Universidad San Jorge

Escuela de Arquitectura y Tecnología

Grado en Ingeniería Informática

Proyecto Final

<u>Super Toy Cars 2: A brief level design study</u>

Autor del proyecto: Víctor Ramos Menéndez

Directora del proyecto: Eduardo Jiménez Chapresto

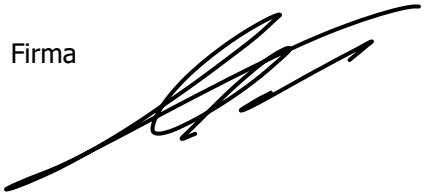
Zaragoza, 09 de septiembre de 2021

universidad
SANJORGE
GRUPO SANVALERO

Este trabajo constituye parte de mi candidatura para la obtención del título de Graduado en Ingeniería Informática por la Universidad San Jorge y no ha sido entregado previamente (o simultáneamente) para la obtención de cualquier otro título.

Este documento es el resultado de mi propio trabajo, excepto donde de otra manera esté indicado y referido.

Doy mi consentimiento para que se archive este trabajo en la biblioteca universitaria de Universidad San Jorge, donde se puede facilitar su consulta.

Firma  Fecha **9 de septiembre, 2021**

Index

Resumen

Este proyecto recoge la información empleada para el desarrollo de niveles en el videojuego *Super Toy Cars 2*, desarrollado por Eclipse Games.

El objetivo de este proyecto es poner en contexto todas las decisiones tomadas a la hora de generar los distintos niveles de una determinada manera, además de por supuesto generar los propios circuitos.

Partimos de la descripción del rol de un diseñador de niveles, y analizamos la evolución de los juegos de carreras a lo largo de los años, para crear un marco desde el que poder posicionarnos para analizar el trabajo realizado. Para finalizar, se detalla el modo de trabajo empleado durante este periodo de tiempo.

El resultado es un set de 5 circuitos de carreras, que forman parte del total de 16 pistas publicadas a la salida del juego.

Abstract

This project gathers the information used in order to craft levels in for the video game *Super Toy Cars 2*, developed by Eclipse Games.

The aim of this project is to contextualize every decision taken when creating the different levels, apart from actually creating those levels at the same time.

We start by describing what a level designer is, and we analyze the evolution of the racing genre throughout the years, in order to create a window that allows us to evaluate the work done. For closure, we get into how things have been done during this period of time.

The result is a set of 5 different circuits, that were published as part of the 16 official levels when the game was ready.

1. Introduction

The idea behind this project is to take part in the development of a video game from start to finish, and learn about the comings and goings of the level designer position, which is one I aspire to fill someday. As the video game had already been started I entered the production when most systems of the game were already complete, being able to focus on my primary role on the team.

Eclipse Games' *Super Toy Cars 2* is the video game in question. It is a racing game, for 1 or 2 players, that uses toy cars and day-to-day environments that feel oversized in comparison. I was in charge of building several levels from scratch and QA for all those levels. My work during this period of time has consisted on the creation of five different circuits in two different environments and assuring the correct interaction among all of its systems.

1.1. The role of the level designer

Consider Rudolf Kremer's definition of level design: "This is a basic purpose of level design, to interpret the game rules, and to translate them into a construct (a level) that facilitates play." (Kremers, Rudolph. *Level Design: Concept, Theory, and Practice*. Wellesley, MA: A.K. Peters, 2009.). Another way of expressing this is by stating that 'level design is applied game design'.

A level is each one of the pieces that, put together, form a particular video game experience. As well as all of them together should form an experience and a story, when separated they should stay functional and coherent. Level design is about creating each of those spaces that the player is going to encounter. It is also about how those players are going to interact with the space, since levels are ideally engaging and interesting. A level designer must perfectly understand the game's gameplay, since the level must be designed around it. Moreover, each level has to be both a test of skill and remain interesting to the player at the same time.

Organizing a game into levels serves both the designer and the player. Workload is more manageable this way, and a more isolated experience allows for the developer to iterate in smaller pieces of information. Each of the levels generate a series of dinamics for the player, that can be experienced again, worked on, and perfectionated.

