Medium

Interview with RollerCoaster Tycoon’s Creator, Chris Sawyer

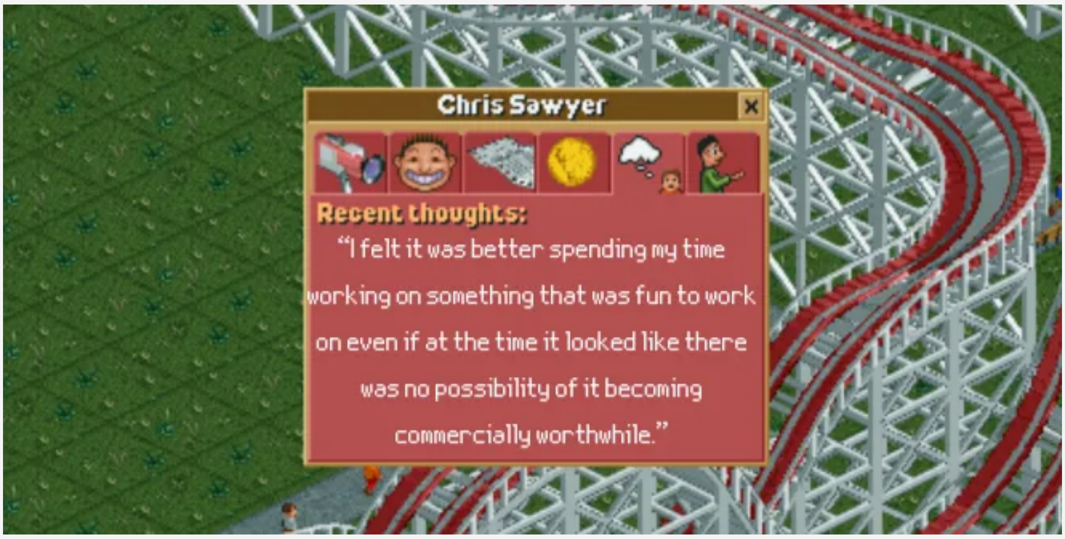
CarlyF · Follow
Published in atari-club
12 min read · Mar 27, 2024

When and how did you first start thinking about creating a game about rollercoasters?

Back in 1996 or 1997 I was really struggling for inspiration and focus while working on Transport Tycoon 2. This sequel to my earlier game Transport Tycoon seemed a good idea at the time but the planned improvements over Transport Tycoon Deluxe weren’t that great, and the power and memory of PCs at the time weren’t enough to allow a more ambitious approach with the sequel. At the same time I was developing quite an interest in rollercoasters, a fascination with their design and engineering, and I’d also played games like Theme Park which included crude representations of rollercoasters. It got to the point where I needed to take a break from Transport Tycoon 2 and have a bit of fun playing around with other ideas for a while, and luckily the success of Transport Tycoon Deluxe meant I could afford to postpone the sequel and take some time off.

Was it an easy decision to start working full time on RollerCoaster Tycoon?

It was a long gradual process — I was under contract to create Transport Tycoon 2 so initially all I was doing was postponing it while I took a break and had a bit of fun exploring other ideas. I expected to go back to Transport Tycoon 2 at some point once inspiration and focus returned. The great thing about the way I worked back then was that I could get ideas programmed and working really quickly, as I was basically doing it all myself so no issues with managing (or paying for) a large team or having to explain or justify ideas to anyone else. So tinkering with different ways of getting a rollercoaster working in the Transport Tycoon 2 isometric gameworld system I was working on was quite a quick process, and fun too. I went down a lot of dead-ends, and the display system ended up having to be considerably re-written and improved before anything reasonable was working, but it just sort of grew gradually and I felt it was better spending my time working on something that was fun to work on even if at the time it looked like there was no possibility of it becoming commercially worthwhile.

