

Element D2b Instrument Essay

Introduction

I have decided to make a video game with music and sound effects. I have previous experience working on video games having created an educational film, interactive presentation, and online resource with lesson plans in 2006. This was called “Shakespeare Works!” and was an introduction to Shakespeare for students of all abilities, mainly aimed at Years 5 and 6 primary school children (ages 9 – 11). The project was based on a live performance which I performed in schools across the UK. At this time my skills were utilised as writer, director and producer. I had no knowledge of coding for game design and employed freelancers to work on the project. I understood that they used **Macromedia Director** to develop the interactive presentation and deliver it on DVD-ROM with the film. **Director** was the flagship product for **Macromedia** from 1992; an interactive multimedia-authoring tool used to make presentations, animations, CD-ROMs, and information kiosks. (Wikipedia 2022)

I came up with the concept for the games. The film was broken down into six chapters. The student would watch the film and then play the games; the Shakespeare “challenge”. They would answer multiple choice questions based on each chapter to test their knowledge. For each correct answer they would win time to play the skill game. The skill game was a fun reward for doing well in the comprehension test. For example, the students would watch the section of the film about Shakespeare’s Globe Theatre, answer ten questions about the Globe and then play the game “Put the Rich in their Seats”, where they would have to place everyone correctly in the theatre as they arrived for the performance; the rich people sitting in the galleries, the poor people standing in the yard, the players on the stage and the musicians next to the stage. The Shakespeare Works! DVD-ROM was sold until 2012. Advancements in Mac OSX and Microsoft Windows meant that the interactive presentation required constant updating, which was too expensive at the time, so unfortunately the gaming part of the program was no longer available, even though the film and online resource was still usable.

Having the option to make a video game for this part of the module excited me, because it is almost 10 years since shelving my Shakespeare game and I was interested to find out, whether the software had been created to enable me to rekindle my Shakespeare games and build them myself. I want to create the games for smartphones and tablets to target children ages 9 – 11 so I was keen to research and try out the platforms available. Unfortunately, when I booted up the original artwork files for the games, all the files had become corrupted. To re-create all the artwork again seemed like a thankless task and the more I thought about it, the less passionate I became about redeveloping this project.

After “Shakespeare Works” I wrote a series of children’s picture books called “The Adventures of Uncle Les”. Uncle Les is a fun and lovable farmer, who lives his life a little differently to everyone else in the village. The stories are aimed at children who are starting to learn to read (ages 5 – 7), and also students with English as a second language (ages 7 – 10). The idea is for the parent and child to read together. The parent helping the child to read and eventually the child reading on their own. They are rhyming stories with an element of interactivity and like “Shakespeare Works”, they are based on a live performance, which I performed in schools across the UK and Europe. I also made a short animation with a theme tune to help promote the books and live performance, which you can view on my website at: www.kieronjones.com. The books are available to purchase via Amazon and I have included one of the stories, “Uncle Les and his Fruit and Veg”, at the end of this essay in the appendix. I have all the artwork, which I created for the animation, having scanned in the original hand painted artwork into Adobe Illustrator and then vectorising the images. I realised I could use these images for a video game, especially the digital puppet of the main character Uncle Les. This would provide design continuity across all platforms. Furthermore, I believe a good fun game could help drive traffic to my website, promote my books and drive parents to Amazon to buy the books! As with “Shakespeare Works” I believe the best devices for the game are smartphones and tablets and I have decided to focus on the UK market aiming the game at young children (ages 5 – 7). This age group enjoy playing simple and fun 2D arcade games. According to Statista 50% of them played games online in 2020/2021 (Statista 2022). I felt confident that with over twenty years’ experience writing and performing shows for children that I would be able to create a game that they will enjoy.

Ofcom in the UK estimates that the average 3-4-year-old spends three hours a day in front of a screen. This rises to four hours for ages 5-7, 4.5 hours by ages 8-11, and 6.5 hours for teenagers.

OFCOM (2019) *Why Children Spend Time Online*

In the UK children aged between 5-16 years spend an average of 2-3 hours per day watching television, 1-3 hours on the internet, 1-2 hour playing video games and over an hour on mobile phones (not talk), a total of 6.3 hours of screen time per day.

