```
1  using System;
2  using System.Collections.Generic;
3  using System.Text;
4
5  namespace coursework
6  {
7      class Weapon
8      {
9          public string name;
10         public int range, hitDie;
11         public bool throwable, finesse;
12         Random random = new Random();
13         /// <summary>
14         /// create a wepon
15         /// </summary>
16         /// <param name="name"></param>
17         /// <param name="range"></param>
18         /// <param name="hitDie"></param>
19         /// maximum damage done / sides on the die rolled for damage
20         /// <param name="throwable"></param>
21         /// if throwable, item should be dropped when attacked at a distance of
               more than 1 square
22         /// <param name="finesse"></param>
23         /// if finesse is true, dex modifier can be used instead of strength
24         public Weapon(string name, int range, int hitDie, bool throwable, bool
             finesse)
25         {
26             this.name = name;
27             this.range = range;
28             this.hitDie = hitDie;
29             this.throwable = throwable;
30             this.finesse = finesse;
31         }
32
33         public int attack(int Modifier, int enemyAC)
34         {
35             int damage = 0;
36             int d20 = random.Next(1, 21);
37             if (d20 + Modifier >= enemyAC)
38             {
39                 damage = random.Next(1, hitDie + 1);
40             }
41             return damage;
42         }
43
44         /// <summary>
45         /// check the distance from player to enemy to see if attack is
             possible with current weapon
46         /// if in water, melee attack not possible but long attack (thrown/bow)
               is
47         /// </summary>
48         /// <param name="enemyLocation"></param>
49         /// <param name="playerLocation"></param>
50         /// <param name="graph"></param>
51         /// <param name="thrown"></param>
52         /// indicates whether the weapon is lost after use (collect by going to
```

```
            square thrown to)
53          /// <returns></returns>
54          public bool inRange(int enemyLocation, int playerLocation, Graph graph,
                ref bool thrown)
55          {
56              bool inRange = false;
57              int distance = graph.dijksta(playerLocation)[enemyLocation, 0];
58              if (distance <= range)
59              {
60                  inRange = true;
61                  if (distance > 9 && throwable)
62                  {
63                      thrown = true;
64                  }
65              }
66              return inRange;
67          }
68      }
69  }
70
```