

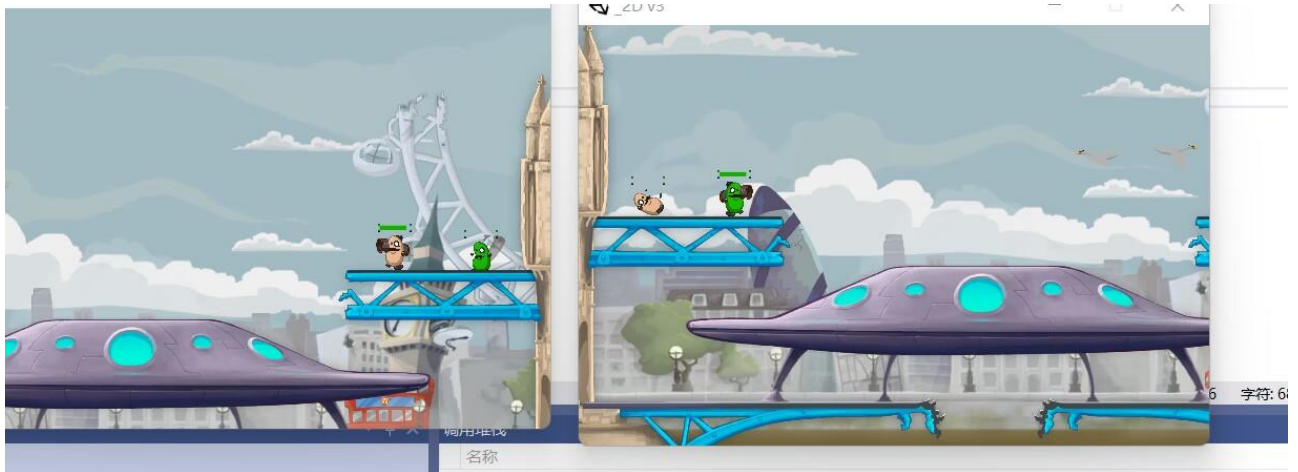
遗留 Bug 的处理

By skuuklzc

为了防止你在展示项目时，出现莫名奇妙的 bug，我们将在这篇文章解决你实验一到四留下的部分 bug 残骸。

首先第一个问题，死亡动画不同步，可以看到，我们的 hero 已经开始播放死亡动画了，但是 hero2 空血状态下还保持 idle 状态。

在上一次提交，我们已经成功解决了 hero2 死亡动画循环播放问题，原因是我们将接收 hero2 死亡的动画放在了 update 函数里，我们修改的方法也很简单，将其放在了 rocket 脚本里



```
56         else if(col.gameObject.tag == "Player")
57         {
58             if (col.gameObject.GetComponent<PlayerHealth>().health > 0)
59             {
60                 // Instantiate the explosion and destroy the rocket.
61                 col.gameObject.GetComponent<PlayerHealth>().health -= 10;
62                 col.gameObject.GetComponent<PlayerHealth>().UpdateHealthBar();
63             }
64         }
65         else
66         {
67             //sendBuff = System.Text.Encoding.Default.GetBytes("gameover " + nobj.userName);
68             //clientSocket.Send(sendBuff);
69             col.gameObject.GetComponent<Animator>().SetTrigger("Die");
70         }
71         OnExplode();
72         Destroy (gameObject);
73     }
74     else if(col.gameObject.tag == "Player2")
75     {
76         if(col.gameObject.GetComponent<PlayerHealth2>().health <= 0)
77         {
78             col.gameObject.GetComponent<Animator>().SetTrigger("Die");
79             sendBuff = System.Text.Encoding.Default.GetBytes("gameover " + nobj.userName);
80             clientSocket.Send(sendBuff);
81         }
82     }
83     OnExplode();
84     Destroy(gameObject);
85 }
```

为什么将检测角色是否死亡放在 rocket 脚本会导致死亡动画不同步。究其原因，我们可以这样理解，rocket 脚本只会在我们发射炮弹的时候调用它，当我们角色剩 10 滴血的时候，炮弹击中我们 player，我们应该走第一个 if 分支，这时候，我们的角色扣完血后，就应该死亡，但我们这是 if else 程序不会去再走一遍 else 分支，这就导致了我们的 hero2 接收到服务器传来血量为 0 播放死亡动画的时候，hero 还在等待我们下一次开火，走 else 分支。

所以我们优化也很简单

```
// Otherwise if the player manages to shoot himself...
else if(col.gameObject.tag == "Player")
{
    if (col.gameObject.GetComponent<PlayerHealth>().health > 0)
    {
        // Instantiate the explosion and destroy the rocket.
        col.gameObject.GetComponent<PlayerHealth>().health -= 10;
        col.gameObject.GetComponent<PlayerHealth>().UpdateHealthBar();
    }
    if(col.gameObject.GetComponent<PlayerHealth>().health <= 0)
    {
        //sendBuff = System.Text.Encoding.Default.GetBytes("gameover " + nobj.userName);
        //clientSocket.Send(sendBuff);
        col.gameObject.GetComponent<Animator>().SetTrigger("Die");
    }
}
```

但是这时候，我又想到在 rocket 脚本里加 hero2 死亡判定合不合理，因为我们 hero2 按理说，应该是接收服务器发来的同步血量，最后才判断是否死亡才对，如果我们在这里添加 hero2 的死亡判断，明显不合理，这就不靠服务器了，这是一个进程就解决了。

