Creating Particle Explosions

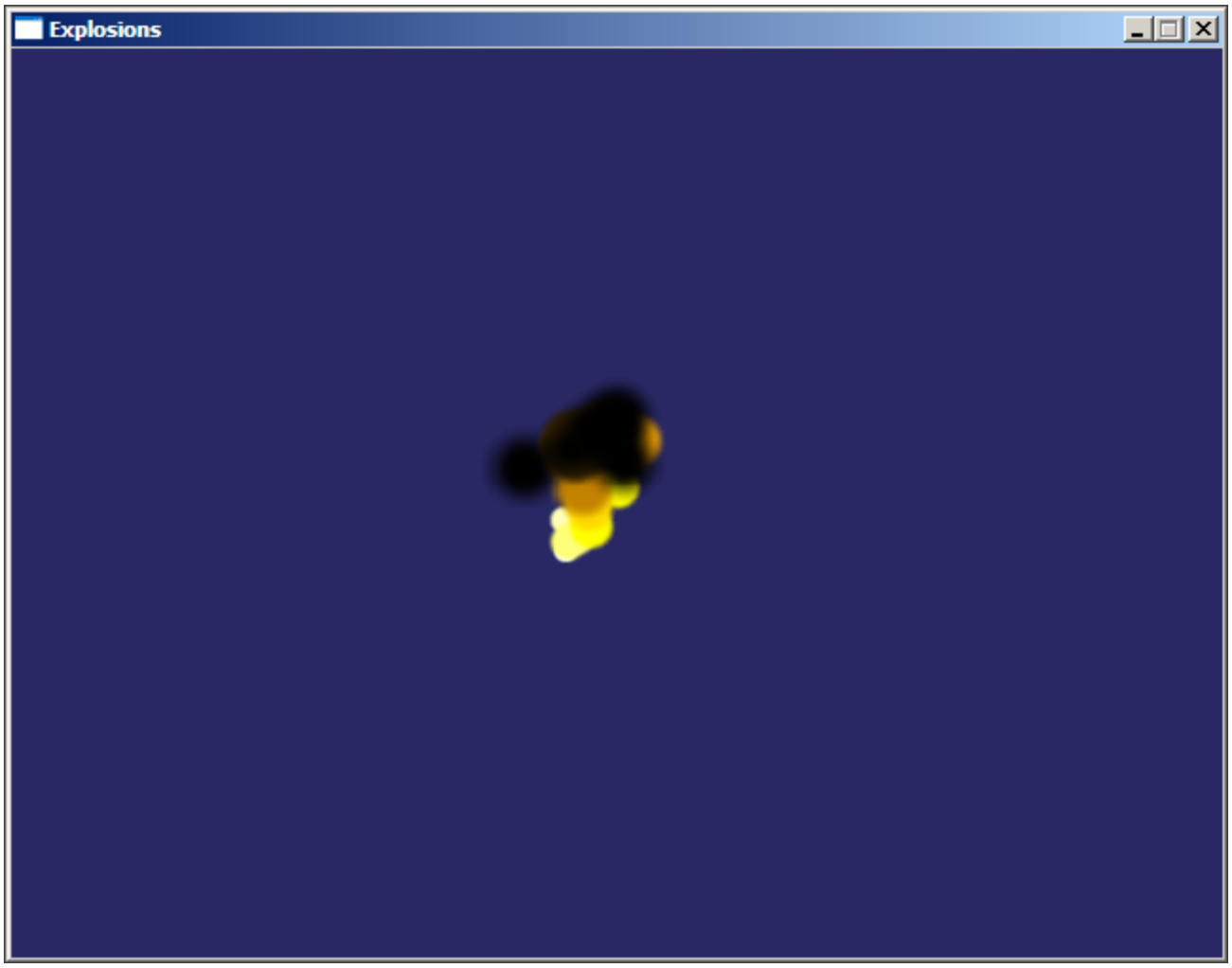
Do you like blowing things up? Of course you do. Everyone does.

How about a nice way to have particle explosions in your game? Great for bullet stormers.

This explosion is quite simple. It is composed of the following objects:

1. A blank object
2. A spawner attached to the blank object above
3. A separate particle object which serves as the type of object the spawner shoots out

This blank object can be placed anywhere. We'll limit it to 10 spawned particle objects, then it will die. We are looking for this kind of effect (though a still image doesn't do it justice):



For this tutorial, the mouse will be used to set an explosion. Click anywhere on screen, and you can click as much and as quickly as you like to fill the screen with booms.