NHS (2021) *Screen Time*

Unity and FMOD

This module required reading the book “Game Audio with FMOD and Unity” by course tutor Ciarán Robinson, published by Focal Press in March 2019. I found this to be an excellent, practical introduction to the free online software packages **Unity** and **FMOD Audio Middleware software**.

Unity was launched in 2005 as a game engine that would make game development accessible to more game developers.

As of 2020, software built with Unity's game engine was running on more than 1.5 billion devices. According to Unity, apps made with their game engine account for 50 percent of all mobile games and are downloaded more than 3 billion times per month, and approximately 15,000 new projects are started daily with its software.

De Vynck, Gerrit (2020).

FMOD was launched in 1995. It is a primary sound-effects system for **Unity** that functions like a digital audio workstation (DAW). It is also very versatile, because it has been written in C++; a coding language that enables it to be used across many different gaming platforms.

Unity is suitable for both 3D and 2D game creation for desktop computers, smartphones and tablets. I was excited to get “stuck in”, learn the basics and following the step-by-step instructions provided by Ciarán’s book, I was able to create a basic First-Person Shooter 3D game with music and sound effects.

Unity provides online tutorials via <https://learn.unity.com> and I wanted to add to my skill base by following their lessons, specifically for 2D game creation. I completed the 14 hour long “Ruby’s Adventure: 2D beginner” and the 3 hour long “Creator Kit: Beginner Code”. I found both tutorials very easy to follow and it enhanced my understanding of the coding language C++.

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or “C with Classes”. The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms.

Stroustrup, Bjarne (1997)

I think video tutorials are an essential tool when learning new software. With **Unity** the videos allow you to see exactly how your game should appear and operate and allows you to go back and check that you haven’t missed anything out. This is very important when writing C++ code, because simply misspelling a command or, for example, omitting a curly brace “}” to indicate the end of a main’s function definition will stop your game from working.

I did encounter many stumbling blocks when trying to write C++. It is a complicated language, and it wasn’t made any easier, because I had to write everything using **Text Editor** (TE) for Mac. When I originally downloaded **Unity**, I was prompted to download the code editor **Visual Studio** (VS) by Microsoft, available at: <https://visualstudio.microsoft.com>

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop computer programs, as well as websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code.

Wikipedia (2022)

Visual Studio for Mac employs many of the same tools as its Windows counterpart: for example, the Roslyn .NET Compiler Platform is used for refactoring and IntelliSense. Its project system and build engine use MSBuild; and its source editor supports TextMate bundles. It uses the same debugger engines for Xamarin and .NET Core apps, and the same designers for Xamarin.iOS and Xamarin.Android.

Wikipedia (2022)

However, when using VS for Mac, I noticed that my computer was overheating and when I checked the computer's performance, it revealed that VS for Mac was using up more than 2000% of my CPU even when it was working in the background. Even though I researched solutions on forums and VS for Mac support, there appeared to be no answers. In the end I had to resort to using TE, which worked fine, but has no shortcut benefits and doesn't highlight any errors in the coding. This meant that writing C++ was a much harder task, especially when looking for that missing command that is stopping your game from working.

Even though I found the **Unity** learning videos interesting, they could only get me to a very basic level of 2D game and nowhere near the look and feel of arcade game I wanted to create for my character Uncle Les. I wanted to explore more the online tutorials provided by professional game developers via **YouTube**. Here there were many developers offering videos on how to make already famous arcade games like **Pac Man**. I wanted to explore the potential to learn the codes for these games and then adapt them for my Uncle Les concept.

In 2008 I taught myself how to build websites using a similar technique. Initially I learnt from books how to write **HTML** and **CSS** using the web development tool **Dreamweaver**. Once I had grasped the basics, I could create a simple website. However, to produce a website in the style I desired, I had to download pre-made codes, templates, and adapt them.

