**Ulster University, Faculty of Computing and Engineering**

COM377 – Programming lll

Semester 2 – Coursework 2

Dr. H. Wang

Due by – Monday 18th April 2016

**Pacman Game - Application Report**

**Contents:**

* Report Introduction Page: 2
* Pacman Game Functionality Page: 3
* Demonstration of Testing Page: 5
* Implementation of Advanced Features Page: 8
* Using Microsoft Team Service Page Page: 9
* Contribution of each Team Member Page: 11

**Group: CW2 – 10:**

* Jonathan McCrink (B00656761)
* Gary McShane (B00656764)
* Aine McCaul (B00686604)

***Report Introduction:***

For this assignment our group was required to create the popular game Pacman using C#. The application must allow a user to play the popular game by moving the Pacman while trying to avoid the moving ghosts around the map. The user will begin with 3 lives and must make their way around the map before losing all their lives. The application must also allow the user to play / pause or exit the game when they wish as well as other features such as viewing their score / remaining lives etc. The game also plays the familiar sounds from Pacman such as the starting tune, the tune played as the Pacman navigates through the map, and the tune played once the Pacman dies.

An additional feature that we added to the game was for the user to be able to view the game rules. We added this into the menu bar so when the user clicks the option ‘Game Rules’ then a form will appear on screen listing a few rules for the game. This can easily be dismissed by clicking a close button.

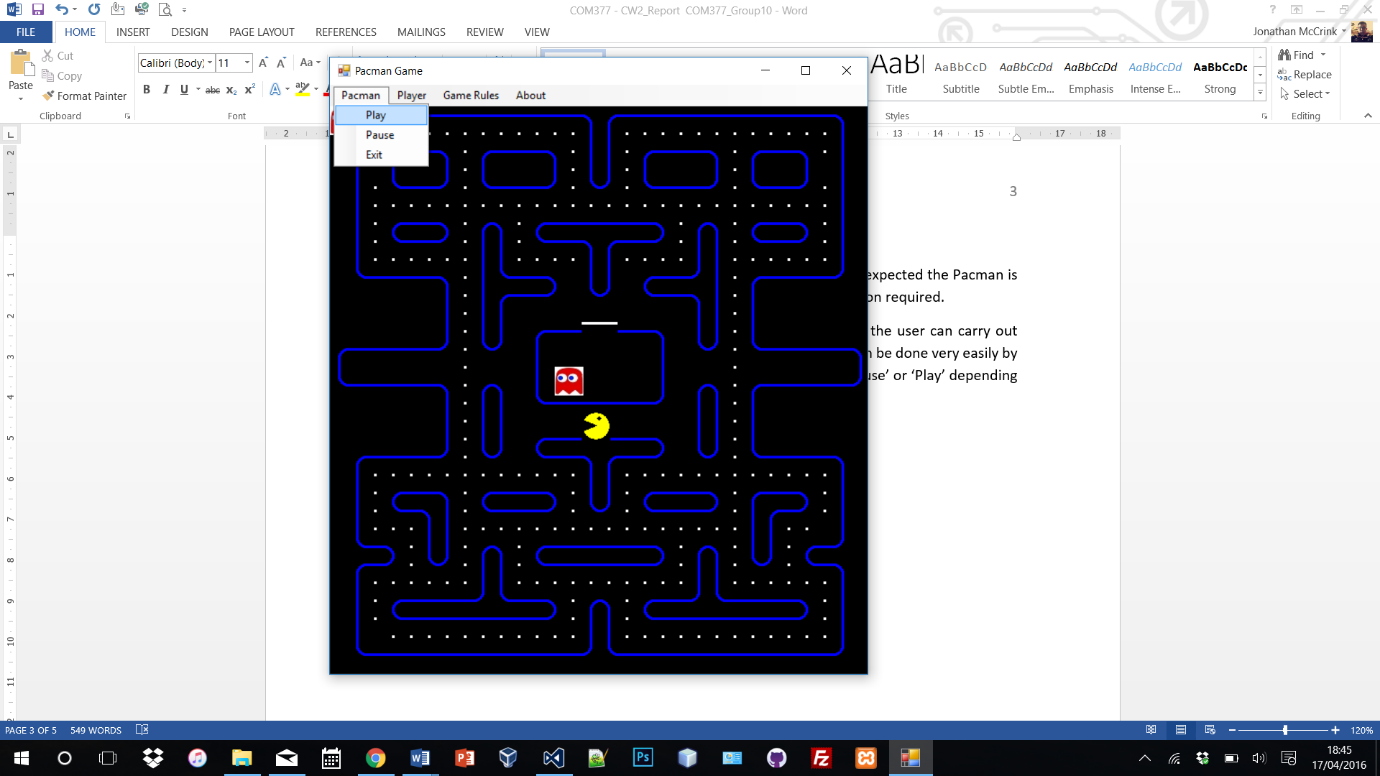
This report will bring you through in depth the development of this game. We will first explain our interface and the features it includes. We will explain how the game can be paused / resumed, how the number of lives are updated for the user. We will also explain how the ghosts move in our game as well as how the Pacman moves including animation of it.