A level deesigner also doubles as a guide, in the sense that every asset placed on a level serves a purpose of guiding the player towards the final goal. Objectives are highlighted by using composition, contrast, movement, light, landmarks, rewards, and guiding props.

1.1.1. Environmental storytelling

One of the most important elements to have in mind when designing a level is environmental storytelling. This literally means 'telling a story through the use of the environment'. This

is8done by planning and minutely organizing a curated selection of the available objects so that the player can infer a story while experiencing the level.

Instead of explicitely describing events, what we achieve by the use of enviromental storytelling is letting the player fill in the gaps and create their own story, a sort of emergent narrative. The kind of things we have to ask ourselves as level designers are how and when does the story happen. There can also be a set of particular conditions, such as weather, social relationships, and the general tone of the game. There are almost infinite examples of this, both in video games and in real life. The key thing to keep in mind is thinking about how people have interacted with the space, and what consequences does that have on it. For instance, if there has been a birthday party happening at a restaurant, the player can probably see a couple of tables together, party food, and party hats.

Figure 1: Environmental storytelling

1.1.2. Game situations and level beat

By game situations we understand the smallest amount of differentiated gameplay. In a beat them up, game situations would be each of the waves that the player has to overcome. Gameplay stays the same throughout the situations, but the experience changes due to the intended design.

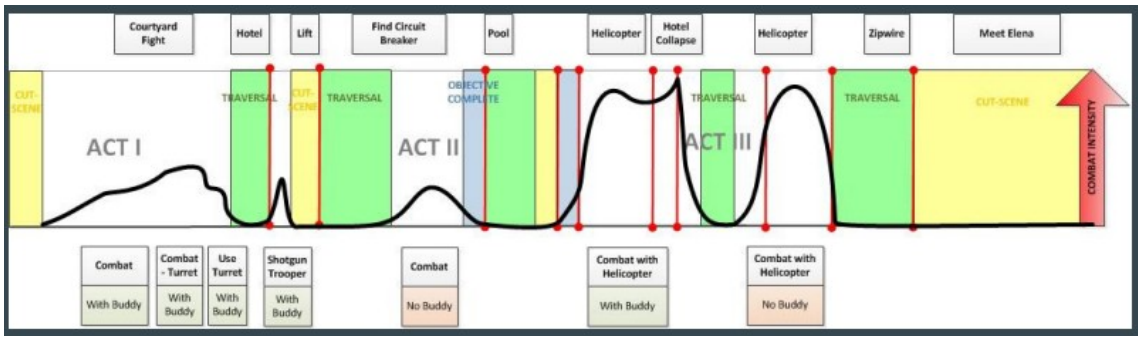


Figure 2: Level beat (Uncharted)

The level beat is the sum of all the different game situations that a level encompasses. This creates a graphic that shows the level's rhythm. Ideally , levels should have situations distributed forming ups and downs.

1.1.3. Pipeline

When creating a level, certain steps have to be followed in order to make sure that every player's experience is the closest to what has been initially planned. The first step is selecting the central theme that the level is going to revolve around.

After that, the team should gather references, ideally creating a sort of moodboard, ideally physical, but it can also be made using Trello or any similar software. There, different pictures can be added for inspiration.

A graph should be the next step, detailing navegavility and the core content of the level. By then, the first initial sketches should start showing up. A narrative should be established at this point of the development. Landmarks can be placed, and the core game situations have to be placed.

The next step is jumping into the software of choice and make a rough layout, that shows the basic flow and that fills a coherent space, Proportions, verticality, limits, and key spaces should start to be detailed during this step.

Greyboxing/Whiteboxing, or commonly known as blocking, is the next thing to tackle. This should be the point at which the duration of the level is adjusted. Level flow and pacing can both be tested from this point onwards.

