In this term project of Digital Image Synthesis, I implemented poisson dist
distribution on PBRT and compared the results of poisson sampler and random
sampler. Below are the properties of three samplers:

unifom grid samples  random samples  poisson disk distribution samples

The procedure of implementation is to copy `src/samplers/random.h` and
`src/sampler/random.cpp`, and rename them as `poisson.h` and `poisson.cpp`. There is also
a need to create/modify a new poisson class in those files.

After having the basic class, I defined a structure in `PoissonDisk`, containing every
coordinate \((x, y)\) of each poisson disk and its available angle degree. This is because
we need to choose a point on perimeter of a poisson disk when doing dart throwing.
Therefore, in order to create a proper poisson disk distribution, we must have a
specific range of degree for a point to select from.

Next, in the class of `PoissonDiskSampler`, I created two functions:
   - `void boundary_maximum();`
   - `bool Check_BBox();`
   The upper one is to create a new poisson disk and calculate its angle degree;
the lower one is to check if the disk is in the proper range.

In `poisson.cpp`, I created:
   - `float distance();`
   - `void intersect_PoissonDisk();`
   - `void computeTheta();`
   - `void intersect_theta();`
The first function is used to compute the distance of two points. The second and the
fourth are called by `boundary_maximum`, in purpose of inserting new poisson disk,
calculate available angle degree, and compute the disk inserting angle according to an
angle degree. Therefore, with the degree angle and radius, a new coordinate can be
computed.
Below are the comparisons of results:

Rendered with random samples, pixelsamples[4], took 12.2 seconds
Rendered with pisson disk distribution samples, pixelsamples[4], took 22.2 seconds
These two images above are the shadow areas in the outputs of random samples pixelsamples[4] and poisson samples pixelsamples[4]. The left one is with random samples, while the right one is with poisson samples.

It can be seen that there is more noise in the left shadow. Though applied with pixelsamples[4], the image on the right hand side looks better for being rendered with poisson disk distribution.
The following is the comparison of the results of random samples and poisson disk distribution samples with pixelsamples[256]:

random samples
poisson disk distribution samples

If we magnified the image above, we can see that the head of the gree dinosaur is clearer than the random-sampled one. Also, the result above is visibly brighter, especially in the margins.

Here is the execution procedure of the files:

First, put poisson.h and poisson.cpp into src/samplers.
Second, add

```cpp
#include "samplers/poisson.h"
```

into core/api.cpp

add the followings at about line 639

```cpp
else if(name == "poisson")
    sampler = CreatePoissonDiskSampler(paramSet, film, camera);
```

Then, make the whole pbrt again for sure.
If you want to use poisson sampler for rendering, you just modify the pbrt file you want to render and change Sampler into "poisson", then you are all set. :)}