We will also show testing of the game using a test plan as you will see below. We used Microsoft’s Team Service during the development of this game to aid in interaction between team members, planning contribution and also version control which was key for a group with a number of members. We will explain in this report how we used this and how it helped us in the development of the game. Finally we will explain the contribution of each team member to the code development of this game.

***Pacman Game Functionality:***

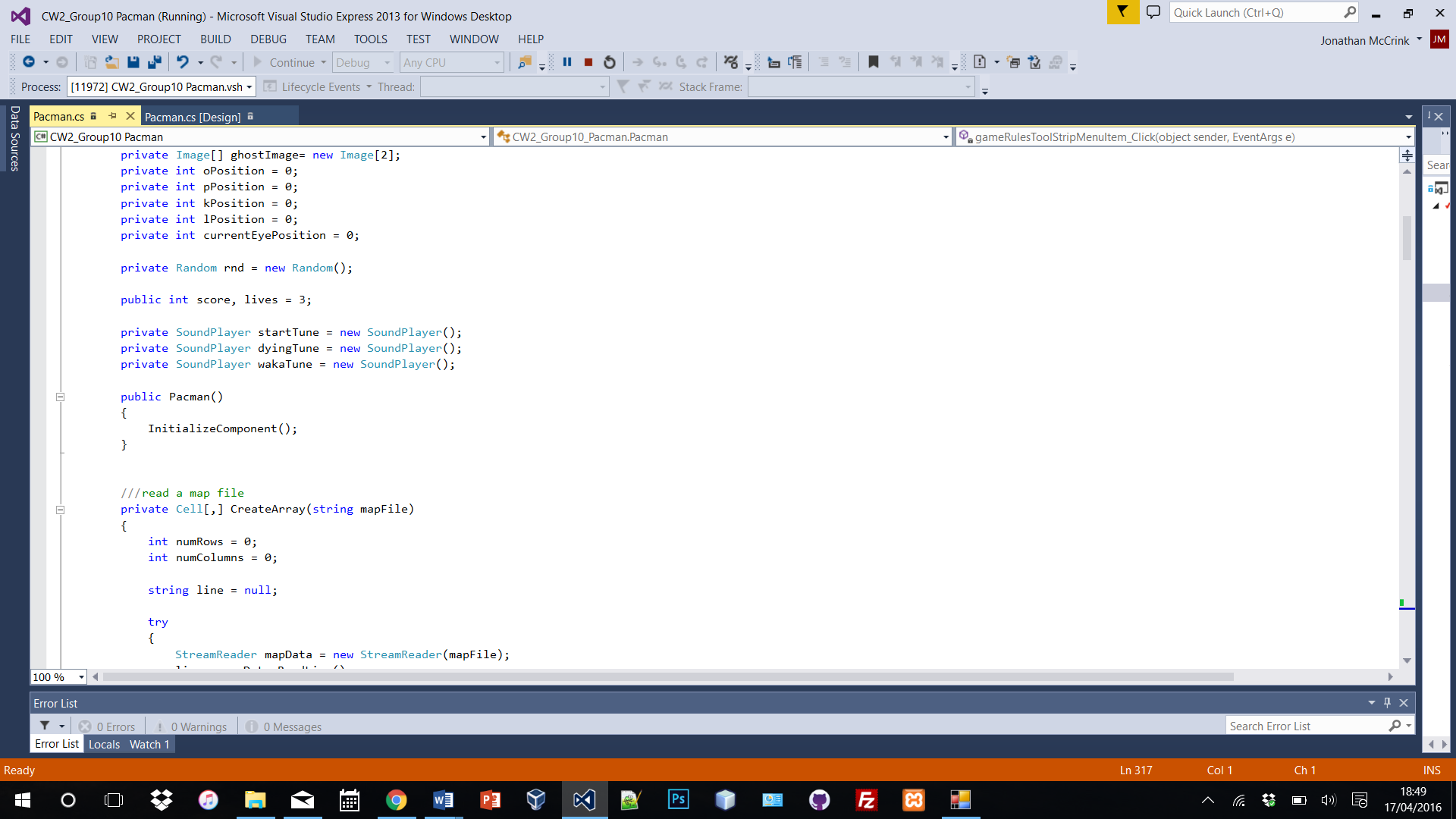
We made sure to keep the controls of the game very simple for the user. As expected the Pacman is controlled by the arrow keys and to control the game this is the only interaction required.

We do however have a menu bar placed at the top of the interface where the user can carry out different events. One of these being able to pause / resume the game. This can be done very easily by the user simply clicking onto the ‘Pacman’ menu option and then clicking ‘Pause’ or ‘Play’ depending on whichever they need. See the screen shot below of this in action.

