zombieattack.sol

```
pragma solidity ^0.4.25;

import "./zombiehelper.sol";

contract ZombieAttack is ZombieHelper {
  uint randNonce = 0;
  uint attackVictoryProbability = 70;

  function randMod(uint _modulus) internal returns(uint) {
    randNonce = randNonce.add(1);
    return uint(keccak256(abi.encodePacked(now, msg.sender, randNonce))) %
_modulus;
  }

  function attack(uint _zombieId, uint _targetId) external
onlyOwnerOf(_zombieId) {
    Zombie storage myZombie = zombies[_zombieId];
    Zombie storage enemyZombie = zombies[_targetId];
    uint rand = randMod(100);
    if (rand <= attackVictoryProbability) {
      myZombie.winCount = myZombie.winCount.add(1);
      myZombie.level = myZombie.level.add(1);
      enemyZombie.lossCount = enemyZombie.lossCount.add(1);
      feedAndMultiply(_zombieId, enemyZombie.dna, "zombie");
    } else {
      myZombie.lossCount = myZombie.lossCount.add(1);
      enemyZombie.winCount = enemyZombie.winCount.add(1);
      _triggerCooldown(myZombie);
    }
  }
}
```