Setting Up And Input

Start with a blank, working orx project.

In your Init() method, add a handler for input so we can read the mouse to set explosions:

```
    orxEvent_AddHandler(orxEVENT_TYPE_INPUT, EventHandler);
```

Our event handler method with look like this:

```
    orxSTATUS orxFASTCALL EventHandler(const orxEVENT *_pstEvent) {

        if (_pstEvent->eType == orxEVENT_TYPE_INPUT){

            if(orxInput_IsActive("LeftMouse") &&
orxInput_HasNewStatus("LeftMouse") ) {

                orxVECTOR mouseLocalScreenPosition = { 0.0, 0.0, 0.0 };
                orxMouse_GetPosition(&mouseLocalScreenPosition);

                ExplodeAt(mouseLocalScreenPosition);
            }

        }

        return orxSTATUS_SUCCESS;
    }
```

In this event handler method, we are detecting if the left mouse has been clicked.

The “LeftMouse” being called will be defined in our config:

```
    [Input]
    SetList = TutorialInput

    [TutorialInput]
    MOUSE_LEFT  = LeftMouse
    KEY_ESCAPE  = Quit
```

ExplodeAt() And Graphic

An ExplodeAt() function being called. This is where we create a an explosion and position it. We'll send it our current mouse position.

The ExplodeAt() function looks like this:

```
    void ExplodeAt(orxVECTOR position){
        orxOBJECT *explosion = orxObject_CreateFromConfig("Exploder");
        orxObject_SetPosition(explosion, &position);
```

```
    }
```

Now, what are our explosion frames going to look like? Below are the animation frames of a single sub-explosion. This is a 594 x 46 pixel animation sprite set, each frame being 46 x 46 pixels:



As mentioned at the beginning, the blank explosion object will have a spawner, and this spawner will spawn 10 copies of these individual sub-explosions. Each will follow quickly after the other and be randomly placed away from the centre blank object. This will help the explosion look a little more natural. And each will be different from the last.

Project Setup Config

Everything else happens in the config. First, some more screen set up stuff:

```
[Mouse]
ShowCursor = true

[Display]
ScreenWidth    = 640
ScreenHeight   = 480
Title      = Explosions

[Viewport]
Camera          = Camera
BackgroundColor = (40, 40, 100)


[Camera]
FrustumWidth  = @Display.ScreenWidth
FrustumHeight = @Display.ScreenHeight
FrustumFar    = 2.0
Position      = (320.0, 240.0, -1.0)
```

Explosion Animation

Next, the sub explosion object, graphics and animation frames:

```
[ExplodeObject]                      ;explosion object, the actual orange
flare
Graphic           = ExplodeGraphic         ;supplies default image size for
all frames
AnimationSet      = ExplosionSet
Position       = (-20,-20,0) ~ (20,20,0)     ;<--- random positions offset
from the centre of the explosion core
```

```
LifeTime       = 0.35                ;<!--- each sub-explosion will die
after 0.35 seconds.

[ExplodeGraphic]
Texture     = explosion.png
TextureSize = (46, 46, 0)
Pivot       = center

[ExplosionSet]
Texture       = explosion.png
KeyDuration   = 0.03
StartAnim     = ExplodeAnim      ; Ensure our only animation is the default.
FrameSize     = (46, 46, 0)
ExplodeAnim-> =                  ; After ExplodeAnim has completed, go
nowhere. Stop animating.
```

The Spawner

Finally, the blank object and the explosion spawner:

```
[Exploder]                   ;invisible object
Spawner  = ExplosionSpawner
Position = (100,100,0)
LifeTime = 2                   ;<--- Each blank explosion object will die
after 2 seconds.

[ExplosionSpawner]
Object     = ExplodeObject
TotalObject    = 10               ;<!-- only have 10 sub-explosions
WaveSize   = 1
WaveDelay  = 0.02
```

Finished

That's it. Hope you like the effect. Of course you could tweak all kinds of things like having the spawned objects move away from the centre, or spin, of zoom. Designing up your own explosion objects will give a variety of different results.

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
https://www.orx-project.org/wiki/en/tutorials/animation/particle_explosions?rev=1598882079

Last update: **2025/09/30 17:26 (14 months ago)**