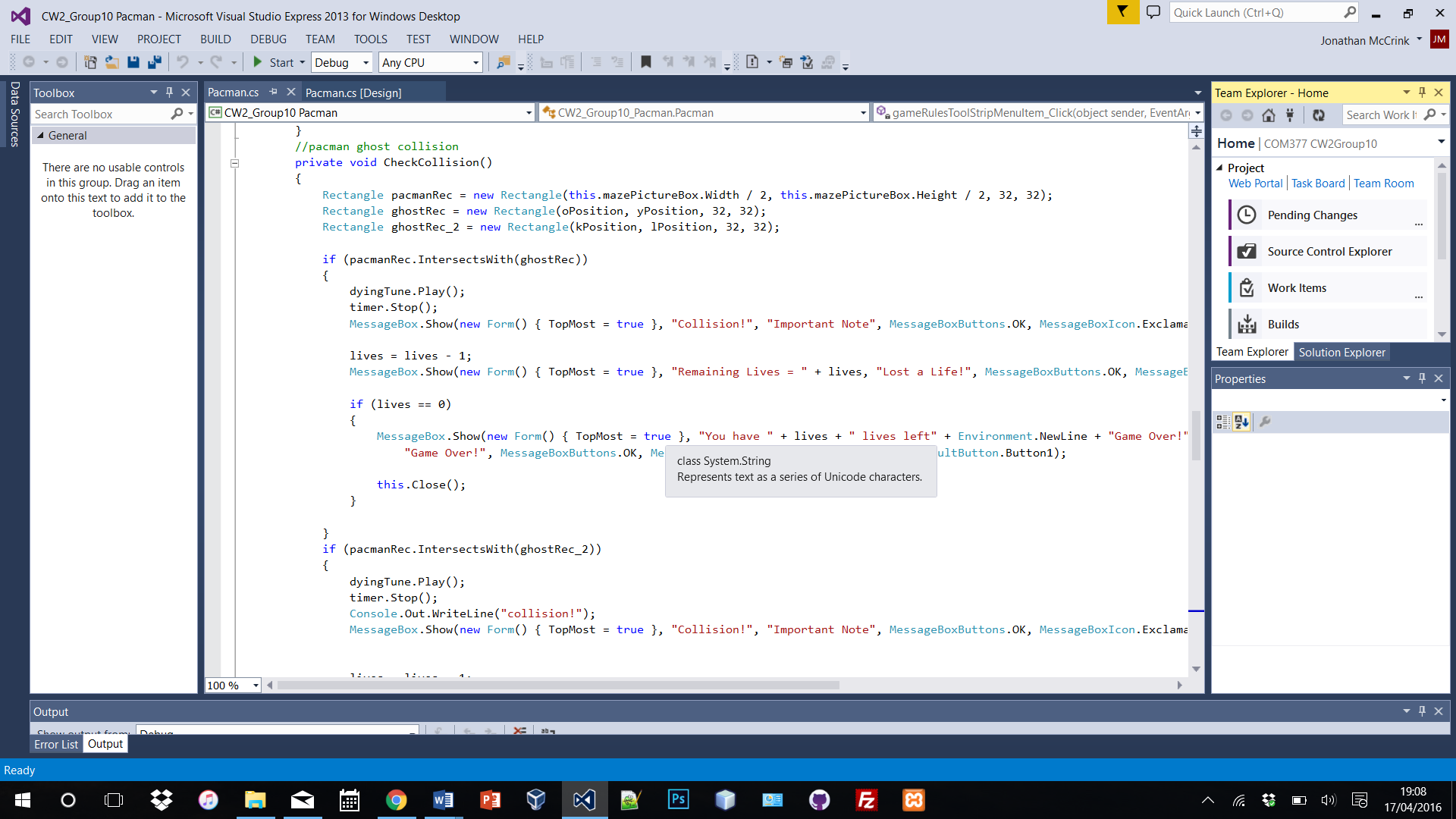


The user can also exit the game from this option too if they so wish. Once the pause option is clicked then this will trigger the ghosts to stop as well as the Pacman stopping. Any sound that is playing in the background will also be stopped once Pause is clicked. Once Play is clicked then all these features mentioned will again resume and the user can continue playing the game.

As mentioned before and as we know from the classic game, the user will also start off with 3 lives for the Pacman. We made sure to implement this into the game by also initially having a lives variable in the game to be assigned to three. The piece of code below will show this:

