Lab 7 – Debugging

Fork and clone the projects for Git Hub <https://github.com/Abertay-University-SDI/CMP105_W7> . This repository contains a collection of small programs that suffer from some compilation or runtime errors and need fixing.

1. Starting with application Week 7\_1, the application contains a player object that teleports to a random location when the *space bar* is pressed, and an enemy object that teleports to a random location after a set amount of time. However, the application does not compile successfully. You must find and fix the error, in the box below write down what the error was and how you found it (what debugging steps did you take, what information was useful).

|  |
| --- |
| There were a few errors with this program. Firstly, Player.h included “Enemy.h” when there was no need to at all. Secondly, in player.h, there was no override specified on the handleInput function which was causing problems. This was an easy fix by adding override to the end of the prototype in the header file. As it inherits from GameObject, it needs the override to be specified. Enemy.h didn’t have a semicolon after the final bracket which I rectified. This is the syntax needed for a class and thus it was causing issues when it wasn’t there. Lastly, enemy.h was also missing an override keyword but this time on the Update function. I fixed this the same way as before by adding override at the end of the prototype. It was useful knowing what the program was supposed to do as I could then test to see if it was working. I used F5 to build and see where the errors came up. I then read through the header and cpp files to see if I could spot anything further. |

1. The application Week 7\_2, this application contains a player object that teleports to a random location when the *spacebar* is pressed, and the player contains a bullet object that spawns at the player’s location and fires when the *Enter* key is pressed. However, the application does not compile successfully. You must find and fix the error, in the box below write down what the error was and how you found it (what debugging steps did you take, what information was useful).

|  |
| --- |
| I fixed a few errors within this program. Bullet.h was missing the override keyword on the update function. I sorted this by adding said keyword. Player.h was missing a colon after the private visibility label. This was an easy fix by adding the missing colon. I also called the bullet’s update function in level.cpp. This made the actual firing functionality work. Lastly, I changed the bullet’s position, so it got set to the player’s position rather than a random one. I found the first two errors by using F5 to find the compile errors. The second 2 I found out just by reading through the cpp files and seeing how the actual functionality compared to the intended functionality. |

1. Application Week 7\_3 is similar to 7\_2, however this application compiles, but suffers from a runtime error. You must find and fix the error, the fixed application should have a player the teleports and shoots several bullets. In the box below write down what the error was, how you found it (what debugging steps did you take, what information was useful) and how you would fix it.

|  |
| --- |
| Problem was that the Bullet was a null pointer. Fixed this by giving it a bullet to point to. |

1. Application Week 7\_4 contains a player and bullet objects. This application compiles but suffers from a runtime error. You must find and fix the error, in the box below write down what the error was, how you found it (what debugging steps did you take, what information was useful) and how you would fix it.

|  |
| --- |
| The program was originally throwing up an error to do with access violation at first. The steps I took to fix this included giving the Bullet pointer a bullet object to point to. I then altered the getBullet function to return a bullet rather than a pointer to one. I would set this equal to the bullet object and then ensure that the pointer was pointing towards it also. I was instantly able to narrow down the problem to the pointer by pressing F5 as it was always referring to the bullet having issues. I also added a reload function so there is a way of resetting the hasFired Boolean. |

1. Application Week 7\_5 contains a player object and a companion object. The player object teleports around the level when *spacebar* is pressed. The companion will teleport to the player object when the *Enter* key is pressed. This application has a compilation error. You must find and fix the error (and make sure the application runs correctly), in the box below write down what the error was, how you found it (what debugging steps did you take, what information was useful) and how you would fix it.

|  |
| --- |
| The error in this program was that Player was including Companion.h and Companion was including player.h. Companion only needed access to the position of the player so I removed the include line and altered the moveToPlayer function so that it only needs a Vector passing through. I could have also done this a different way and just created a reference to the player in the companion class but I figured this wasn’t necessary in this case. I found the error after seeing that all of the errors stemmed from the Companion variable in the player class. Nothing was syntactically incorrect so clearly something else was at play. I then remembered a previous issue I had where I had classes that included each other and that’s how I figured out the issue. |

1. Application Week 7\_6 contains several *Bits*; small sprites that are given a random position and move up and down from the position. This application compiles but suffers from a runtime error. You must find and fix the error, in the box below write down what the error was, how you found it (what debugging steps did you take, what information was useful) and how you would fix it.

|  |
| --- |
| The error for this was quite simple to find. Pressing F5 led me to the for statements and I was getting the access violation error again. I looked closely and saw that the for statement said <= 6 which means that it is trying to access bits[6] which doesn’t exist as the elements go from 0 to 5. Further down, I noticed a <= sizeOf(bits)/sizeOf(bits[0]) which was exactly tha same as writing <= 6 as that calculation finds the size of an array. This was an easy fix by simply removing the = sign from both for statements. |

1. Application Week 7\_7 should render a simple yellow game object near the centre of the window, but it doesn’t. You must find and fix the error, in the box below write down what the error was, how you found it (what debugging steps did you take, what information was useful) and how you would fix it.

|  |
| --- |
| I found this error pretty quickly. The size, position and colour were all stated so I knew that wasn’t the problem. What happened was, a local go variable was created within the constructor and this is what was altered when we were defining size, position and colour. Meanwhile, the public variable go was untouched as the changes were made to the local. The draw function only has access to this untouched variable so it was drawing an empty gameObject. This was easily fixed by deleting the line that created the local variable as it wasn’t needed. |

Make sure you show/discuss your answers with a member of staff once completed.

1. The coursework brief is on My Learning Space, under the Assessment tile. Read over the brief and write down a rough outline/design for you coursework game idea. Think carefully and write down how you game meets each of the coursework requirements.