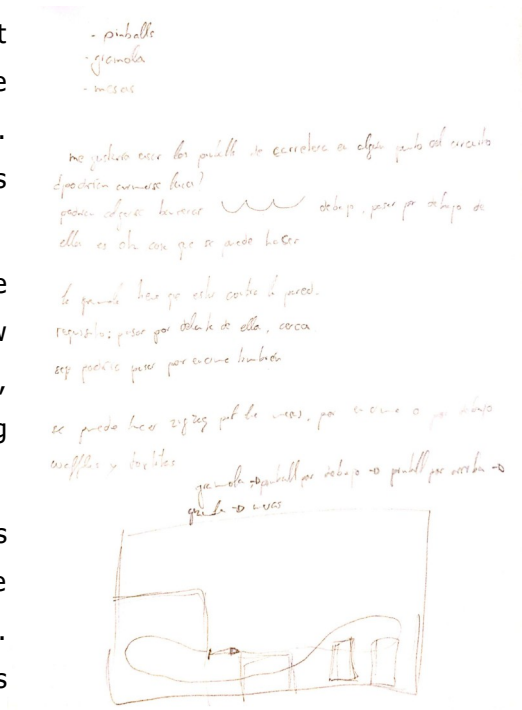


Figure 3: Initial planning plus sketch

Finally, iteration is king. Different assets will be produced for the level, and all have to be placed, tested and adjusted until the experience feels just like it is meant to.

2. State of the Art

Jay Wilbur and John Romero have described level design as "where the rubber hits the road" and "the main responsible for the implementation of the gameplay in a title". The figure of the *designer* existed before it was used to talk about it in the video game frame. In most early titles, level design was often made by the same person designing gameplay (or even making the entire game). Nowadays there are more titles being developed by multiple people, making a figure in charge of level design needful. This led to the conception of the *level designer* role. But what is the role of a level designer for racing games, one of the genres commonly regarded as "more realistic" or that even recreates existing circuits on a 1:1 scale?

2.1. A brief history on level design for racing games.

Being this genre the primary focus of this project, we are going to go through several of its most important titles, covering their evolution, making emphasis on those which generated technical or gameplay advances. Understanding the predecessors and how the medium has evolved during its developing stage will allow us to prepare the ground for a posterior analysis of *Super Toy Cars 2*'s design.

After observing and analyzing the evolution of the racing video game genre, three branches have become apparent to us, being one an exception to the rule.

Figure 4: 1959 Mini Drive, a racing mechanical arcade

2.1.1. The arcade years (1974-1983).

Yes, racing video games have been around since the beginning of the arcades. Even before that time, department stores used to have mechanical versions that will eventually turn into arcades, like the 1959 *Mini Drive.*

This first era of racing games is characterized by its fast evolution and emphasis on controls, which are one of the most recognizable features of this genre. The steering wheel controller has been an essential element of this genre since the start, and it is present from the very first game on this list to the last. Design is not in mind during this first era, but the technical advances that took part during these first years cannot be overlooked and are necessary for context.

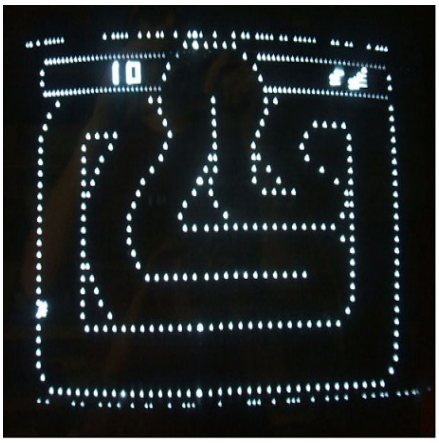


Figure 5: Gran Trak 10 (1974)

Atari's *Gran Track 10*, the first of the racing arcade games, appears during 1974. It featured a 2D circuit viewed from the top and a car, and the aim of the game was to run against the clock, accumulating as much points as possible. Just a year later the same company launches *Hi-Way*, which reuses the top-down view but introduces one of the fundamental features of arcade racing games, the scrolling highway. It also includes a basic AI.