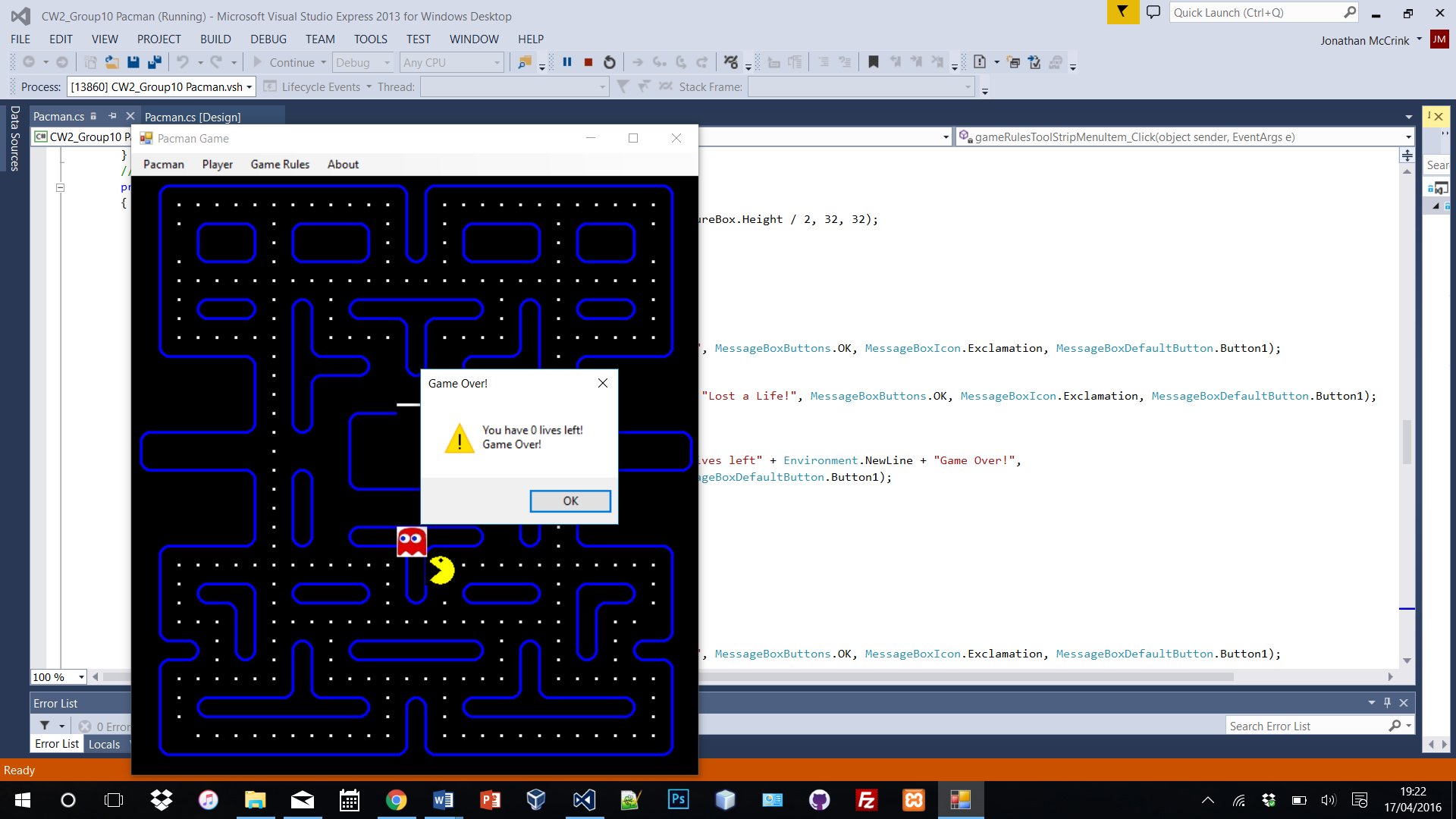


Each time the Pacman collides with a ghost the application will output a message explaining that a collide has happened however the Lives varialble will also be updated as 1 is subtrated to this. Another output messgae will appear then showing the remaining lives left. Once the value reaches 0 then a message also appears to show that the game is over and the applcation will then end.

At any time the user also wishes to view their remianing lives they can do so by selecting the Lives option from the ‘Player’ menu item. The code below will show how the lives variable is updated by subtrating 1 from the current lives.

The if statement above is triggered if a collision happenes in the game. As you can see, highlighted in red above, that the lives variable has one taken off it and an output message is displayed explaining this. There is also an if statement within this to check that if the lives reached 0 then an output message will also be displayed here explaining this and the application will then end.

See a screenshot below of this happening:



One important aspect of the game is to ensure that there the ghosts do catch the Pacman in order for lives to be lost. It was quite difficult to be able to create a ghost that follows the Pacman however we implemented it in a way that it will always be a few steps behind the Pacman and follow it in one direction, however if the Pacman turns then it will collide with that ghost.

We added another ghost into the game and this one would travel around the map randomly. We first created a variable that would hold a randomly generated number and then assigned this to the positioning of the ghost. This ghost does not travel in the way we want or need it to however after weeks of trying to get this correct we could not fix it properly. It goes travel to random spots on the map and eventually does collide with the Pacman which is the main idea for this ghost. In order to start this random ghost the user needs to press the ‘R’ key and to stop they press ‘P’.

