Contents

1. Method List and Comments

A detailed description of each argument is attached to the code.

```
namespace Assets.oojjrs.Script
{
    public class MyRandom
    {
        /// Returns a random integer that is within a specified range.
        /// [minValue, maxValue)
        public static int Range(int minValue, int maxValue);

        /// Returns a random floating-point number that is within a specified range.
        /// [minValue, maxValue]
        public static float Range(float minValue, float maxValue);

        /// Returns a random floating-point number that is within a specified range.
        /// [minValue, maxValue]
        public static double Range(double minValue, double maxValue);

        /// Returns a randomly selected entry in an entries.
        public static T Select<T>(IEnumerable<T> entries);

        /// Returns a N randomly selected entries in an entries.
        /// Returns Empty if count is less than or equal to 0, and returns the entries in random order if count is greater than or equal to
the number of entries.
        public static IEnumerable<T> Select<T>(IEnumerable<T> entries, int count);

        /// Returns a randomly selected entry in an entries.
        /// Each item has a probability of being selected equal to the item's float value from the sum of the floats of all items.
        /// If all float values are 1, it will have a value of 1/N, which works the same as Select<T>(entries).
        public static T Select<T>(IEnumerable<(T, float)> entries);

        /// Returns a N randomly selected entries in an entries.
        /// Items already selected in entries can also be selected again.
        public static IEnumerable<T> SelectWithRepetition<T>(IEnumerable<T> entries, int count);
```

```
        /// Returns a N randomly selected entries in an entries.
        /// Items already selected in entries can also be selected again.
        /// Each item has a probability of being selected equal to the item's float value from the sum of the floats of all items.
        /// If all float values are 1, it will have a value of 1/N, which works the same as SelectWithRepetition<T>(entries, count).
        public static IEnumerable<T> SelectWithRepetition<T>(IEnumerable<(T, float)> entries, int count);

        /// Returns the entries in random order.
        public static IEnumerable<T> Shuffle<T>(IEnumerable<T> entries);
    }
}
```

All methods declare a method that can receive a random seed as the last argument.

```
namespace Assets.oojjrs.Script
{
    public class MyRandom
    {
        public static int Range(int minValue, int maxValue, Random random);
        public static float Range(float minValue, float maxValue, Random random);
        public static double Range(double minValue, double maxValue, Random random);
        public static T Select<T>(IEnumerable<T> entries, Random random);
        public static IEnumerable<T> Select<T>(IEnumerable<T> entries, int count, Random random);
        public static T Select<T>(IEnumerable<(T, float)> entries, Random random);
        public static IEnumerable<T> SelectWithRepetition<T>(IEnumerable<T> entries, int count, Random random);
        public static IEnumerable<T> SelectWithRepetition<T>(IEnumerable<(T, float)> entries, int count, Random random);
        public static IEnumerable<T> Shuffle<T>(IEnumerable<T> entries, Random random);
    }
}
```