During 1976, Atari launched *Night Driver*, the first ever racing game to use a first-person view, which still holds its status as a classic. The screen shows two lines of white dots, being the road the space between them. The goal, of course, is to keep the vehicle within those lines. It was a step back in graphics, but a step forward in immersion. But SEGA's emergence also happened during this decade, and they hit the spot with *Grand Prix*, a high-speed game with an also faster AI. By 1981, they also released *Turbo*, which uses sprite scaling with a vanishing point, adding a more realistic perspective to the experience. This sprite scaling technology is going to be perfected and used by Namco for *Pole Position* (1983), which used a "Super Scaler" arcade board. It also is the first racing game to take a real-world track and use it, as it follows the trajectory of the famous *Fuji Speedway*, spawning the idea of video games as a simulation of the real driving experience. This same year, EPYX *Pitstop* and Tatsumi-Namco-Atari *TX-1* are also launched, and they both also bring realism to the table. The first one, by using pitstops, introducing a strategic element to the race, and the second one being stricter with its braking and gearing system.

Figure 6: Pole Position (1983)

By the end of this period it is clear that realism is what video game companies were aiming for. There is not much level design to discuss during this period, since it was not a priority at the time. Except perhaps for *Pole Position*, which had to adapt a real-world track to the screen, in

every other case the track was still just used as a canvas and was left as secondary. For some authors such as Steven L. Kent, the golden age of arcade games also comes to an end by 1983.

2.1.2. Genre identity and differentiation (1984-1997).

At the very first year of this period we can already see two currents forming; those games that aim for a totally realistic experience, and those that would rather enjoy the simple, playful feel from the arcade titles. *REVS* and *Buggy Boy* are the games that start this period, the first one being described as a simulation, while the second one was used as the introduction of karts to the video game industry.

Figure 7: Revs (1984)

Most of *REVS*'s development time was used gathering data and developing an accurate physics model, so the game would only feature the Silverstone circuit once it came out. Since then, closing the gap between reality and simulation has been one of the main focus for racing video game developers. Many games over the next few years worked over this concept, adding new features or different gameplay mechanics. *Turbo Esprit* from 1986 was so successful that was eventually licensed by Lotus. Just a year later, *Test Drive* was released. We are not going to look into its first installment though, we are more interested on its second part, *Test Drive II*, which turned the series into a classic. Its subtitle, *The Duel*, is what really catches our eye, since it meant that the player was, for the first time, put against computer-controlled opponents.

Figure 8: Turbo Esprit (1986)

1987 also marks the entry of *Indianapolis 500*. Next to the physics, the game also set high standards in terms of configuration. Its developer, Papyrus Design Group, was fixated in simulating the real Indianapolis 500

Figure 9: Indianapolis 500 (1987)

experience. Their games were so close to reality professional racers would use them in training. Two mechanics were introduced alongside this game: the possibility of customizing the cars, and damage dealt to them (not depicted in the graphics until full destruction).

Figure 10: Super Mario Kart (1992)

It is not until 1992 that we get another attempt at the karting branch of the racing video games. Nintendo's *Super Mario Kart* was the first of many deliveries that this franchise was eventually going to have, and it became an instant hit. It looked similar to *Buggy Boy* but being made for the Super Nintendo Entertainment System meant that it could benefit from its superior computing power (*Buggy Boy* was created for the C64). This game was fast, colorful, and included an idea that was introduced back in the 80s by *RC-Pro-Am*, the integration of weapons into the cars. Nintendo's IP refurbished the idea into different fantasy power-ups, making it safe for the whole family.

Two years later (in 1994), Papyrus releases *Nascar Racing*, proving that simulation should never be taken to the extremes. They took the decision of making the game as close to reality as possible, including the racing times. Each of the races took the same time to run as the real track would do, making the game long and slow. As a positive point, the game modeled damage and also included pit-stops.

Figure 11: Nascar Racing (1994)

From this point onward, innovation was mostly halted. The 90s produced only two more important simulation titles: *F1* and *Gran Turismo*. The first one was entirely focused on Formula 1 racing, and the second one turned out to set a new standard for the genre while also becoming the most profitable *PlayStation* game. Both became franchises, and their last iterations on the formula are going to also be present later at this chapter.

2.1.3. Yu Suzuki and Out Run (1986).

Figure 12: Out Run (1986)

This epigraph is dedicated to a single title, *Out Run*, that generated a third branch for the racing genre. This branch is a fusion between the physically modeled simulations and fantasy style kart games.