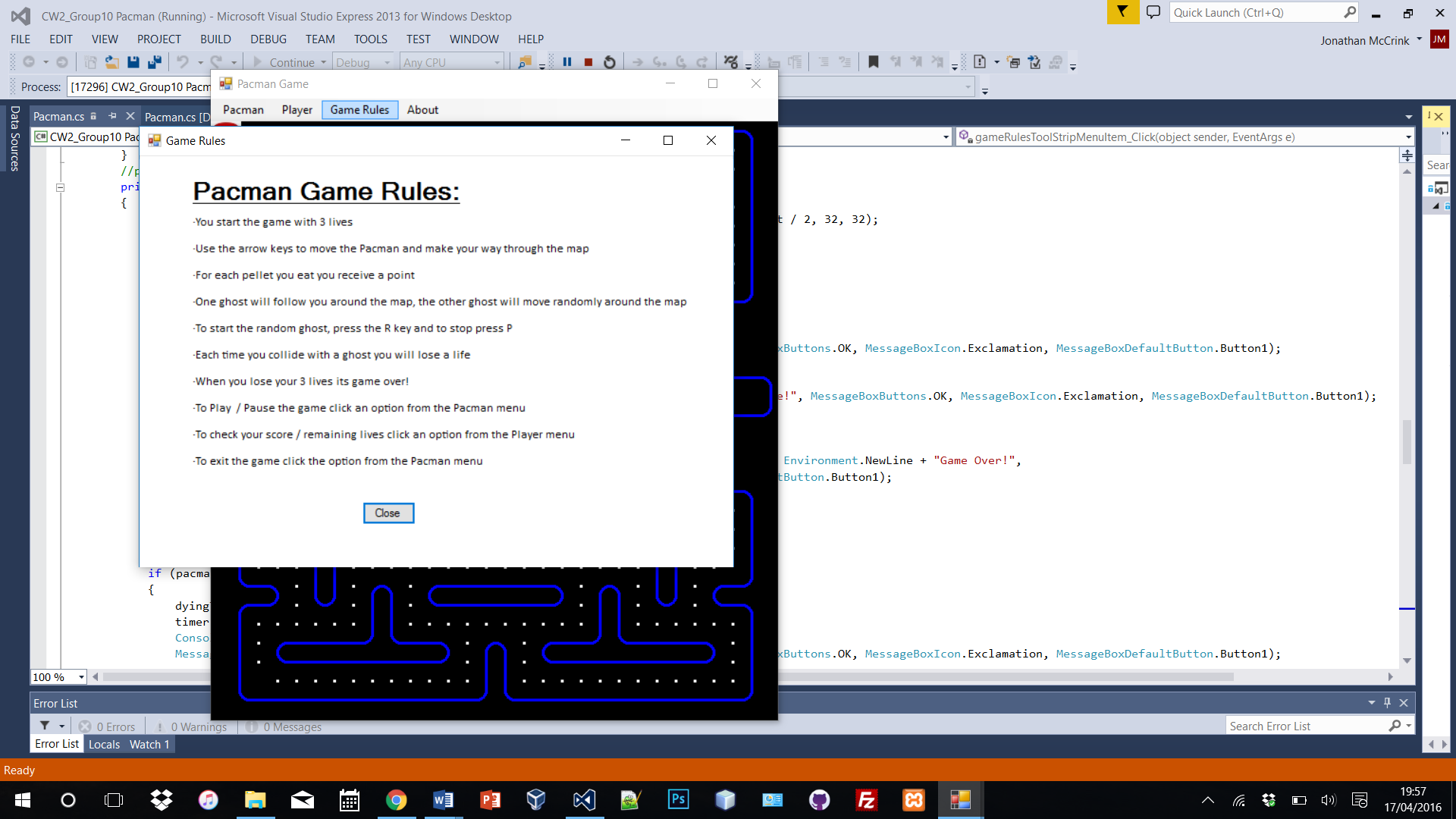
***Demonstration of Testing:***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Purpose Of Test | Test Num | **Expected Output** | **Actual Output** | **Screen shot** |
| Test to see what happens when the user uses the arrows to move the pacman |  | The pacman and one of the ghost move in the same direction | As excepted |  |
| Test to see what happens when the user presses the r key |  | The second ghost move around the screen randomly. | As excepted |  |
| Test to see what happens when a collision happen between the ghost and the pacman |  | A message box appears to information the user the pacman and ghost have collided and informs the user of the amount of lives left | As excepted |  |
| Test to see what happens when the user selects the pause option in the Menu tab |  | A message appears informing the user the game has been pause and the music stops | As excepted |  |
| Test to see what happens when the user selects the Play option |  | The user is able to continue to play the game | As excepted |  |
| Test to see what happens when the user select the ‘About’ tab |  | The ‘About’ tab will bring the user to information at the group that created the application. | As excepted |  |
| Test to see what happens when the user clicks the ‘Game Rules’ tab |  | A game guides form will appear | As excepted |  |
| Test to see what happens when the user clicks the ‘Lives’ option with the player tab. |  | A message box informing the user the amount of lives remaining |  |  |
| Test to see what happens when the user clicks on the ‘Exit’ button |  | The application should close | As excepted | N/A |

***Implementation of Advanced Features:***

For the Pacman game we decided to implement an advanced feature. This being a ‘Game Rules’ tab which will launch a form listing the rules for the Pacman game. We understand that this is a simply feature however one that is keen to the user and important. We respect some are novice users to the Pacman game and may not have played it before plus some ways to interact with our version of the game may be different to those on the classic Pacman game, therefore the use of a Rules form is very effective.

The form is linked to a ‘Game Rules’ tab on the main games form and once clicked will launch this. The Rules form has just one button which allows the user to close it once finished. This button will not close the main application however therefore can be opened and closed as much as the user wishes.



