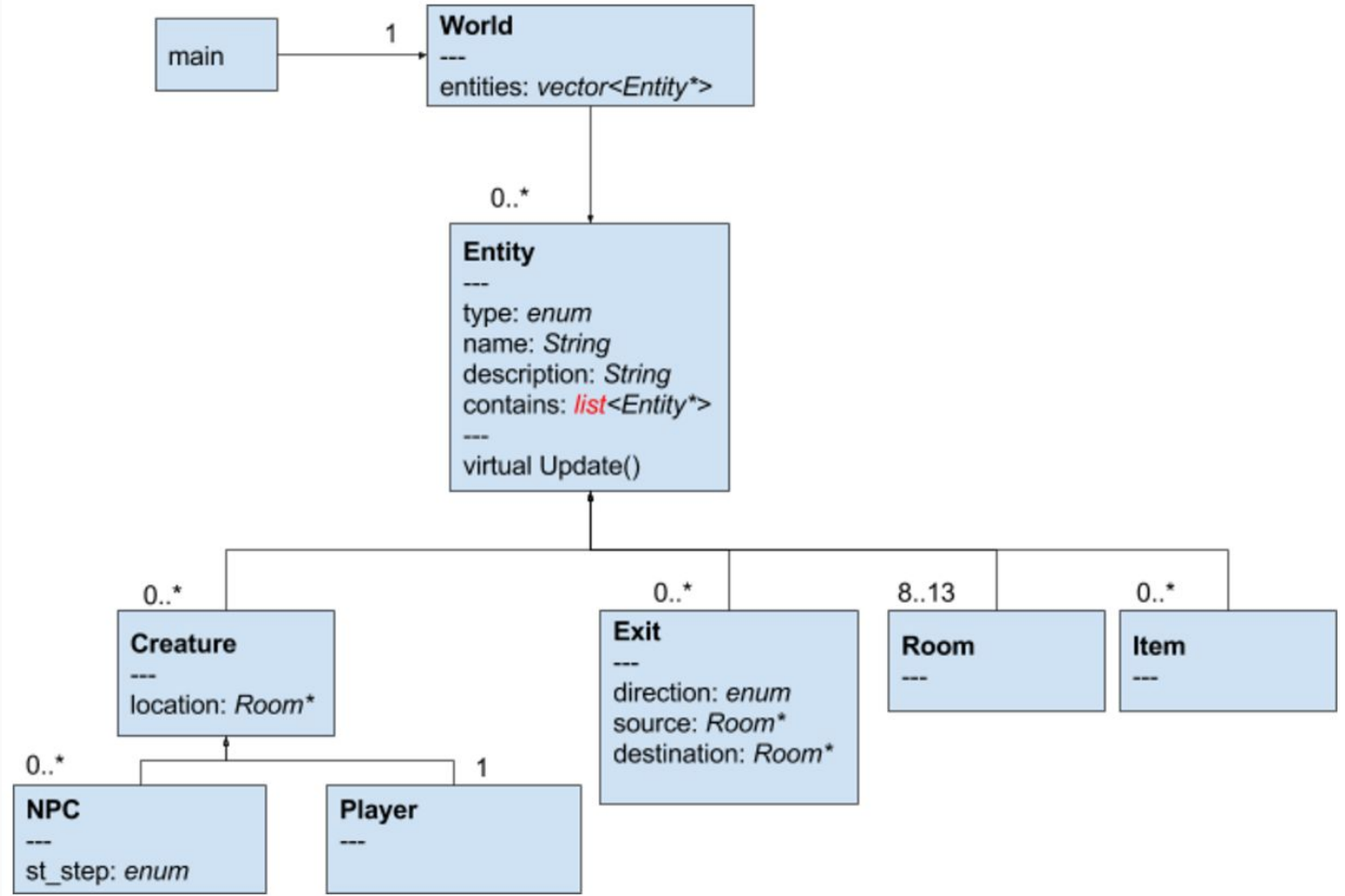
- Using OOP model we can describe all Entities in the game
- Exercise: Expand this structure to full UML with methods and properties:



Entity Systems

- The main advantage is that we can distribute data and functionality
- Data as class properties and functionality as class methods:
  - GameObject to contain a position
  - Interactive to have move() method
  - Vehicle to have speed and turn angle
  - Tank to *have fire* cannon methods ?
  - Car to have a radio ?
  - Billboard to have a OrientToCamera()





Remember Zork ?

