

Software engineering practices

HW3 : Release bugs (core dump analysis)

學號： 112526011

姓名： 林睿瀚

A. Screenshots of the dump analysis

- ✓ DetectThread(圖 2 為開啟外部程式碼之 call stack)

呼叫堆疊		語言
名稱		
[外部程式碼]		
KeyboardWarIII.exe!KeyboardWar.KeyboardWar.DetectOutScreen() 第 57 行		C#
KeyboardWarIII.exe!KeyboardWar.KeyboardWarForm.CreateDetectThread.AnonymousMethod_7_00 第 60 行		C#
[外部程式碼]		

呼叫堆疊		語言
名稱		
mscorlib.dll!System.ThrowHelper.ThrowInvalidOperationException(System.ExceptionResource resource)		不明
mscorlib.dll!System.Collections.Generic.List<System.__Canon>.Enumerator.MoveNextRare()		不明
mscorlib.dll!System.Collections.Generic.List<KeyboardWar.Monster>.Enumerator.MoveNext()		不明
KeyboardWarIII.exe!KeyboardWar.KeyboardWar.DetectOutScreen() 第 56 行		C#
KeyboardWarIII.exe!KeyboardWar.KeyboardWarForm.CreateDetectThread.AnonymousMethod_7_00 第 66 行		C#
mscorlib.dll!System.Threading.ThreadHelper.ThreadStart_Context(object state)		不明
mscorlib.dll!System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, ...)		不明
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object s...)		不明
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object s...)		不明
mscorlib.dll!System.Threading.ThreadHelper.ThreadStart()		不明

- ✓ CreateMonsterThread(圖 2 為開啟外部程式碼之 call stack)

呼叫堆疊		語言
名稱		
[外部程式碼]		
KeyboardWarIII.exe!KeyboardWar.KeyboardWarForm.CreateMonsterThread.AnonymousMethod_6_00 第 47 行		C#
[外部程式碼]		

呼叫堆疊		語言
名稱		
mscorlib.dll!System.Threading.Thread.Sleep(int millisecondsTimeout)		不明
KeyboardWarIII.exe!KeyboardWar.KeyboardWarForm.CreateMonsterThread.AnonymousMethod_6_00 第 51 行		C#
mscorlib.dll!System.Threading.ThreadHelper.ThreadStart_Context(object state)		不明
mscorlib.dll!System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, ...)		不明
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object s...)		不明
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object s...)		不明
mscorlib.dll!System.Threading.ThreadHelper.ThreadStart()		不明

- ✓ UpdateThread(圖 2 為開啟外部程式碼之 call stack)

呼叫堆疊		語言
名稱		
[外部程式碼]		
KeyboardWarIII.exe!KeyboardWar.KeyboardWarForm.Update() 第 90 行		C#
[外部程式碼]		

呼叫堆疊		語言
名稱		
mscorlib.dll!System.Threading.Thread.Sleep(int millisecondsTimeout)		不明
KeyboardWarIII.exe!KeyboardWar.KeyboardWarForm.Update() 第 90 行		C#
mscorlib.dll!System.Threading.ThreadHelper.ThreadStart_Context(object state)		不明
mscorlib.dll!System.Threading.ExecutionContext.RunInternal(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, ...)		不明
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object s...)		不明
mscorlib.dll!System.Threading.ExecutionContext.Run(System.Threading.ExecutionContext executionContext, System.Threading.ContextCallback callback, object s...)		不明
mscorlib.dll!System.Threading.ThreadHelper.ThreadStart()		不明

B. Possible causes of the bug

因為此應用程式為多執行序程式，而透過反組譯執行檔，透過 call stack 可以發現錯誤是報在：

```
public void DetectOutScreen()
{
    List<Monster> deleteMonsterList = new List<Monster>();
    foreach (Monster j in Monsters)
    {
        if (j.Y + j.Size.Height > (float>windowHeight)
        {
            j.Die();
            score += minusScore;
            MinusScore(j);
            deleteMonsterList.Add(j);
        }
    }
    foreach (Monster i in deleteMonsterList)
    {
        DestroyMonster(i);
    }
}
```

而透過此反組譯原始碼可以發現，會使用 foreach iterator 去遍歷並修改 .NET 中的集合結構(List<Monster> Monsters => monsters;)，而因為在遍歷 Monsters 時，.NET 會檢查該集合當下的版本 (version) 是否和建立 enumerator 時一致，如果在這個遍歷過程中（尚未跑完整個遍歷過程），有任何其他執行序對同一個 Monsters 集合做了 Add、Remove、Clear 等動作，就會導致版本號改變，進而拋出 Collection was modified; enumeration operation may not execute. 之 InvalidOperationException，而如果此應用程式未對此例外做處理，就很有可能造成整個應用程式 crash。

而解決此問題的方式可以透過 lock 等同步機制，去確保在同一時間只有一個執行序可以存取或修改集合，避免版本號在遍歷過程中被其他執行序改變，進而去避免引發此例外。