THE STORY BEHIND...

DIZZY FAST FOOD

Can it be? A new Dizzy game from The Oliver Twins? Yes, it can! For the Twins have been working on a new version of 'Fast Food' Dizzy this time developed with the FUZE4 coding application for the Nintendo Switch. So, with the prospect of a new Dizzy game for the Switch on the horizon, we at Fusion felt we had to find out more. Colin Bell catches up with Phillip Oliver to talk about a new Dizzy project.

Colin Bell: Philip, thank you for taking the time to speak to me today. I believe you've been working on a new Dizzy Project with your brother Andrew for the Nintendo Switch using the FUZE4 development tool. Before we talk about the game, can you tell us all a bit more about FUZE4?

Philip Oliver: Yes! FUZE4 is essentially a coding application which was created by a man named Jon Silvera. It features a language that is ideally suited to make games and apps and is fully accessible to users of all ages and experience. As you may know, Andrew & I are passionate about encouraging the next generation to learn the digital skills of tomorrow because of how important they are within the gaming industry. So, when we saw what Jon was doing with FUZE we could see that he 100% understood as we did the importance of teaching the younger generation to code. Since then, we have got to know Jon well, helped out with some successful fundraising and continue to help promote and spread the word about the fantastic FUZE application.

CB: Sounds like a fantastic piece of software, so where does Dizzy come into all this?

PO: That initially all came about some time last year when speaking with Jon he recommended that perhaps we could create a new version of Fast Food Dizzy using FUZE4, which we felt was a good idea as all the graphics were already there. Fundamentally Fast Food is a maze game which we've always felt is an ideal start point for making the first game.

So at first, we said, yes that would be a great idea — 'someone' not us, should do it. Before we knew it, Jon had contacted a man named Jonathan Temples who about a week later sent over

lovely concept screenshot showing Dizzy in a 3D maze with some enemies and food dotted around the maze. Both Andrew & I thought this looked awesome and that it would play well on the Switch so immediately wrote back saying, 'That's brilliant! Let's get someone to make this'. A sprite sheet then turned up with all the different foods, power-ups, characters and pictures of hats.

CB: Yes, I noticed the hats in the concept art you kindly sent over. What part do they play in the new game?

PO: PO: Well, at first we asked Jonathan, 'Why did you put all these hats in? Fast Food never had hats'. Jonathan replied, 'Well I thought they would be quite cool' and to be fair we thought, well Dizzy does always look better with a hat on, hence why from Fantasy World Dizzy onwards you'll always seem Dizzy wearing

one. So now Andrew & I are thinking, well we have all these hats, how are we going to integrate them? So, with ten hats in total we felt that if we made ten levels then and had a different hat per level, then there's now a reason for the hats, but what are they used for? Putting our heads together we thought of Pac-man and the power pill that allows you to eat the ghosts and so we thought that putting on a hat was a good visual cue that Dizzy is now in control and has become the predator and not the prey. So that's how we came to use the hats, never throw away a good graphic.

CB: It sounds like the game was beginning to take shape, and at a quick pace, what happened next?

PO: Next, we had to find a programmer to help code the game. At the time the Corona virus lockdown had come into full effect and with

Andrew & I unable to do our usual game industry business we found ourselves with a lot more time on our hands. We did a significant update of our website which took about two months, and when showing off the finished website to Jon Silvera he said, "You guys need a new project, you guys should program Fast Food Dizzy!" At first, we were a little unsure, but then Andrew started to have a little bit of a play with FUZE4 and found he could quickly put sprites on screen. He then created a basic grid pattern for the maze and started to experiment with bigger mazes — this led to him putting in scrolling, and then we thought well if you scroll you can surely zoom? We took the best part of a week to get that working. So we started to believe that we could make this game.

