Determining drawing result

# Calculation method

**RNGCryptoServiceProvider** is used as the random number generator for determining the drawing result. The **GetBytes** method is used to fill 2 bytes with a cryptographically strong sequence of random values. The byes were converted into a 16-bit unsigned integer (2 bytes non-negative). The number is converted into a regular 32-bit integer, divided by 1000, then taking the remainder as the result, **n**. The range of **n** is from 0 to 999.

The remainder is compared with a probability in percentage (1~100), **p**, set by user for determining the winning result. If remainder is less than the probability multiplied by 10, then the result is a winning result otherwise the result is a losing result.

1000 is chosen to provide a greater set of numbers for the drawing result.

P is a number set in the **appSettings** section of the **web.config**

**Current phase**: a non-changing random number is determining the winning result. The probability is fixed.

# Calculation coding

byte[] bInt = new byte[2];

RNGCryptoServiceProvider cr = new RNGCryptoServiceProvider();

cr.GetBytes(bInt);

int n = Convert.ToInt32(BitConverter.ToUInt16(bInt, 0)) % 1000;

double p = double.Parse(ConfigurationManager.AppSettings["DrawingChance"]);

if (n < Convert.ToInt32(p \* 10))

return true;

else

return false;