Implementation: Timed Updates

```
bool EntityManager::Update(float dt)
{
	accumulated_time += dt;
	if(accumulated_time >= update_ms_cycle)
		do_logic = true;

	UpdateAll(dt, do_logic);

	if(do_logic == true) {
		accumulated_time = 0.0f;
		do_logic = false;
	}
	return true;
}
```

Implementation: Entity Factory

```
enum Types
{
	npc,
	player,
	room,
	exit,
	item,
	unknown
};
```

```
Entity* EntityManager::CreateEntity(Entity::Types type)
{
	static_assert(Entity::Types::unknown == 5, "code needs update");
	Entity* ret = nullptr;
	switch (type) {
		case Entity::Types::npc:    ret = new NPC();     break;
		case Entity::Types::player: ret = new Player();  break;
	}

	if (ret != nullptr)
		entities.push_back(ret);

	return ret;
}
```

Implementation: Creating Entities

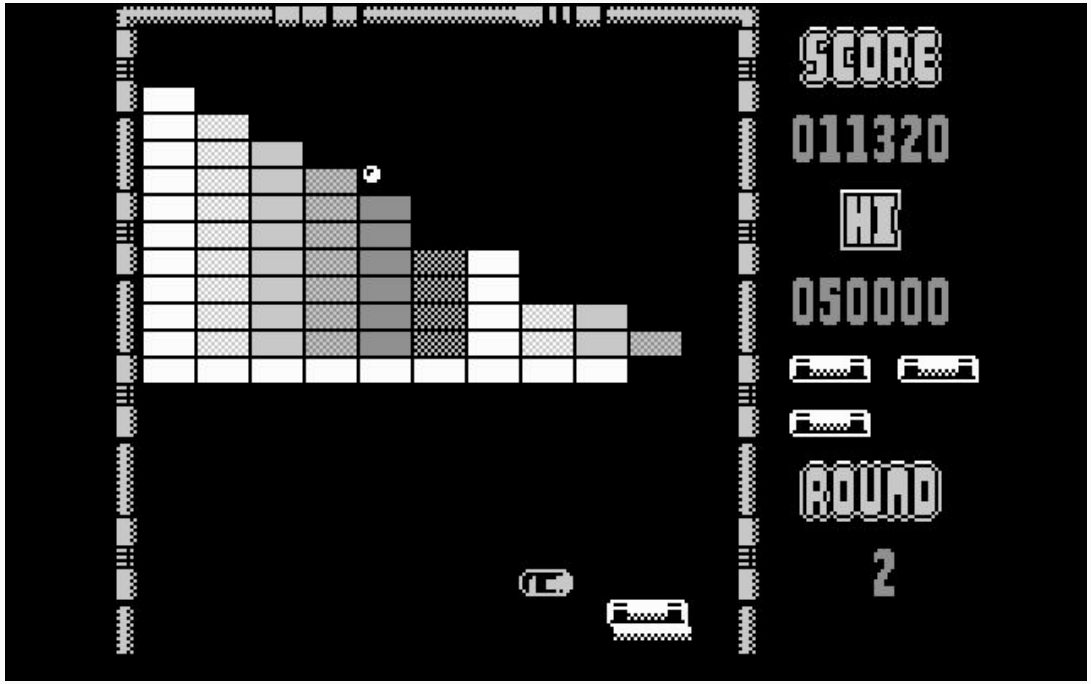
```
Entity::Entity(Types type) : type(type)
{}
```