The only problem was that we knew that there were a considerable number of Dizzy

```
3070
3071        VertMax=-((MazeTopEdge)*zoom)
3072        VertMin=-((MazeBottomEdge)*zoom)+_GameAreaMidY*2
3073        ZoomY.Start=ZoomY.Curr //
3074        ZoomY.Curr=ZoomY.Start
3075
3076        ZoomY.End=clamp((_GameAreaMidY-y*zoom),VertMin,VertMax)
3077        ZoomY.Count=_PanZoomCount ///(abs(ZoomY.Start-ZoomY.End)/4)
3078        ZoomY.Size=_PanZoomCount
3079
3080 return void
3081
3082 /////////////////////////////////////////////////////////////////
3083 function SetZoom(ref Zoom,start,end,count)
3084        Zoom.End=end
3085        Zoom.Start=start
3086        Zoom.Curr=start
3087        Zoom.Count=count
3088        Zoom.Size=count
3089 return void
3090 /////////////////////////////////////////////////////////////////
3091 function SetZoomZ(goto)
3092        if goto>WidthZoomLimits[MazeWidth-16] then goto=WidthZoomLimits[MazeWidth-16] endif
3093        if goto>HeightZoomLimits[MazeHeight-16] then goto=HeightZoomLimits[MazeHeight-16] endif
3094        ZoomZ.End=goto
3095        ZoomZ.Start=MapScale
3096        ZoomZ.Curr=MapScale
3097        ZoomZ.Count=_PanZoomCount
3098        ZoomZ.Size=_PanZoomCount
3099 return void
3100 /////////////////////////////////////////////////////////////////
3101 function SinZoomTowards(ref Zoom)
3102 float Progress
3103        Progress =(sin((((_PanZoomCount-Zoom.Count) / Zoom.Size) * 180) - (180 / 2)) + 1) / 2
3104        Zoom.Count-=1
3105        Zoom.Curr = lerp(Zoom.Start, Zoom.End, Progress)
3106 return void
3107 /////////////////////////////////////////////////////////////////
3108
  Show Keyboard    + Run    - Return    Undo/Redo    Select
```

Above: Code behind the game.

fan games out there and that we have a reputation of quality so if we were going to make Fast Food Dizzy, we had to do it properly or not at all. The last thing we want is for people to say, well it's taken 25 years for the Oliver Twins to write a new Dizzy game, but you know what Fantasy World Dizzy is better and they should've just stopped there. It makes you suddenly realise the benchmark you've set yourself.

CB: So, with the benchmark set and Andrew getting to grips with the coding and implementation side of things, which parts did you focus on?

PO: I said I would do all the maps and all the design which originally I was going to do old school on graph paper. But then I thought if Andrew needs ten mazes and now they are quite large why do I want to do this on graph paper when we could potentially put together a little basic editor with which I could then create maps, play test them and then tweak them, which turned out to be quick and easy to do.

Then I thought, well we have an editor now, can we tidy this up and leave it in there? So that is what we did — Fast Food has ten basic maps and ten user-definable maps which you can share with your friends or upload to the central FUZE4 hub for wider sharing.

CB: An editor is a fantastic idea and addition to the game. Like the original Fast Food will we see any mid-level cartoons of Dizzy dispatching his enemies in different humorous ways.

PO: No, sadly not because we just felt that this would bulk the code out and people aren't going to learn anything from it. Jon has however added some code to produce a title screen which features all the characters so actually if anyone did want to know how to create a cartoon sequence then the code is now essentially there.

CB: So, you view this new version Fast Food as more of a learning experience as well as a fun game?

PO: That's right, it wasn't all about making a fun game but more about trying to inspire kids to learn to program, so we had to be careful how much code there was. We also had to make sure that although it's a great game with a high benchmark, that actually when someone opens the code, which you can simply do by pressing the '+' button at any time on the controller, they don't get put off completely.

We want kids to tinker and to mess with the code. We want them to start with the simpler code and learn first how it works and then only look at the more complicated sections of code, such as the zoom function, only when they've got the hang of the other bits. They can just tinker and mess with it and let's see what happens when they change this number or variable, which is essentially the same as we did and how we learned to code.

This was also another reason why we decided to implement a clock mechanic within the game because as well as adding to the gameplay

Below: Andrew play-testing the game.

we felt that it provided yet another core mechanic that you could learn and understand how to put into your own games with just a few simple lines of code.

