

# Part 20 - Text and Game Over

For our last hurrah, we'll add a score object in the top left corner of the screen and increase it for each monster shot and when the star is collected.

First the score text, graphic and object:

```
[ScoreText]
String = 000

[ScoreGraphic]
Text = ScoreText
Pivot = top left

[ScoreObject]
Graphic = ScoreGraphic
Position = (-380, -280, 0)
Scale = 2
```

Interesting to note the special Text definition. ScoreGraphic uses this instead of a texture from file. And the ScoreObject uses the graphic section as others before.

Now, in code, add a score variable and a scoreObject at the top of the file:

```
orxOBJECT *hero;
orxOBJECT *herosGun;
orxOBJECT *scoreObject;
orxS16 score = 0;
```

Next, create a score object and assign it to the scoreObject variable in the Init() function:

```
hero = orxObject_CreateFromConfig("HeroObject");
herosGun = (orxOBJECT*)orxObject_GetChild(hero);
scoreObject = orxObject_CreateFromConfig("ScoreObject");
```

Add a little function to increase the score and update the ScoreObject:

```
void UpdateScore(int increase){
    score += increase;

    orxCHAR formattedScore[6];
    orxString_NPrint(formattedScore, sizeof(formattedScore), "%d", score);

    orxObject_SetTextString(scoreObject, formattedScore);
}
```

Add 250 points whenever a monster is hit with a bullet. Add UpdateScore() function calls to:

```
if (orxString_Compare(senderObjectName, "BulletObject") == 0) {
```

```
CreateExplosionAt0bject(pstRecipient0bject, "JellyExploder");
orx0bject_SetLifeTime(pstSender0bject, 0);
orx0bject_SetLifeTime(pstRecipient0bject, 0);
UpdateScore(250);
}

if (orxString_Compare(recipient0bjectName, "Bullet0bject") == 0){
CreateExplosionAt0bject(pstSender0bject, "JellyExploder");
orx0bject_SetLifeTime(pstSender0bject, 0);
orx0bject_SetLifeTime(pstRecipient0bject, 0);
UpdateScore(250);
}
```

And of course, 1000 points bonus if the star is reached. Add the following code:

```
if (orxString_Compare(sender0bjectName, "Star0bject") == 0){
orx0bject_SetLifeTime(pstSender0bject, 0);
UpdateScore(1000);
}

if (orxString_Compare(recipient0bjectName, "Star0bject") == 0){
orx0bject_SetLifeTime(pstRecipient0bject, 0);
UpdateScore(1000);
}
```

Compile and run it. Shoot some monsters and collect the star and observe the score increasing.

That takes care of having an active score object.

A final thing for the game will be to add a game over panel when the hero dies. For this, we'll have an object that gets created using a timeline track after a two second delay. First, a game over asset:



Save this into the data/texture folder as gameover.png.

Create a simple gameover graphic and object:

```
[GameOverGraphic]
Texture = gameover.png
Pivot   = center

[GameOverObject]
Graphic  = GameOverGraphic
Position = (0, 0, -0.1)
```

Create a timeline track with a single command to create the GameOverObject:

```
[PopUpGameOverTrack]
2 = Object.Create GameOverObject
```

Then finally, change the physics handler code, so that when the hero is destroyed, create the timeline track. After a two second delay, the GameOverObject will be created on screen:

```
if (orxString_Compare(recipientObjectName, "HeroObject") == 0 &&
    orxString_Compare(senderObjectName, "MonsterObject") == 0
){
    CreateExplosionAtObject(pstRecipientObject, "HeroExploder");
    orxObject_SetLifeTime(pstSenderObject, 0);
    orxObject_Enable(pstRecipientObject, orxFALSE);
    orxObject_AddTimeLineTrack(scene, "PopUpGameOverTrack");
}

if (orxString_Compare(senderObjectName, "HeroObject") == 0 &&
    orxString_Compare(recipientObjectName, "MonsterObject") == 0
){
    CreateExplosionAtObject(pstSenderObject, "HeroExploder");
    orxObject_SetLifeTime(pstRecipientObject, 0);
    orxObject_Enable(pstSenderObject, orxFALSE);
    orxObject_AddTimeLineTrack(scene, "PopUpGameOverTrack");
}
```