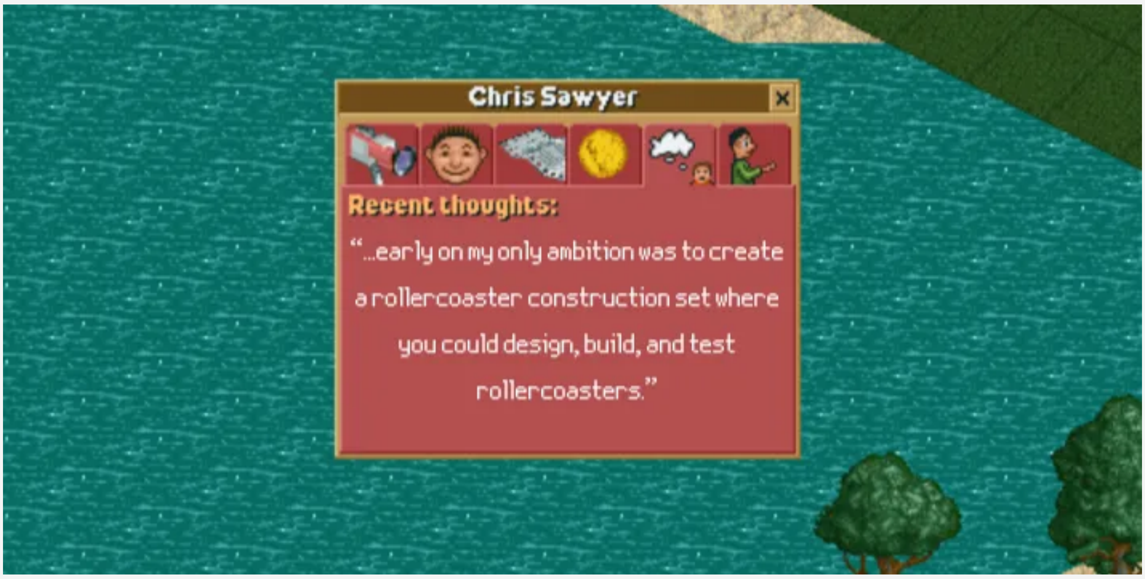

Chris Sawyer

Recent thoughts:

"I felt it was better spending my time working on something that was fun to work on even if at the time it looked like there was no possibility of it becoming commercially worthwhile."

How did the design process for RollerCoaster Tycoon work?

There was very little in the way of advance designing or planning at all — I just worked from the bottom-up, starting with the technical aspects that interested me, and the game gradually grew organically from there. Much of the early days were spent trying to come up with a data and display system to make it possible to build simple rollercoasters and have them functioning and looking good, and that was as far as any design plan or ambition went at that point. Once a particular stage was working well enough the plan for the next stage grew, but without really knowing what the finished game might be like, or even if there would be a game at the end of it. I think early on my only ambition was to create a rollercoaster construction set where you could design, build, and test rollercoasters.


Chris Sawyer

Recent thoughts:

"...early on my only ambition was to create a rollercoaster construction set where you could design, build, and test rollercoasters."

Which elements of the game's design were most important to you?

Early on the most important thing for me was to get the look and feel of the rollercoasters right — not necessarily completely realistic but enough to make them look good aesthetically and give the right impression of speed and acceleration and so on. I also felt it important to show the pros and cons of different types of rollercoaster track and trains, initially just with one type of tubular-steel tracked rollercoaster and one type of wooden rollercoaster. I wanted players to learn the unique limitations and possibilities of each type, and realise that even the lowlier and more restricted types of rollercoaster can be fun in their own way rather than always “bigger or faster or more inversions is better”. That’s perhaps still unique about the original RollerCoaster Tycoon games compared to other similar games — RollerCoaster Tycoon’s quantity and depth of different types of rollercoaster is unequalled in other games and each type has its own pros and cons which need to be learned and exploited to make the most of them. That’s not to say it was the only core aspect of the game I wanted to get right, but it’s probably the earliest one at the heart of the game.

How did very early versions of the game look?

It started off looking very like Transport Tycoon, complete with ugly angled sections of straight track for slopes and corners and so on. I remember the first test graphics Simon Foster did for the game were sections of straight rollercoaster track at various angles, either to match the length of a landscape block or half a block long. It quickly evolved into something much closer to the final structure of preset smooth curves and smooth slope transition track sections which could be plugged into the block-like landscape. The whole isometric display engine evolved throughout development, sometimes with small changes or additions here and there, and sometimes whole sections completely re-written when needed or when ideas warranted it.

