Professional Practise in IT – Project – 2020

*Shaun Daly & Brian Coppinger*

Unity 2D platformer made using C#

1. Project requirements
2. Technologies used
3. Design Methodologies Implemented
4. Architecture of the solution
5. Test plans
6. Conclusion

Project requirements

Our aim was to design a simple 2D platform game in Unity. We chose Unity as it is a very powerful and yet fun tool to work with.

We decided to create a game with multiple levels that follow a theme in design and playability that slightly increases in difficulty level by level.

We had many different design ideas but, in the end, came together and decided to use a bright friendly theme.

The playable character is a designed to be funny to look at while still being able to achieve his goal.

We demonstrated many skills while working on this project. We had to communicate remotely using various methods to make a plan and then follow the steps set out, we believe we achieved this quite well and got an end product with were both satisfied with.

We also had to use our skills on programming to make all the features of the game work effectively in a bug free manner. This brought skills from many of our modules together (Mobile app development, Professional practise) including the final testing of the project (Software testing).

Technologies used

We used Unity as the core program for our project, along with Visual studio for the coding aspect. Unity is a very powerful tool that has many inbuilt feature with works extremely well along side visual studio for creating and editing scripts.

Unity comes with an asset store with lots of useful and fun asset to work with, but you can also design your own using third party editing software such as Gimp etc.

We used Github for version control and as a way to share work with each other remotely.

We also communicated using Microsoft Teams, and other messaging services such as WhatsApp.

We both used different versions of Unity which brought up some additional issues that we were able to work around in the end.

Design Methodologies Implemented

We primarily used an Agile development methodology in our project. We found that little increments with regular testing would be the best way to avoid a catastrophic bug or commit. Given that we were both coding in separate locations and with different styles this worked well for us.

One con of this method would be an increase in time taken as we had to regularly test features. However, in the long run we both believe this saved a lot of time.

*Four values of Agile*

*The four core values of*[*Agile software development*](https://searchsoftwarequality.techtarget.com/definition/agile-software-development)*as stated by the Agile Manifesto are:*

1. individuals and interactions over processes and tools;
2. working software over comprehensive documentation;
3. customer collaboration over contract negotiation; and
4. responding to change over following a plan.

*The 12 principles*

The 12 principles articulated in the Agile Manifesto are:

1. Satisfying customers through early and [continuous delivery](https://searchitoperations.techtarget.com/definition/continuous-delivery-CD) of valuable work.
2. Breaking big work down into smaller tasks that can be completed quickly.
3. Recognizing that the best work emerges from self-organized teams.
4. Providing motivated individuals with the environment and support they need and trusting them to get the job done.
5. Creating processes that promote sustainable efforts.
6. Maintaining a constant pace for completed work.
7. Welcoming changing requirements, even late in a project.
8. Assembling the project team and business owners on a daily basis throughout the project.
9. Having the team reflect at regular intervals on how to become more effective, then tuning and adjusting behavior accordingly.
10. Measuring progress by the amount of completed work.
11. Continually seeking excellence.
12. Harnessing change for a competitive advantage.

Architecture of the solution

We both took on a role similar to a solution architect. We worked together and spent time deciding what roles and jobs needed to be filled, and then put that in to practise.

We both have similar work styles and ethics so it became apparent that we did not need one particular individual to specify the solution.

Test Plans

We incrementally tested code after each new feature or change was added. The plan was simple and the project in scale is rather small. We playtest through the level after each change to ensure no new bugs have come up. We also both test each new part of code to have two different views on the testing process and the outcome.

Then again we both tested the finished product thoroughly to ensure we were both satisfied before submission.

Conclusion

Overall, it was an enjoyable project to work on and we believe as a team we worked well together.

We chose a topic we were both interested in by designing a game, which made the workflow more natural.

Unfortunately, along the way there were more challenges than first expected. With the closing down of colleges it made us difficult to meet and decide on how to attack the project. We struggled initially getting used to using virtual conferencing tools to discuss the project.

We also had various technical issues between us with our home systems which did cause unforeseen delays.

However, in the end we benefitted by learning new skills along the way that we did not realise would be needed.

Final words

Thanks Daniel for supporting us along the way. It was a great benefit to have your expertise and experience at hand when we needed it.