Flashbang post process effect

Description

Flashbang is post-process effect that used to visualize explosion of flash bang grenades. It is consist of several phases. Firstly game screen was captured in time of explosion. First several seconds white screen is showed indicating blindness effect. In next step white screen faded out while stored game screenshot faded in. Than stored game screen slowly faded out with normal game rendering is faded in.

Associated classes and structures

FlashBangEffectParams	Definition of single flash bang effect parameters.
FlashbangVisualController	Class that manages multiple flash bang effect, controls its behavior through lifetime.

class FlashbangVisualController

This class manages multiple flash bang visual effects. Performs world-space to screen-space coordinate conversion, adjusts effects strength depending on viewer distance and schedule post-process effects into post-process chain. Visibility of explosion position is also determined, to prevent showing effect if explosion epicenter is occluded.

Sources: FlashbangVisualController.h, FlashbangVisualController.cpp

Important methods

void IssueVisibilityQuery()

Issue hardware visibility query. This operation consist of d3d query issue and sphere rendering operations.

void CheckEffectVisibility()

Check if effect should be visible using query issued by IssueVisibilityQuery function. Result of this check stored in internal variable.

void StartEffect(const FlashBangEffectParams &params, float delay)

Start flash bang visual effect.

Parameters:
params – parameters of visual effect to start.
delay – start effect delay, in seconds.

void ApplyPostFXEffects()

Apply post-process effects to post-process effect chain. This function is act like update loop. After current effect phase determination, corresponding post-process effects is added, forming necessary visual picture.

struct FlashBangEffectParameters

This structure defines new flash bang effect parameters

Sources: FlashbangVisualController.h

Important members

float solidColorTimeFraction
White color display time. In fraction [0..1] of total effect duration.

float color2StillPictureTimeFraction
Transition from solid color to still picture. In fraction [0..1] of total effect duration

r3dVector pos
Explosion epicenter world position.

float duration
Total effect time. In seconds.

Usage example

This class intended to be in single instance, so we create one global manager for all flash bang effects:

```
FlashbangVisualController gFlashbangVisualController;
```

Because we need update loop, and ApplyPostFXEffects function act like update, we should add call in place where all post FX setup reside:

```
void r3dDefferedRenderer::PostProcess()
{
…
   gFlashbangVisualController.ApplyPostFXEffects();
…
}
```

Now we can add flash bangs to the manager, and it will do all the work related to updating effect state and post fx submit. For example look at onExplode function of obj_Grenade object:

```
void obj_Grenade::onExplode()
{
…
   // for now hard coded for flash bang grenade
   if (m_Weapon->m_itemID == 101137)
   {
     …
      FlashBangEffectParams fbep;
      fbep.duration = GPP->c_fFlashBangDuration * str;
      fbep.pos = GetPosition();
      gFlashbangVisualController.StartEffect(fbep);
    …
```

```
    }
  ...
}
```