Why did you write RollerCoaster Tycoon in assembly language (machine code) rather than a high level language like C?

At the time there was no option other than to use machine code for RollerCoaster Tycoon. I was struggling to keep performance at a reasonable level on PCs of the era even using highly optimised machine code, and writing in a high level language would have made the game far too slow, or limited the complexity of the simulation in order to keep speed up. The look and feel of the game was really important to me and part of that was to maintain a high frame rate while also having a large enough and detailed enough view of the park, and also being able to simulate enough trains and rides and guests to avoid the game feeling constricted. It wasn’t just small chunks of code that benefited from being very efficient machine code either — because of the number of objects the game had to cope with, virtually every bit of code involved with object handling needed to be ultra-efficient or the inefficiencies quickly multiplied up with a busy park. I’ve also always preferred low-level assembler programming and can write machine code faster and more reliably than any high level language, so for me the only downside was lack of compatibility of the x86 machine code with other platforms, which at the time wasn’t too much of an issue as the game was really only aimed at desktop PC players. Latterly the machine code came back to haunt us when the decision was made to re-launch the original game on mobile platforms as RollerCoaster Tycoon Classic a few years ago, and it took several years and a small team of programmers to re-write the entire game in C++. It actually took a lot longer to re-write the game in C++ than it took me to write the original machine code version 20 years earlier.

Did RollerCoaster Tycoon take inspiration from any other games?

I remember playing Theme Park, and some of the more serious business management games like Railroad Tycoon and the like, and certainly they influenced my games. I was never personally into the business side of these games as much as other players, but really enjoyed the construction and the tinkering, and watching the little world running and seeing what was going on, so in some ways I wanted to create a simpler game than the out-and-out business strategy games. Theme Park frustrated me in that the rollercoasters were represented so crudely in the game. At the time there were a few really nice looking isometric 3D strategy games on the PC –Sim City 2000, my own game Transport Tycoon of course, and possibly several of Sid Meier's strategy games, and I thought that something like Theme Park with a detailed isometric viewpoint like those games would look fantastic. I remember reading a "making of" article in a games magazine about Theme Park and the developers had actually tried isometric graphics but dismissed them due to too many problems — maybe that inspired me too!

The isometric viewpoint and crisp pre-rendered graphics of RollerCoaster Tycoon have become iconic over the years, but why did you make these choices when 3D was already starting to be used in other games?

Working on the PC in those days you were constantly up against limits of power and memory, or indeed graphical resolution or detail. Frame rate was my top priority, which for the graphics meant pre-rendered bitmaps rather than 3D rendering on the fly, but with the downside that the viewpoint was fixed and vast numbers of bitmaps were needed, taking up valuable memory (and Simon Foster's time while rendering out all the thousands of bitmaps for each object). Some restrictions led to what turned out to be very useful decisions, like the decision to use an 8 bit "byte per pixel" fixed palette display mode because it's quicker to access and process than full palette display modes — this led to some very useful on-the-fly bitmap palette processing algorithms to allow objects in the game to be displayed in a wide variety of different colour schemes with virtually no extra processing overhead. A lot of time and effort went into designing and writing the code for displaying the isometric gameworld — this was actually a really fun part of the development process for me, and very satisfying when it all worked.

How did the graphical style of the game come about?