Searching games' developers on **YouTube** I came across a developer called Jason Weimann. According to his LinkedIn profile (<https://www.linkedin.com/in/jason-weimann-bab0984>), Jason has been a developer and instructor for unity3d.college since March 2017. He had an online video called "How to Make a Game - Unity Beginner Tutorial". You can view this here: https://www.youtube.com/watch?v=OR0e-1UBE0U&list=RDCMUCX_b3NNQN5bzExm-22-NVVg&index=4

The reason why I chose Jason's tutorial was because it had already received over 1.5 million views plus a lot of great, positive feedback. I was excited because he would cover every aspect of creating the massively popular 2D game "Angry Birds", providing the artwork and step-by-step instructions for writing C++. "Angry Birds" was originally released by Rovio Entertainment in 2009. Rovio have produced 21 games in the series to date with 66.8 million monthly active users (Videogamesstats 2022). The "Angry Birds" brand has also spawned TV shows, movies, food products and a theme park, just to name some of the lucrative spin-offs from this global hit!

Jason's tutorial is very detailed and even though the watch time is around 3 hours, it took me more like 5 hours to complete. This was mainly because Jason used VS to write the code, which meant I had to stop start the tutorial a lot to make sure I had copied the code correctly into TE. My ambition at this stage was to complete Jason's tutorial and then try to reskin the background, the objects, the characters, and enemies with my Uncle Les graphics. For example, Uncle Les could be throwing vegetables to try and knock over different farm animals, who would "moo", "honk", "cluck", "neigh" and "bleat" accordingly. After completing the online tutorial, however, I didn't feel my knowledge of C++ was enough for me to confidently build more levels and furthermore I wasn't happy with how my Uncle Les character was appearing on the screen.

Oblius Games

I decided to research further the options available for downloadable templates for Unity based 2D arcade games. [Obliusgames.com](https://obliusgames.com) appeared to answer my prayers promising high quality templates and unlimited support. Their codes also work on both IOS and Android; perfect for my game targeted at smartphones and tablets.

There were two games that inspired me; "Banana Bash" and "Smash the Totem". In my version of "Banana Bash" called "Apple Picking with Uncle Les", I could envisage Uncle Les hopping up through trees picking apples and avoiding angry "cawing" crow. In my version of "Smash the totem" called "Pack the Vegetables with Uncle Les", Uncle Les could be packing baskets of vegetables as they fall from the sky as fast as he could, trying not to drop them!

Unfortunately, the templates I purchased didn't work on the latest version of **Unity**. There were many compiler errors created due to files being replicated. Here as an example of one of the issues encountered:

```
Library/PackageCache/com.unity.purchasing@4.1.0/Runtime/WinRTStub/Factory.cs  
(15,23): error CS0433: The type 'IWindowsIAP' exists in both  
'UnityEngine.Purchasing.WinRTCORE, Version=0.0.0.0, Culture=neutral,  
PublicKeyToken=null' and 'winrt, Version=1.0.0.0, Culture=neutral,  
PublicKeyToken=null'
```

Oblius offered solutions, for example, deleting the replicated files, however, once one issue was resolved a similar error appeared. After exchanging more than 20 email, they were unable to rectify the issues and, finally, they pointed out to me that in their small print they state that their games only work on an older version of **Unity** 2018.3.3f1. Unfortunately I was not able to download the version required. To my

dismay, it was not compatible with my MacBook Pro 2019 running Catalina 10.15.7. My journey with **Oblivus** regretfully had to end.

Buildbox

Returning to the internet, I continued searching for a solution and I came across **Buildbox** - <https://buildbox.com>

Buildbox was founded by Trey Smith in August 2014 with the goal "to democratize game development and create a way for anyone to be able to create video games without having to code". It is a cross platform development tool that can be run on both Windows Operating System and OSX. Primarily used to create mobile apps, Buildbox exports finalized games to iOS, Android, Amazon Mobile Devices, Amazon TV, Mac, PC and Steam.

Wikipedia (2022)

I was impressed that there was 2D software available for free called **Buildbox Classic**. I liked the look and feel of the website. I was sold by their vision to create games for everyone, that it was their mission to break down the barriers for game developers. The creators claimed that, due to **BuildBox's** unique user interface, the process of making games could be fluid and not held back by the requirement for any scripting, programming or software design experience.