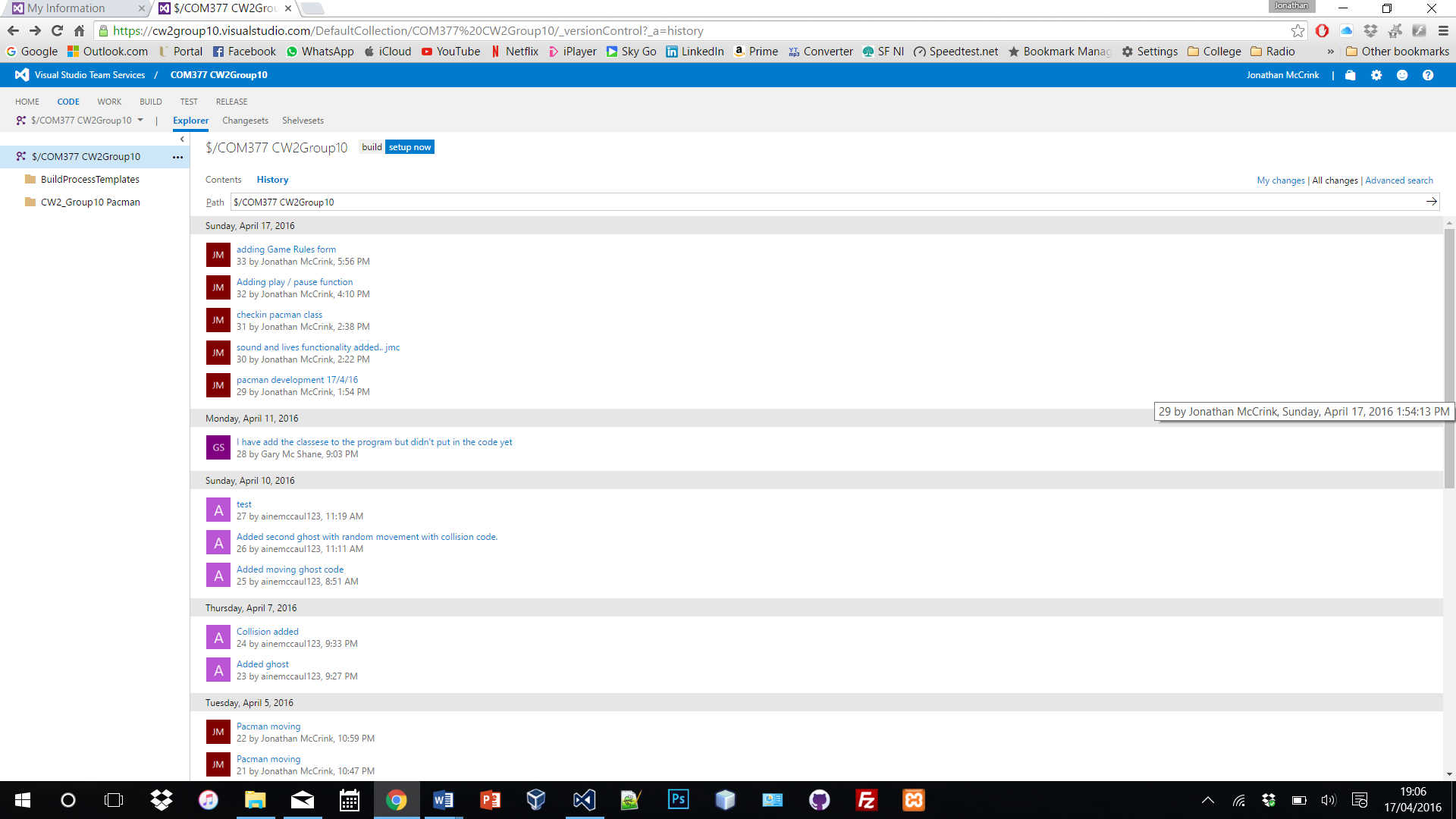
The screenshot above shows this Rules form in operation. In the background you will see that the user just needs to click the ‘Game Rules’ tab to launch this. The form is quite informative and outlines all rules and ways to use the Pacman Game. This is easily closed by using the ‘Close’ button.

***Using Microsoft’s Team Service:***

During the development of the assignment we used the Microsoft Visual Studio Team Service a lot to help us in planning, development, interaction and version control.