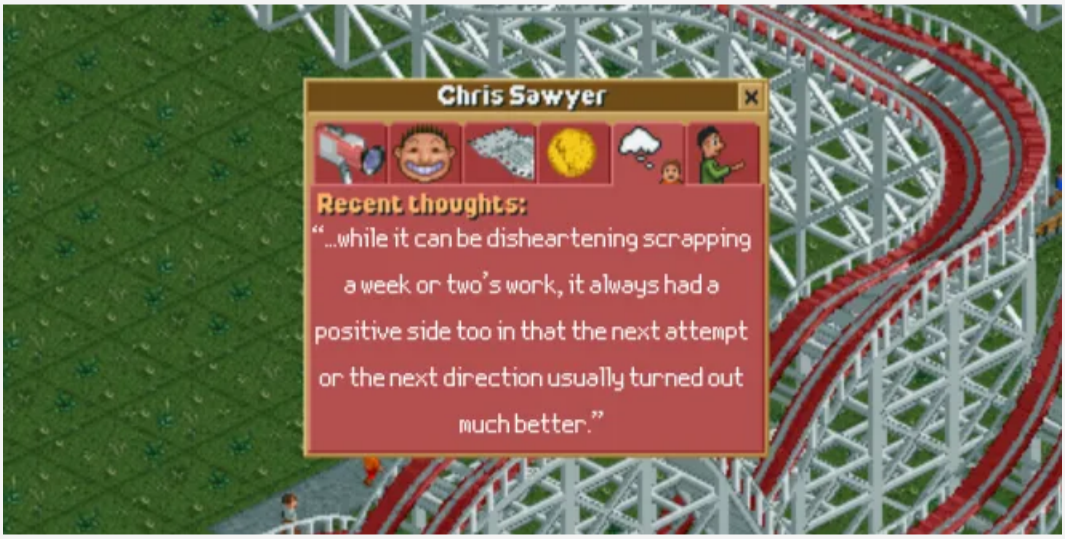
The graphical style evolved quite organically as Simon adapted to working with 3D modelling software and experimented with various different 3D rendering packages to create the pre-rendered bitmaps. My focus initially was just to ensure that the isometric viewing angles were correct and consistent, and that the bitmaps were lit consistently so they looked bright and colourful but showed their detail and shape well. I remember there being a lot of trial and error on Simon's part, trying to get something that just looked "right" in the game — I remember his frustrations with my constant requests for tiny changes and re-renders! The graphical style of the guests also ended up the way it was just through trial and error, and it's amazing how much character they seem to have despite only being a few pixels wide.

Where did the sound effects in the game come from?

There's quite a mix of sounds in the game — some were licensed from commercially-available sound collections but there were very few that were usable for things like rollercoasters and screams. I did a bit of audio recording myself on trips to Blackpool Pleasure Beach and Lightwater Valley but despite using decent digital recording equipment the results were a bit disappointing and only a couple of samples ended up being used. I had some rollercoaster club trip videos from a friend, David Ellis, which despite being nothing more than home movies on VHS tape had the occasional clips of rides with the kind of sounds I needed, so after arranging licensing with David I sampled everything useful from the VHS tapes and sent the audio files to Allister Brimble to see what he could do with them. Allister managed to clean up the recordings and performed some magic to make them "loop" smoothly, so we ended up with quite a few different genuine rollercoaster running sounds and lift chain sounds and so on, many of which have now become iconic.

Did the game's design evolve over time or was there a master plan?

There never really was an overall plan, so each step forward with the game's development was often just through trial and error, seeing what worked and how well it worked, and then possibly going back and trying again or even going back several stages to redo something else. Without the pressure of a contract it meant I could afford to take wrong turnings, and while it can be disheartening scrapping a week or two's work it always had a positive side too in that the next attempt or the next direction usually turned out much better. Towards the end of the project I did have to plan a bit more and have a better grasp of what the finished game should contain, and that was quite challenging.



"...while it can be disheartening scrapping a week or two's work, it always had a positive side too in that the next attempt or the next direction usually turned out much better."

Was it difficult to simulate the rollercoasters in the game realistically?

This was relatively straightforward but led to some interesting decisions. The physics is easy — at any point the game knows the mass of each rollercoaster car and its passengers, and can calculate the vertical and horizontal forces resulting from it depending on the angle of the track it's on, and from that (and friction and any power the train has) it's easy to calculate an acceleration value for the whole train. However it didn't look "right" in the game, so after a bit of trial and error I actually went for a slight visual scaling on the speed and acceleration, exaggerating it a bit. I felt it just looked better that way, and you get a better impression of speed down hills and the changes in speed throughout a ride. It perhaps makes it slightly less realistic, but a better game, and perhaps that is the crux of RollerCoaster Tycoon compared to many of its more modern competitors, that it's primarily a game not a simulation?

How did you go about balancing the game in terms of park profit and construction and running costs?