Yu Suzuki's first project for SEGA was *Hang Out*, a motor bike simulation. His idea was to approach it from the technical perspective, using a "torsion bar" technology. This, in 1985, gave birth to the motor bike arcade cabinet which included an actual piece of equipment the player could ride. This project being such a success made SEGA have high hopes for Suzuki and wanted him to do the same thing for car racing games (after all, his ideas for motor bike games had revolutionized how their controllers were made). This time, they also wanted to do things differently and take more time.

Yu Suzuki was sent on a two-week tour through Europe. His objective was to cruise through Europe in order to understand what was so fun about driving. When he returned, he had plenty of new ideas, and wrote one of the games that would remain an influence for years.

There are three main reasons why *Out Run* was such a success. One of its main goals was for the player to feel superior. In a game which is just about driving, that translates into being inside of the most luxurious cars. This is why the Ferrari Testarossa was chosen as the lead vehicle. Secondly, *Out Run* was one of the first games to feature a dedicated in-game soundtrack, and finally, its design made it different that anything created before. *Out Run* moved the action to the open road, leaving circuits aside. It was an "open world" driving

simulation. The idea behind it was to have a surprise behind each bend. It was created to be a relaxing experience, just as the vacation Yu Suzuki had while at Europe.

He went on to design *After Burner*, *Virtua Fighter* and *Shenmue*, which became classics as well. But whenever he gets asked about which project he would go back to, he always mentions *Out Run*.

During this whole period (1974-1997) video games have gone through lots of changes, being one of the bigger ones leaping from 2D to 3D. This is especially important for designers. Including a third dimension, whichever it is, means that there are potentially more mechanics a game could envelope. And more mechanics equals more thought into level design.

2.2. Recent history.

Racing video games have established themselves as a staple throughout every generation. Some franchises have been racing down their tracks since the last years of the 20th century. At the same time, smaller games race by their sides to fight for the spotlight. They all approach the same formula (getting from point A to point B) in different ways, some of them dipping in other genres to make a more complex experience. This mix-and-match makes modern video games a very convoluted medium to classify and analyze, so for our interest, *Super Toy Cars 2*, we are going to keep our focus on the three main sub-genres.

These genres are sim racing, kart racers and arcade.

2.2.1. Sim racing.

Simulated racing or racing simulation is probably the easiest of the three to explain. Simulation in video games is trying to perfectly mimic any activity from real life and adapt it into the video game form. Thus, racing games that fall into this category are those that try to emulate how racing is and feels in real life.

These games can be played with a regular controller or keyboard and mouse if you are on pc, but most hobbyists prefer to race using a wheel and pedals, a controller specifically built for this kind of experience, in order to make it as close to reality as possible.

Figure 1: F1 2021 (2021)

One of the key differences between these games and others from other sub-

genres is the use of real-world variables. Players in sim racing games have to deal with fuel usage, damage, tire wear, and grip, among others. There are several franchises that come into mind when talking about racing simulators.

F1 has been a yearly series since 2000, working under the *EA Sports* banner since 2021. It holds licenses for both the FIA Formula One World Championship and the FIA Formula 2 Championship. This means that it holds the right to have all the circuits that conform this two leagues in-game. Its simulation capabilities is so reliable that due to COVID-19 this last years' Grand Prix was disputed using this software.

Our second giant is *Gran Turismo*, already mentioned before as the highest selling video game franchise under the PlayStation brand. Its appeal comes from its large number of licensed vehicles and the ability to tune performance.

Figure 14: Forza Motorsport (2021)

Forza is the ultimate racing sim for the *Xbox* platform. It is separated into two series, the *Motorsport* and the *Horizon*. While the first one is based on professional tracks, the *Horizon* series features a fictional festival, in the shape of an open world video game.

Honorable mentions include *Collin McRae*, later called *Dirt*, and *Real Racing*, the most popular option for mobile devices.

2.2.2. Kart racers.

Kart racers, kart racing games, or go-kart racing games are those which have simplified driving mechanics, unusual track designs, obstacles, and that include some type of "combat", in the sense that players can impede other players' progression.