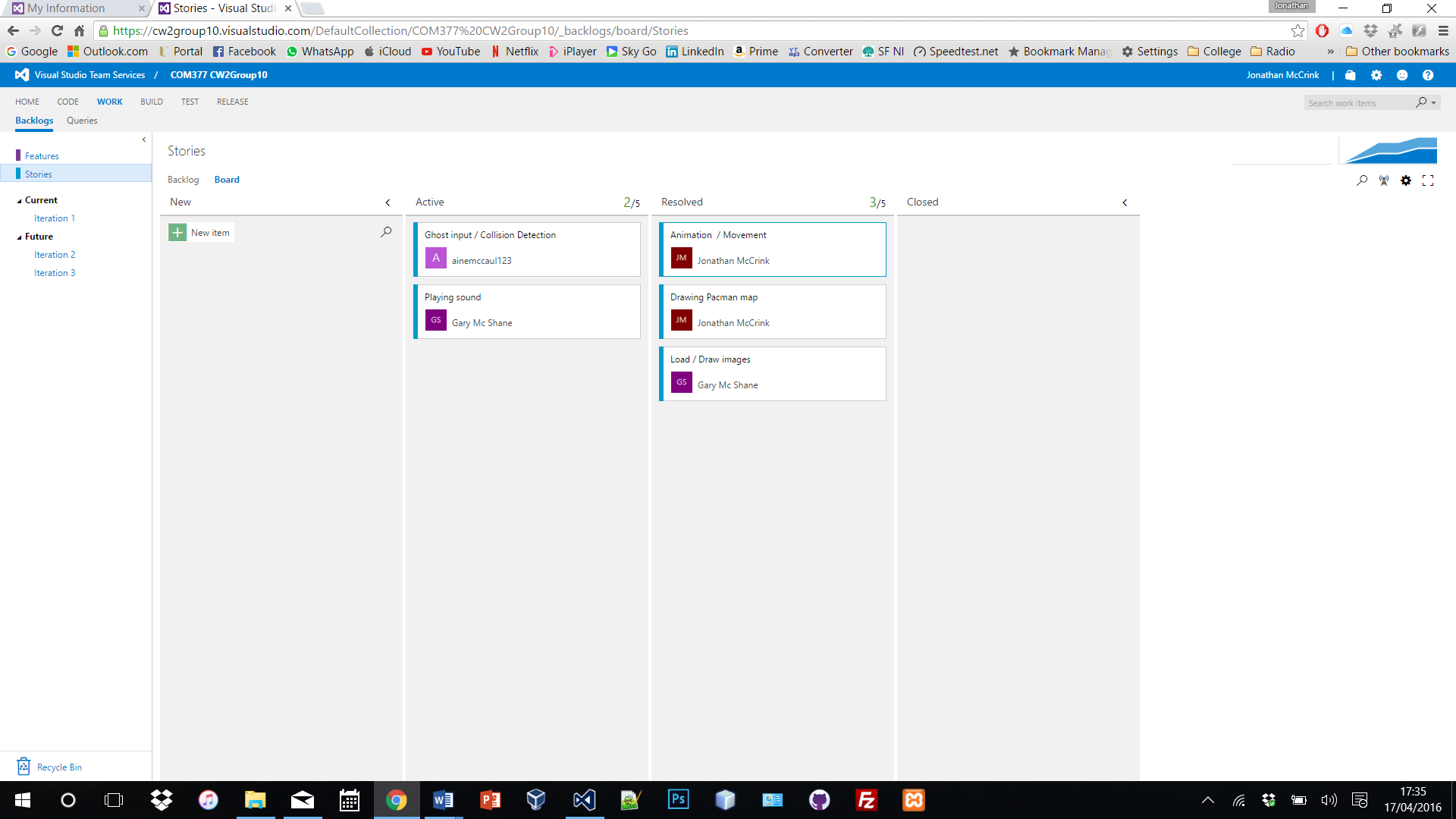
As we had a group with several members it was very important to be careful with version control during the development of the program. We wanted to ensure that when one person was working on the code that another could not work on that piece which may have resulted in overwriting that file and potentially losing a lot of good work. Therefore we resulted to using Team Service which help a version of our code which we were able to check out files from, edit in Visual Studio and then again check in once completed. This way work would not be replaced or lost. This is also good practice when working in a large team.

See below a screen shot of our groups check in history throughout the development of the assignment. You will see that each member has made a number of check ins with code that has been uploaded to the site. This way also each member will always have the most updated version of the assignment at all times without asking to send via emails etc.