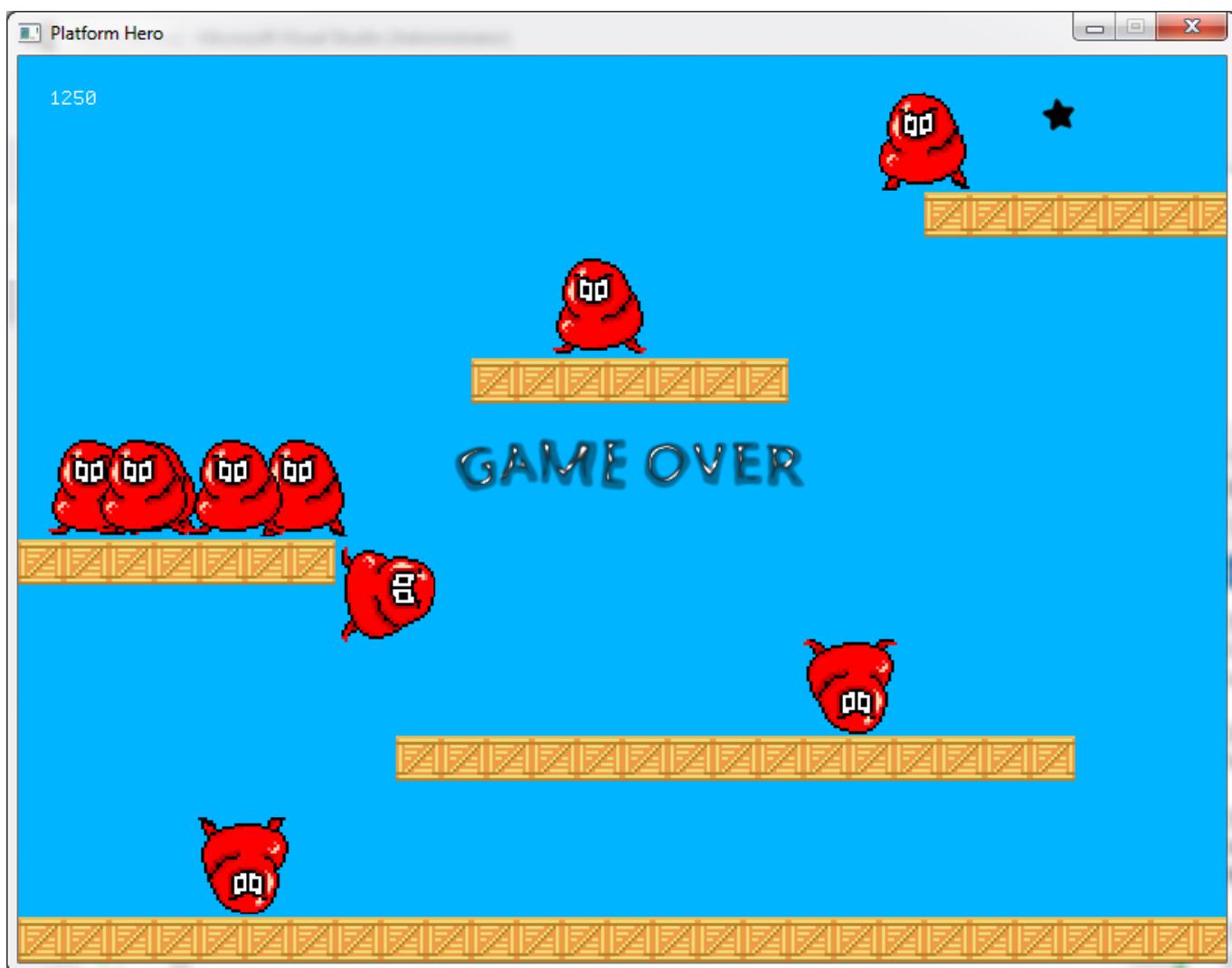
Because we never had a reference to the scene before in order to add the PopUpGameOverTrack, we'll need to make a variable reference to it at the top of the code:

```
orxOBJECT *hero;
orxOBJECT *herosGun;
orxOBJECT *scene;
orxOBJECT *scoreObject;
orxS16 score = 0;
```

And then set the scene variable on the orxObject\_CreateFromConfig("Scene") in the `init()` function:

```
scoreObject = orxObject_CreateFromConfig("ScoreObject");
orxObject_Enable(herosGun, orxFALSE);
scene = orxObject_CreateFromConfig("Scene");
```

Compile and run. Kill your hero and wait two seconds. The game over panel will appear:



Well friends, that's game over. If you reached the end here, great job! You've learned many of the major features of Orx.

If you need more help, go to the [tutorials](#) section where many of these concepts are covered in greater detail.

If you need quick examples, you can visit the [examples](#) section and search by subject.

If you get stuck, please post over on our [Discord](#). Hope to see you there.

---

- [Part 1 – Downloading Orx](#)
- [Part 2 – How Orx works](#)
- [Part 3 – Setting up a new game project](#)
- [Part 4 – A tour of an Orx project](#)
- [Part 5 – Viewport and the camera](#)
- [Part 6 – Objects](#)
- [Part 7 – Spritesheets and Animation](#)
- [Part 8 – Platforms and Texture Repeating](#)
- [Part 9 – Physics](#)
- [Part 10 – Input Controls](#)

- Part 11 – Running and Standing
- Part 12 – Changing Direction
- Part 13 – Getting our hero to shoot
- Part 14 – FX
- Part 15 – Collision Events.
- Part 16 – Jelly Monsters
- Part 17 – Timeline Tracks
- Part 18 – Exploding Monsters
- Part 19 – The Hero's survival.
- Part 20 – Text and Game Over

From:

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



Permanent link:

[https://www.orx-project.org/wiki/en/guides/beginners/text\\_and\\_game\\_over](https://www.orx-project.org/wiki/en/guides/beginners/text_and_game_over)

Last update: **2025/09/30 17:26 (3 months ago)**