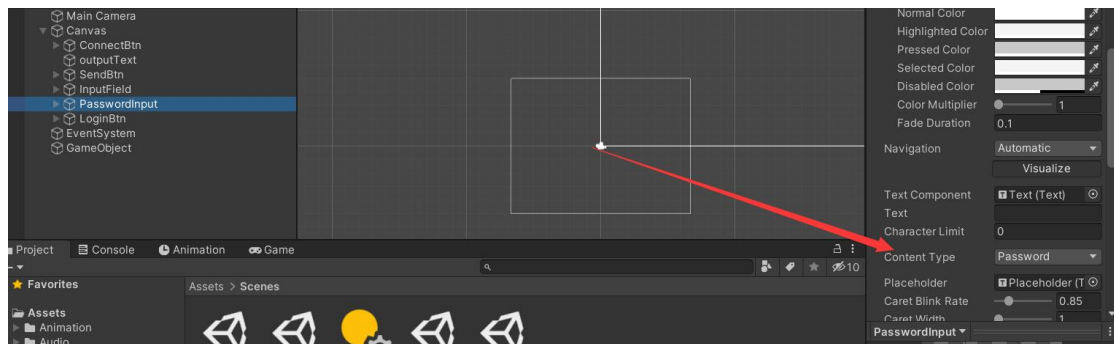
所以我们应该删除掉 rocket 脚本对 player2 的 elseif 分支内容。转而还是在我们 GameNetwork 的 update 函数里去调用。我们只需要在 update 函数里加个 tag，只允许它调用一次就可以解决循环播放动画问题。

```
85 float h2 = float.Parse(values[3]);
86 GameObject.Find("hero2").GetComponent<PlayerHealth2>().health = h2;
87 GameObject.Find("hero2").GetComponent<PlayerHealth2>().UpdateHealthBar();
88 if (h2 <= 0)
89 {
90     flag2++;
91     if (flag2 == 1)
92         hero2_Die();
93 }
```

第二个问题，是我们的注册登录密码是显示的，这看着就很 low，我们还是让他隐藏起来好点。（感谢嘉神川 Noelle 指出）



我们只需要在注册登录场景里选中我们的密码输入框，修改其内容类型就好。



代码部分修改如下：

```
outputText = GameObject.Find("Canvas/outputText").GetComponent<Text>(),  
inputPassword = GameObject.Find("Canvas/PasswordInput").GetComponent<InputField>();
```

第三个问题：当我们登录上去后一个进程后，我们的数据库表里的 online 值置为了 1，但我们后续代码没有将他置为 0，导致我们下次登录相同账户时会服务器会发出用户已登录的提示。

解决这个问题有多种方法，我随便说一种：我们在开启服务器的时候，就利用我们的 sql 语句来将数据库里的所有用户的 online 置为 0 和 userIP 置为空，虽然很反人类，但是能解决我们的问题，满足甲方爸爸的需求，缺点就是我们退出游戏后，还是能在数据库里看到我们的 online 和 userIP 依旧存在。

```
1. try  
{  
    listSock.Bind(ipe);  
    listSock.Listen(10);  
    listSock.BeginAccept(new AsyncCallback(acceptCb), listSock);  
    outputText.Text = "开始监听10001号端口";  
    button1.Enabled = false;  
  
    SqlConnection conn = new SqlConnection(connStr);  
    conn.Open();  
    String sqlStr = "UPDATE table5 SET online=0, userIP=null";  
    SqlCommand comm = new SqlCommand(sqlStr, conn);  
    comm.ExecuteNonQuery();  
    comm.Dispose();  
}
```

第四个问题：如何计算得分？

当我们 hero 将 hero2 打死后，我们希望在数据库里看到这个 user 的 score 分数能够加 1。解决这个问题，我们需要编写一个协议，比如我编写的协议就是：gameover，当我们 user1 战胜 user2 后，我们将 user1 的 username 作为协议内容传递给服务器，服务器接收到我们发送的 username 后，会调用一个计算得分的函数，关于这个函数的编写也很简单，我就只提 sql 语句该怎么写：

UPDATE 表名 SET SCORE=SCORE+1 WHERE USERNAME=传入的 username

至于 SET 部分根据个人的需求想怎么加就怎么加，甚至给另一个玩家扣分都行。多提一嘴，在哪里获取 username 作为协议参数传给服务器呢，我们要想到 username 的起源在登录注册那里，我们只需在那里定义一个 public 参数，将 username 保存下来，在其他场景调用这个 public 参数就好。

第五个问题：如何让客户端与服务器断开连接，同样当我们游戏结束后，我们不希望客户端还一直向服务器发送心跳包，这时候服务器与客户端应该断开连接，游戏结束。

这个问题我自己都有点不太搞清楚(欢迎指正和交流)，断开连接需要利用 Socket 的 shutdown 函数和 close 函数，但是在 unityEditor 里面会报错。不过好像还是能凑合解决这个问题哦。

当我们角色死亡时，我们就应该关闭连接，在角色动画播放后，服务器计算完得分后，客户端就可以与服务器断开连接了。

我写的(模仿的服务器代码凑的)伪代码大概就是：

```
if(Socket.Connected&&Socket!=null)//判断是否连接上了
{
    Socket.Shutdown(SocketShutdown.Both);
    lock(Socket)
        Socket.Close();
}
```