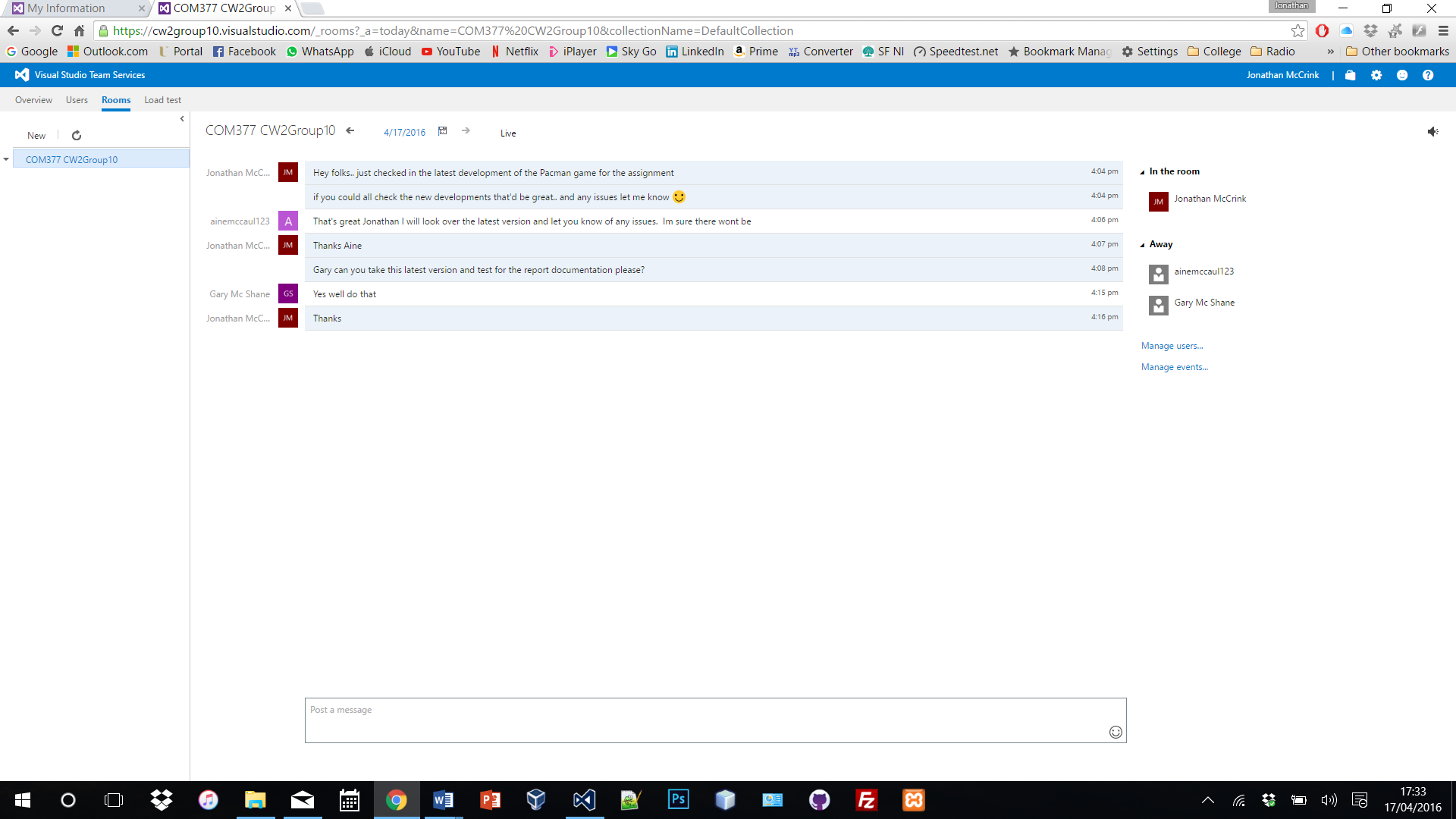


Team Service was also very helpful at the beginning and throughout the planning stages of our assignment. As it uses the Sprint agile methodology, we were able to plan work items for each stage of the development of the assignment and assign these to group members. This was very useful as it allowed us to create a Kan Ban board and place work items on this assigned to each member. We were then able to move these work items and see their life cycle from the beginning where they were created, to Active, Resolved and closed.

Our team decided to meet and discuss each work item and then as group leader, Jonathan was then able to go off and record these on Team Service and assign each to the group member that the group previously agreed to. The team found this very helpful. See below a screen shot of a board with work items placed on it. You will also see that each of these is assigned to a member of our group and are all at different stages. This board was from a stage when we were developing the game.



Another feature within Team Service which we found was very necessary during the development of the game, was the ability to be able to interact with each other via a ‘Room’. Team Service has a feature which allows a team to create a group team chat where they can then speak freely to each other within the app about the development of the site. We used this feature as well as setting up a group chat via Facebook to always interact with each other, inform of progress as well as bringing up any issues that we could help each other with. Without being able to interact with each other in this way would prove to be very difficult for us all, as talking to each other and discussing the development was key! See a screen shot below of the three group members chatting to each other using the Team Service Room feature.



***Contribution of each team member:***

You will see in each group member’s individual assessment that we all agreed that we contributed equally to the assignment with each having a percentage of 33.3% totalling to the 100% contribution.

See below the contribution of each team member.

Jonathan started the development of the assignment by creating the maze map for the Pacman game. Once he completed this he then carried on to load images into the application for the Pacman. From the lectures and practical labs in class Jonathan was then able to create animation of the Pacman to create movement of the mouth opening and closing. This was down by loading in three/four images and creating animation from this.

After this Jonathan then worked on the movement of the Pacman using the arrow keys on the keyboard. This was good progress for the team because of Jonathan’s input.

Jonathan also added sound into the program which includes the starting tune, the waka movement tune and finally the dying tune.

Other tasks that Jonathan carried out was initially setting up the Team Service account for the team and adding both Aine and Gary to this. The report was also developed by Jonathan with the help of Gary who developed our test plan.

Aine’s contribution to the development of the program included adding in the ghosts to the game and getting these moving.

After Jonathan was complete with his development of the Pacman, then Aine took over and added in the ghosts. This included altering the code Jonathan developed and ensuring that the ghost was a few steps away from the Pacman.

Aine also developed code for a second ghost which this time would travel randomly around the map. This did prove difficult for her and the team members did help, however with a lot of time we decided as a group to allow this ghost to travel this way. Unfortunately it’s not the way we needed it to go however we could not get this properly corrected.

Gary’s contribution to the code was in the development of our menu bar. This menu bar has a lot of functionality as it controls the play / pause of the game as well as being able to exit it.

With the help of Jonathan, Gary was also able to implement the remaining lives output into the menu bar and develop a score.

Gary and Jonathan also developed an advanced feature into the code with the use of a Rules page. Jonathan created the form and added the rules into it and once this was completed, Gary was then able to link this to the main form by adding a ‘Game Rules’ tab onto the menu bar.

Other tasks that Gary carried out included the testing of the game and then developing a test plan with the tests and adding this to this report.

To conclude, we as a team, believe that each member has given an equal contribution to the development of this assignment. We do believe that a lot more could have been improved on the final outcome of the Pacman game if we’d more time, resources and of course knowledge however given what we have accomplished as a team, we are happy with out outcome.