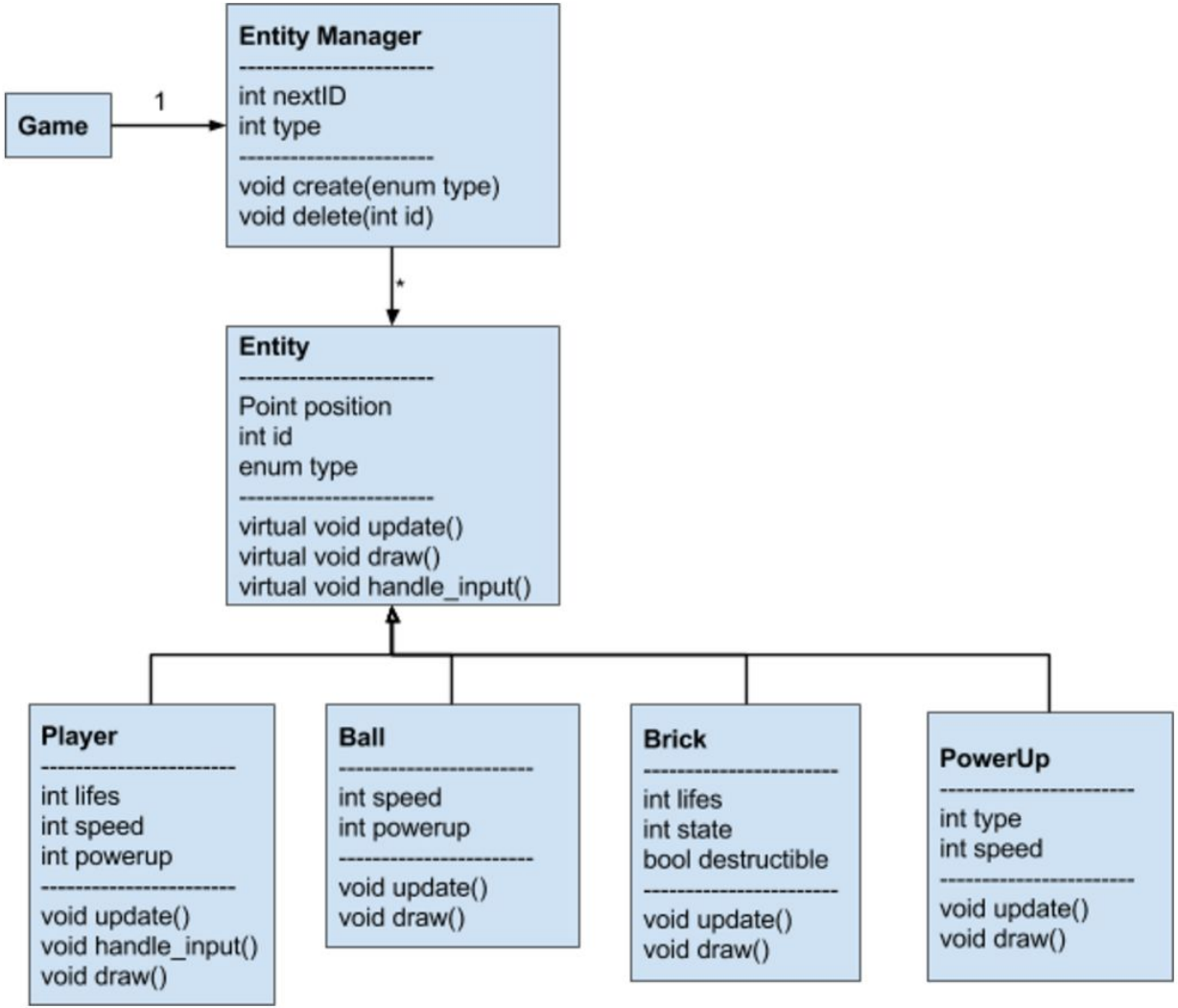
```
class Player : public Entity {
public:
	Player ();
	~Player ();
...
}
```

```
Player::Player() : Entity(Types::player)
{}
```

```
Player* player = (Player*) App->entities->CreateEntity(Entity::Types::player);
```

