```
1 using System;
 2 using System.Collections.Generic;
 3 using DragonBallGame.CharacterClass;
 5 namespace DragonBallGame
 6 {
 7
       public class RecruitSystem
 8
            private static RecruitSystem _instance;
 9
10
            private List<Character> _recruitedCharacters;
            private Random _random;
11
12
13
            private RecruitSystem()
14
                _recruitedCharacters = new List<Character>();
15
16
                _random = new Random();
            }
17
18
19
            public static RecruitSystem Instance
20
            {
21
                get
22
                {
23
                    if (_instance == null)
24
25
                        _instance = new RecruitSystem();
26
27
                    return _instance;
28
                }
            }
29
30
31
            public RecruitResult RecruitRandomCharacter(Player player)
32
33
                if (player.Zeni < 100)</pre>
34
35
                    return new RecruitResult
36
37
                        Status = RecruitStatus.NotEnoughZeni
38
                    };
                }
39
40
41
                List<Character> availableCharacters = GetAvailableCharacters
                  (player);
42
                if (availableCharacters.Count == 0)
43
44
45
                    return new RecruitResult
46
                        Status = RecruitStatus.NoAvailableCharacters
47
48
                    };
```

```
...\Projects\DragonBallGame\GameSystems\RecruitSystem.cs
                                                                                  2
49
50
51
                player.DeductZeni(100);
52
53
                 // Select a random character
                 int randomIndex = _random.Next(availableCharacters.Count);
54
55
                 Character randomCharacter = availableCharacters[randomIndex];
56
                 Character existingCharacter = player.GetRecruitedCharacter
                   (randomCharacter.Name);
57
                 if (existingCharacter != null)
58
59
                 {
60
                     existingCharacter.OnDuplicateRecruited();
61
                     return new RecruitResult
62
63
                         Status = RecruitStatus.CharacterEvolved,
64
                         Character = existingCharacter
65
                     };
66
                 }
67
                else
68
                 {
69
                     return new RecruitResult
70
71
                         Status = RecruitStatus.NewCharacterRecruited,
72
                         Character = randomCharacter
73
                     };
                }
74
75
            }
76
77
            private List<Character> GetAvailableCharacters(Player player)
78
79
80
                List<Character> allCharacters = new List<Character>
81
                     new Goku(),
82
                     new Vegeta(),
83
                     new Gohan()
84
85
                 };
86
87
                List<Character> availableCharacters = new List<Character>();
88
89
                 foreach (Character character in allCharacters)
90
91
                     Character recruitedCharacter =
                       player.RecruitedCharacters.Find(c => c.Name ==
                       character.Name);
92
93
                     if (recruitedCharacter == null ||
                       recruitedCharacter.TransformationLevel <=
```

```
... \verb|\Projects| Dragon Ball Game \verb|\GameSystems| Recruit System.cs|
```

```
recruitedCharacter.MaxLevel)
                     {
94
                        availableCharacters.Add(character);
95
96
                     }
                }
97
98
                return availableCharacters;
99
            }
100
101
102
        }
103 }
```

3