CB: So essentially you're helping to lower the barrier to what appears to be black magic to many and make it far more approachable and more comprehendible.

PO: Exactly, which is what BASIC did for us when we were starting out and is what Jon is trying to do with FUZE4.

CB: I see, as with sticking to the tradition of many modern games that you have included some 'unlocking' and 'achievement' features in Fast Food, could you tell us more about them.

PO: The way modern games have evolved is with the concept of 'unlocking' so we thought we'd do an unlock system whereby the first three levels will be unlocked, and after that, the players

will have to unlock each level by completing the last.

It's the same with the user definable levels which when unlocked open what is essentially a duplication of the level you've just beaten, the difference being that you can now go in and mess around with it, add twice as many enemies, knock out a few walls and before you know it your having fun creating you own levels which you can go on to share with your friends.

Regarding achievements, we did put a high score system in so that every level has its own score to beat. We have implanted a star system as well, whereby if you want to, you can collect all the stars for extra points. Collect all the stars though, and you'll get a big bonus. So, if you want the best score, you have to get all the stars, which is not easy.

CB: So, to confirm, Fast Food will not be a physical

Above: The game running on the Switch throught the FUZE development environment.

release?

PO: No, we want this to be free, and we want people to have fun with it, but at the same time we want Fast Food to inspire people to learn to code so it will be released as part of the FUZE4 player which currently as we speak, Jon is waiting for Nintendo to approve so that new player can be officially released in the Switch market place.

CB: Fantastic. We'll I'm sure all Switch owners will look forward to playing Fast Food when the new FUZE4 player is finally released. In the meantime, do you have any updates for us regarding Wonderful Dizzy on the ZX NEXT?

PO: It is getting close to an Alpha release now, I promise. All the levels are finished and playable, and all the objects are in there.

The Programmer is currently ironing out some of the object logic which he is almost done, and so it's just playtesting after that. There have been a few compression issues with fitting all the dialogue in due to the restricted memory size, but we both think that there's a real art form in writing 8-bit games that does not exist anymore, the art form of squeezing everything into a small space which Andrew & I feel we both mastered.

So, when it came to doing Wonderful Dizzy initially the developers wanted to use all the extra memory the ZX NEXT

was capable of using, but we felt that wasn't playing to the art form and that we should aim to fit the game into 48K like the original Spectrum. We were however talked around to using 128K which was required to get 60 frames per second which we thought okay, we're happy to make that compromise.

CB: Some good news there for NEXT fans, so after Fast Food and Wonderful Dizzy, what does the future hold of Dizzy?
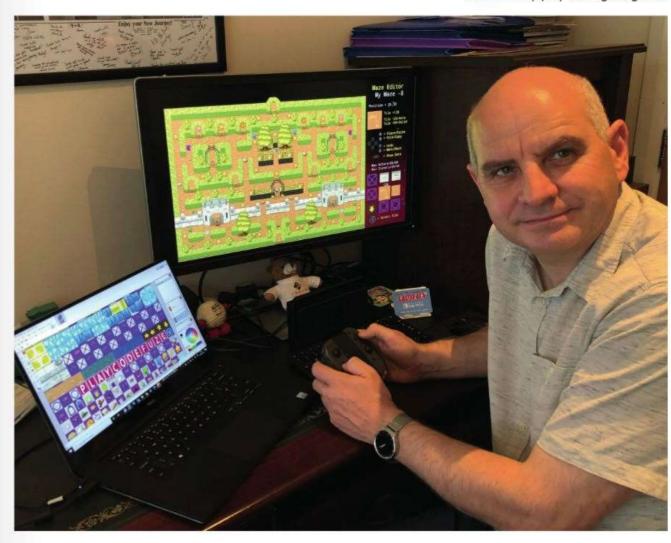
PO: First things first lets Wonderful Dizzy finished, then 'IF' the game does well then, we may think about

updating the graphics and doing it on FUZE4 for the Switch. Which may happen, but we'll need to see.

CB: And with that tantalising possibility I think we'll end there, Philip Thank you for taking the time to talk to me today.

PO: Thanks Colin, my pleasure.

Below: Philip play-testing the game.