

```
1  using CustomProject.GameObjects;
2  using SplashKitSDK;
3
4  namespace CustomProject
5  {
6      public class GameMap : GameObject
7      {
8          // Game map height and width
9          private int _mapWidth, _mapHeight;
10
11         // Game map objects
12         private Bitmap _background;
13         private Bitmap _jungle, _sky, _sea, _desert;
14
15         private List<Land> _lands;
16         private List<CollectibleBomb> _bombs;
17         private List<CollectibleGold> _golds;
18         private List<Sword> _sword;
19         private List<SpeedPotion> _speedPotion;
20         private List<JumpPotion> _jumpPotion;
21
22         // For reading maps and draw
23         private string[] _txtMap;
24
25         public GameMap(string[] ids, string name, string description) :     ↪
26             base(ids, name, description)
27         {
28             _lands = new List<Land>();
29             _bombs = new List<CollectibleBomb>();
30             _golds = new List<CollectibleGold>();
31             _sword = new List<Sword>();
32             _speedPotion = new List<SpeedPotion>();
33             _jumpPotion = new List<JumpPotion>();
34
35             ReadMapFromFile("map.txt");
36             LoadBackgrounds();
37         }
38
39         private void ReadMapFromFile(string filename)
40         {
41             StreamReader _rd = new StreamReader(filename);
42             _txtMap = _rd.ReadToEnd().Split(new[] { '\r', '\n' },
43                 StringSplitOptions.RemoveEmptyEntries);
44             _rd.Close();
45
46             _mapHeight = _txtMap.Length;
47             _mapWidth = _txtMap[0].Length;
48         }
49     }
```

```
48     private void LoadBackgrounds()
49     {
50         _jungle = SplashKit.LoadBitmap("jungle", "jungle.png");
51         _sky = SplashKit.LoadBitmap("sky", "sky.png");
52         _sea = SplashKit.LoadBitmap("sea", "marine.png");
53         _desert = SplashKit.LoadBitmap("desert", "desert.jpg");
54         _background = _jungle;
55     }
56
57     // Set up the 2D array of the text map
58     public void SetUpGameMap()
59     {
60         for (int i = 0; i < _mapHeight; i++)
61         {
62             for (int j = 0; j < _mapWidth; j++)
63             {
64                 switch (_txtMap[i][j])
65                 {
66                     case '#':
67                         _lands.Add(new Land(j * 50, i * 50));
68                         break;
69                     case 'x':
70                         _bombs.Add(new CollectibleBomb(j * 50, i * 50));
71                         break;
72                     case 'o':
73                         _golds.Add(new CollectibleGold(j * 50, i * 50));
74                         break;
75                     case 's':
76                         _sword.Add(new Sword(j * 50, i * 50));
77                         break;
78                     case 'a':
79                         _speedPotion.Add(new SpeedPotion(j * 50, i * 50));
80                         break;
81                     case 'j':
82                         _jumpPotion.Add(new JumpPotion(j * 50, i * 50));
83                         break;
84                     default:
85                         break;
86                 }
87             }
88         }
89     }
90
91     // Draw the game map using the 2D arry of text map
92     public void Draw()
```

```
93         {
94             SplashKit.DrawBitmap(_background, Camera.X, Camera.Y);
95             foreach (Land land in _lands)
96             {
97                 land.Draw();
98             }
99
100            foreach (CollectibleGold gold in _golds)
101            {
102                gold.Draw();
103            }
104
105            foreach (CollectibleBomb bomb in _bombs)
106            {
107                bomb.Draw();
108            }
109
110            foreach (Sword sword in _sword)
111            {
112                sword.Draw();
113            }
114
115            foreach (SpeedPotion potion in _speedPotion)
116            {
117                potion.Draw();
118            }
119
120            foreach (JumpPotion potion in _jumpPotion)
121            {
122                potion.Draw();
123            }
124        }
125
126        public int Width { get => _mapWidth * 50; }
127        public int Height { get => _mapHeight * 50; }
128        public List<Land> Lands { get => _lands; }
129        public List<CollectibleBomb> Bombs { get => _bombs; }
130        public List<CollectibleGold> Golds { get => _golds; }
131        public List<Sword> Swords { get => _sword; }
132        public List<SpeedPotion> SpeedPotion { get => _speedPotion; }
133        public List<JumpPotion> JumpPotion { get => _jumpPotion; }
134        public Bitmap Background { get => _background; set => _background = value; }
135        public Bitmap Sea { get => _sea; }
136        public Bitmap Jungle { get => _jungle; }
137        public Bitmap Sky { get => _sky; }
138        public Bitmap Desert { get => _desert; }
139    }
140 }
```