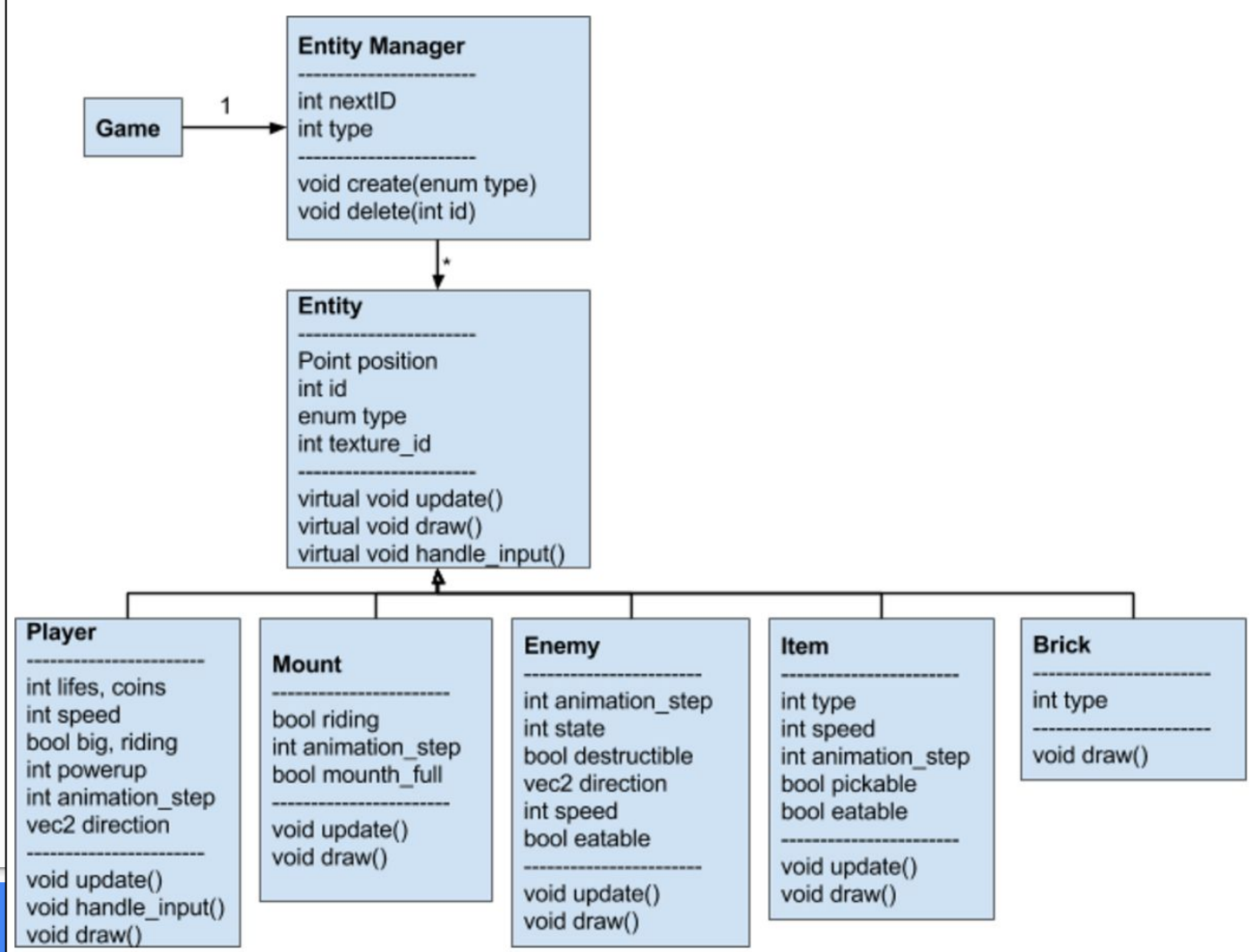
Write the UML for Mario Entity System





Assumptions: Bricks have state of blinking before removal. Walls are made of indestructible bricks. We have an enum with all entity types.

Write the UML for Mario Entity System



Assumptions: Background is not an entity. We have an enum with all entity types.

References

- Entity systems as explained here are considered old fashioned nowadays
- Component Based Systems are an evolution of Entity Systems
- More info here

Homework

Write down the UML for elements in this screenshot.

Code a simple *Entity System* that represents those entities.