On downloading the software, I completed the provided online tutorial to build a basic game called "GLTCH". You can access the resources here:
<https://www.buildbox.com/tutorials-buildbox-classic/>

The online tutorials were easy to follow and I was very happy with the final result; I had made a 2D working game! I was able to adapt and reskin the game without any technical issues plus I could also try out and play the game in "real time" I found the creative process to be very intuitive, enabling me to quickly make and review changes as I developed my gaming ideas. I was able to easily drag and drop my assets onto the **BuildBox Drop Wheel** as a background, character, object or action and then edit with no coding required. It was also simple to add music and effects by dragging and dropping them to the relevant asset.

Buildbox Classic provides the user with a selection of 2D arcade templates that you can customise. I chose the one called "Racer" which is a simple game where you move your vehicle to dodge obstacles and collect tokens which equate to points. The perspective for this game is as if you are looking down on the action. I wanted to change this so that my Uncle Les character is looking at the person playing the game and so that the obstacles and tokens are falling from the sky. I think Uncle Les is a fun and lovable cartoon character and it is important for him to be looking at the player so he can engage and connect with the player, especially because the game is aimed at young children aged 5 – 7. The game I have created is called "Eat biscuits with Uncle Les!". I want the game to be fun so Uncle Les catches and eats biscuits falling from the sky. The biscuits are the tokens. The player is rewarded points for each biscuit munched and consumed. This is in reference to the theme tune's lyrics; "he sits beneath his tree, dunking biscuits in his tea!". The theme tune

plays when you are on the menu screen. As the levels get harder the idea is that the objects that Uncle Les needs to dodge get bigger, funnier, and more frequent. The objects start as small snowballs, but progressively get bigger and then in silly shapes like a cow or a donkey. I have produced two tunes to enhance the gaming experience. I don't want the game to be too manic because of the target age group. The background music while in the game world is therefore light yet fun with a nice beat and rhythm so the player can hum and sing along. I created an upbeat tune for the "Game Over" screen so the player doesn't feel sad and is encouraged to play again. Uncle Les also makes silly noises and talks when he munches on the biscuits and is hit by the falling obstacles. This adds to the humour and fun encountered while playing the game.

Conclusion

To create a 2D game is not an easy task and, on this journey, I have made many mistakes as I endeavoured to learn new software and a new coding language. I have had to work on trial and error and even though the learning materials have been excellent, gaming software is developed so quickly that one has to be constantly vigilant to new updates in order to guarantee that the game can be published. I learnt that the hard way with **Visual Studio** and my investment in **Oblivus Games**.

Even though **Buildbox** claims that you don't need any scripting, programming, or software design experience to make a game, I would recommend reading Ciarán's book, following **Unity**'s online tutorials and the online tutorials of professional games developers via **YouTube**. Having a basic understanding of game creation and C++ definitely helped when I was using **Buildbox**, especially when adding animation to assets and adapting how assets interact with each other. In hindsight, I don't regret my experience with **Oblivus Games** for it was that bad experience that led me to **Buildbox** and ultimately the correct solution for my Uncle Les game.

It is important to note that to publish the game to IOS and Android, **Buildbox** has a minimum requirement of 100 playable screens. They state that anything less doesn't justify enough gaming time to make the game publishable to these platforms. My game is still in the early stages. My daughter, who has just turned 8 years old, and her friends have been my main source of feedback to date as I have worked on a demo version of the game. They have found it a lot of fun, however, through talking with them and watching them play, I intend to make the following improvements to enhance the experience.

- 1) The game doesn't finish when Uncle Les is hit by one falling obstacle. This means that it would take three hits from a falling obstacle before the game ends. This is possible by adding a Health Component to Uncle Les. This would extend game time and should also add to the player's excitement as they move Uncle Les to eat as many biscuits as he can.
- 2) I want to adapt some of the tokens so they give health back to Uncle Les as well – maybe a golden biscuit!
- 3) Regarding the falling objects I want to explore how to make them more fun. Maybe they can affect the background music, potentially speeding up the background tunes or even drowning them out.