I have to admit the financial balancing is crude and really isn't all that well balanced — it's very difficult to balance things properly when there are so many variables, and so many different ways of playing the game. The fundamental thing though was that the game should be relatively easy to succeed with if you grasp the concepts of building good rollercoasters and understanding your guests' needs, so everything was adjusted by trial and error until this worked. I did a lot of testing while building the park scenarios which came with the game, and it was a case of tweaking things here and there in the code while doing this to ensure the financial balance was reasonably ok. If anything the game is too easy, as I didn't want it to become too much about business strategies and so on — I wanted it to be about having fun building parks and rollercoasters that entertain guests.

What were your feelings about the game when it was ready to be published?

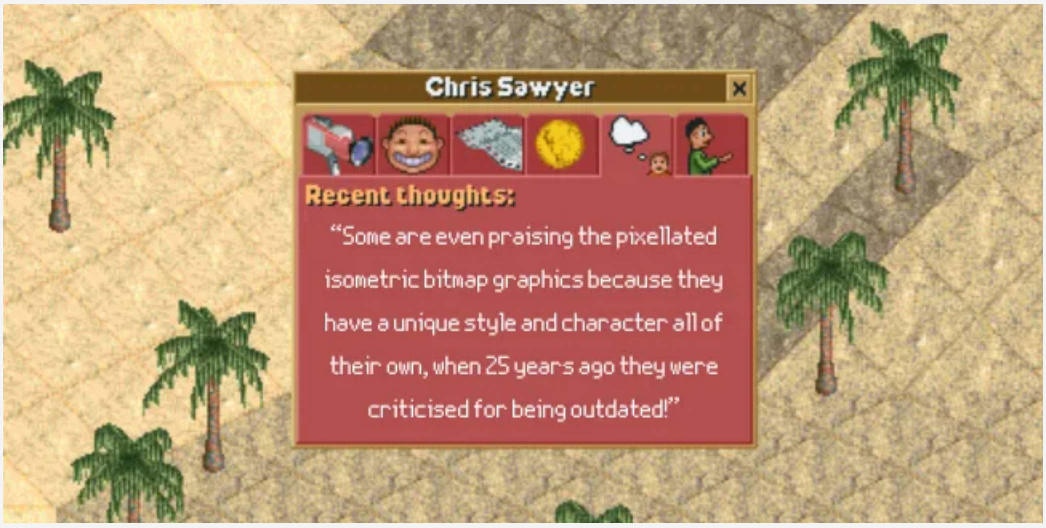
I was quite proud of the game at release time — I personally felt I'd achieved something worthwhile with the game, and I knew from the few testers who'd seen the game early on that they were really enjoying playing it. However the wider feeling was that it was always going to be a niche product, much more so than Transport Tycoon, and might not take off sales-wise at all. In fact I don't think sales did take off initially — it was quite a gradual climb in sales before the game hit the big time.

Did the way players played the game ever surprise you?

What became apparent quite early on is that the game was appealing to such a wide demographic — girls as well as boys, women as well as men, and people of all ages. And players were playing the game in so many different ways — some were just enjoying designing the flowerbeds and footpaths and making sure their guests were enjoying their stroll around the park, while at the opposite end some were pushing the limits of rollercoaster design and construction and creating the most technically amazing and well-themed rides.

What are your thoughts about the series now, 25 years later?

It's fascinating looking back over the RollerCoaster Tycoon series after 25 years and many different versions of the game. While the more modern games aren't my personal favourites they all brought something different and unique to the series — proper 3D viewpoints and "riding the rides", more detail and a different focus. What's really interesting now though is that the latest games are now so advanced, and in some ways so different compared to the original that things have almost come full circle and new players are now appreciating the 25-year-old original games in the form of RollerCoaster Tycoon Classic. Some are even praising the pixelated isometric bitmap graphics because they have a unique style and character all of their own, when 25 years ago they were criticised for being outdated! Perhaps it proves that strategy/simulation games don't need to be all about highly detailed immersive 3D graphics or massive free-form construction — a good game just needs to be challenging, rewarding, and fun.



Recent thoughts:

"Some are even praising the pixellated isometric bitmap graphics because they have a unique style and character all of their own, when 25 years ago they were criticised for being outdated!"