

Bayesian Network

```
public float generateProbability()
{
    //The method will require two values in order to calculate a probability
    float probabilityIfTrue=0, probabilityIfFalse = 0, probability = 0;

    //This boolean represents the first stage of the method to work out the probability if
    firing = true
    firstPart = true;

    probabilityIfTrue = fireTrue * (abovePlayer(AbovePlayer, firstPart)) *
    (aboveBarrier(AboveBarrier, firstPart) * (withinRaddius(WithinRadius, firstPart));

    //The boolean represents that the first stage has now ended so that probabilities
    firstPart = false;

    probabilityIfFalse = fireFalse * (abovePlayer(AbovePlayer, firstPart)) *
    (aboveBarrier(AboveBarrier, firstPart) * (withinRaddius(WithinRadius, firstPart));

    //Normalise to complete the equation
    probability = probabilityIfTrue / (probabilityIfTrue + probabilityIfFalse);

    return probability;
}

/// <summary>
/// Retrieves the correct probability given that an enemy is/is not above the player
/// </summary>
/// <param name="isAbovePlayer">Is the enemy above the player?</param>
/// <param name="firstPart">Has the first part of the calculation been completed?</param>
/// <returns></returns>
public float abovePlayer(bool isAbovePlayer, bool firstPart)
{
    if (isAbovePlayer == true)
    {
        return abovePlayTrue;
    }

    else
    {
        if(firstPart == true)
        {
            return (1 - abovePlayTrue);
        }

        else
        {
            return (1 - abovePlayFalse);
        }
    }
}

/// <summary>
/// Retrieves the correct probability given that an enemy is/is not above a barrier
/// </summary>
/// <param name="isAboveBarrier">Is the enemy above a barrier?</param>
/// <param name="firstPart">Has the first part of the calculation been completed?</param>
/// <returns></returns>
public float aboveBarrier(bool isAboveBarrier, bool firstPart)
{
    if (isAboveBarrier == true)
    {
        return aboveBarTrue;
    }

    else
    {
        if(firstPart == true)
        {
            return (1 - aboveBarTrue);
        }

        else
        {
            return (1 - aboveBarFalse);
        }
    }
}
```

```
    }  
  }  
}  
  
/// <summary>  
/// Retrieves the correct probability given that an enemy is/is not within the radius of the  
player  
/// </summary>  
/// <param name="isWithin">Is enemy within radius?</param>  
/// <param name="firstPart">Has the first part of the calculation been completed?</param>  
/// <returns></returns>  
public float withinRaddius(bool isWithin, bool firstPart)  
{  
    if (isWithin == true)  
    {  
        return withinRadTrue;  
    }  
  
    else  
    {  
        if(firstPart == true)  
        {  
            return (1 - withinRadTrue);  
        }  
  
        else  
        {  
            return (1 - withinRadFalse);  
        }  
    }  
}  
}
```