# Updates

None yet.

# Grenade Rationale

### Main UMLs

*What classes exist in your extended system?*

The classes introduced in the extended system will be **Grenade** & **Throw**.

*What is role and responsibility of each new class?*

The role of **Grenade** is to be a throwable, takeable, leaveable entity that a SWActor can interact with. The grenade will explode violently and damage entities.

Following the occurrence of **Grenade,**

* \*\*\*

The role of **Throw** is to provide an action for the SWActor to use a grenade.

Following the occurrence of **Throw,**

* Entities in the location where the grenade is thrown lose 20 points.
* Entities in locations that can be reached in one step from the location where the grenade is thrown lose 10 points.
* Entities in locations that can be reached in two steps from the location where the grenade is thrown lose 5 points
* The instance of grenade is completely destroyed and disappears.

*How the new classes relate to and interact with the existing system.*

**\*\*\***

*How the (existing and new) classes will interact to deliver the required functionality.*

|  |  |
| --- | --- |
| - A grenade is an item that can be taken by an actor. | We can add a take affordance within the **Grenade** constructor.  **this**.addAffordance(**new** Take(**this**, m)); |
| - After it is taken, it can be either put down again (with a leave command), or thrown. | This is covered in the **Leave** class and the **Thrown class.**  Need to initiate a throw affordance for the grenade. |
| - When a grenade is thrown, it explodes violently, doing damage to both actors and other entities |  |
| - Entities in the location where the grenade is thrown lose 20 hitpoints. |  |
| - Entities in the location where the grenade is thrown lose 20 hitpoints. |  |
| - The actor that throws the grenade is not affected. |  |