

Use case: Attack

Summary: The user presses attackbutton (“x”) and the game checks if the attack hit something.

Priority: high

Extends: RunGame

Includes: HitEnemy (only represented here)

Participators: Actual player

Normal flow of events:

The player attacks and misses the target.

	Actor	System
1	The user presses the “attack”-button	
2		Attack-animation plays
3		Checks if the attack hit an enemy
		No target hit

Alternate flows:

The player attacks and hits the enemy whom survives.

	Actor	System
1	The user presses the "attack"-button	
2		Attack-animation plays
3		Checks if the attack hit an enemy
4		An enemy was hit. Plays hit animation
5		Calculates new health of the enemy.

Alternate flows:

The player attacks and hits the enemy whom dies - there are still other enemies in the room after.

	Actor	System
1	The user presses the "attack"-button	
2		Attack-animation plays
3		Checks if the attack hit an enemy
4		An enemy was hit. Plays hit animation
5		Calculates new health of the enemy. Health equals 0.
6		EnemyDies animation plays
7		Remove enemy from room
8		Check if room is cleared from enemies.
9		The room is not cleared.

Alternate flows:

The player attacks and hits the enemy whom dies - this was the last enemy and the room is now cleared from enemies.

	Actor	System
1	The user presses the "attack"-button	
2		Attack-animation plays
3		Checks if the attack hit an enemy
4		An enemy was hit. Plays hit animation
5		Calculates new health of the enemy. Health equals 0.
6		EnemyDies animation plays
7		Remove enemy from room
8		Check if room is cleared from enemies.
9		The room is emptied of enemies.
10		The room is now cleared and enemies will no longer respawn if the room is exited.