- 4) To include an Uncle Les story page from the menu screen with a story to read, a recording of the story, URL link to my website for more information and to purchase the books.
- 5) I want to understand better the potential for “in app” purchases and how this can be used in my game,

I am excited to continue working on my Uncle Les game and my ambition now is to create two further 2D games in the future.

Appendix

Here is one of my stories from “The Adventures of Uncle Les”:

Uncle Les and his Fruit and Veg

Now Uncle Les grows fruit and veg mostly broccoli,

And when he grows too much, he leaves the rest in a basket by his tree.

Anyone can help themselves for absolutely free!

He’s made a little sign as well:

“Help yourself. They’re lovely!”

But people in the village, they do not want his veg.

They rather go to the local shop and buy it there instead.

“I for one do not see why he grows that broccoli?”

“When the shop sells three for 50p,

Why waste your time and energy?”

“He should get a job, go up the shop.

Buy greens with his money!”

So, one day I ask my Uncle Les

“Why do you grow your fruit and veg?”

And this is what he says to me

While munching on some broccoli:

“Now people in the village, they are too busy

To grow their own, in their home,

For them it's not convenient, see?

The local shop sells vegetables every day and every night.

They go there with their shopping bags, buy everything in sight!

There's three for one, there's two for one, there's buy one get one free.

A bargain bin full up with tins of every green veggie!

They're happy buying broccoli. I'm happy growing mine.

It's not a waste of energy but the way I pass my time.

You see, I get up in the morning and I get my watering can

And water all my veggies growing bigger in the land.

I listen to the birds as they chatter overhead.

If I hear a roll of thunder I go sit down in my shed.

If I see a ray of sunshine, I get my deckchair out instead.

My vegetables are free of charge, that they cannot see,

From the biggest bargain bin in the world

With a lifetime guarantee!

Plant one. Grow one. Gather up the seeds.

Plant them. Grow them. You've dozens extra, see?

Mother Nature's special offer is as cheap as cheap can be.

I've grown too much for me to eat, so go on! Have them for your tea!”

And off he potters down the lane

With his silly old smile that would always remain,

To the great big garden and the great big tree

With the basket underneath full up with broccoli

And the little sign above for everyone to see:

“Help yourself. They’re lovely!”

Reference List

De Vynck, Gerrit (2020) *Unity Technologies Aims to Bring Video Game Tools Into the Real World* Available at: <https://www.bloomberg.com/news/articles/2020-05-07/unity-technologies-aims-to-bring-video-game-tools-into-the-real-world>

(Accessed 17 February 2022)

NHS (2021) *Screen Time*

Available at: <https://www.nhs.uk/about-us/professional-support-sites/screen-time/#>

(Accessed 17 February 2022)

OFCOM (2019) *Why Children Spend Time Online*

Available at: <https://www.ofcom.org.uk/about-ofcom/latest/media/media-releases/2019/why-children-spend-time-online>

(Accessed 17 February 2022)

Statista (2022) *Share of children in the United Kingdom (UK) who ever play games online from 2015 to 2021, by age group*

Available at: <https://www.statista.com/statistics/274427/online-gaming-among-children-in-the-uk-by-age-group/>

(Accessed 17 February 2020)

Stroustrup, Bjarne (1997) *The C++ Programming Language (Third Edition)*

Available at: <https://en.wikipedia.org/wiki/C%2B%2B>

(Accessed 17 February 2020)

Videogamesstats (2022) *Angry Birds Stats and Player Count*

Available at: <https://videogamesstats.com/angry-birds-stats-and-facts/>

(Accessed 17 February 2020)

Wikipedia (2022) *Buildbox*

Available at: <https://en.wikipedia.org/wiki/Buildbox>

(Accessed 17 February 2022)

Wikipedia (2022) *Macromedia*

Available at: <https://en.wikipedia.org/wiki/Macromedia>

(Accessed 17 February 2022)

Wikipedia (2022) *Microsoft Visual Studio*

Available at: https://en.wikipedia.org/wiki/Microsoft_Visual_Studio

(Accessed 17 February 2022)

Wikipedia (2022) *Visual Studio for Mac*

Available at: https://en.wikipedia.org/wiki/MonoDevelop#Visual_Studio_for_Mac

(Accessed 17 February 2022)