More often than not, kart racers feature fictional characters from beloved franchises. From Looney Toons to Spongebob Squarepants, all the way to Toy Story. Vehicles also sport unusual designs.

This category of games is the less competitive of the three, due to its inherent randomness.

Figure 15: Mario Kart 8 Deluxe (2017)

The most well-known franchise of this subgenre is *Mario Kart.* As a spin-off to one off *Nintendo*'s most beloved franchises, it includes most characters that appear on the main Super Mario games,and has started including characters and tracks from other franchises.

In 2019 it jumped into our phones with *Mario Kart Tour*, and its latest addition to the franchise is *Mario Kart Live*, a mixed reality experiment.

SEGA can't be less than Nintendo, so they have also created a racing franchise around its mascot. *Team Sonic Racing*, more interested in the narrative component of the experience, tries to cover this niche.

Figure 16: Team Sonic Racing (2019)

Naughty Dog decided to jump on this trend as well, creating *Crash Team Racing*. This saga is ofte regarded as the most demanding of the three, having a semi-open world, boss battles, and a complex drifting mechanic.

Figure 17: Crash Team Racing Nitro Fueled (2019)

Honorable mentions include *ModNation*, the most customizable kart racer, which lets the player create their own car, character, and circuits. This subgenre is the most populated one due to the fact that most indie racing games fall into this category.

2.2.3. Arcade

Arcade racing games are the more carefree version of the racing sim subgenre. Whereas in racing sim games the driver must reduce their speed significantly to take most turns, arcade-style racing games generally encourage the player to "powerslide" the car to allow the player to keep up their speed by drifting through a turn. For the most part, they simply remove the precision required from the simulation experience and focus on the racing element itself.

Figure 18: Burnout Paradise (2008)

Most of this titles license real cars and leagues, but pose them in more exotic settings. Another differential factor is the music. In sim racing games the player hears the motor, and in kart racers it is typical to have funny sounding tunes. Most songs that accompany arcade racing games are rock songs taken from various artists.

Figure 19: Trackmania Turbo (2016)

Perhaps the most well-known franchises are *Burnout* and *Need for Speed,* both inmediately recognizable by their curated playlist, which helps the player get into the perfect mindset for speed, drifting, and crashes.

Another title that has been gaining traction these last years is *Trackmania*. Its low entry barrier makes it a good contender, and its crazy tracks attract a different variety of players.

2.3. Super Toy Cars

Some of the games mentioned before have served as inspiration for *STC2.* Regardless of he amount of importance any of them have had on this video game, the most direct antecedent is obviously *Super Toy Cars*, published by Eclipse Games in 2014. Also, the first installment of the Super Toy Cars franchise.

STC is listed on the company's website as their second produced title. Being the direct predecessor of *Super Toy Cars 2*, it makes sense that an entire epigraph of this project is dedicated to it.

How does *STC* fit into the racing games subgenre? If sim racers, kart racers, and arcade racers are located each at one of the three vertices of a triangle, *STC* lands right between kart racers and arcade racers. It is focused on speed and fun, includes all kinds of different cars, and there is kart-to-kart combat. Another clue we can take is that Eclipse Games is specialized in arcade games, so naturally games they decide to craft all fall into this category. Its soundtrack follows the usual trend for arcade racers; rock songs that make the player move while driving.

Graphics are too a product of their time. Low-poly and cartoony, even sporting a toon shader on some objects. One of its most iconic environments, the nursery, has been remade for *STC2.*

Figure 20: Nursery track at Super Toy Cars

As a way of saving time, some of its assets have also been upscaled and re-utilized. The robot and the duckling are a prime example of objects that appear on both games.

Overall, *STC* circuits strike me as more grounded, being located at more day-to-day locations, and having its tracks run parallel to the ground itself.

3. Objectives

Super Toy Cars 2 is already more than defined when I join the team. As one of the designers, my role is going to be focused on the designing, making, and dressing of the levels. I agree on making three of them. So my main objective is getting to do the three of them as good as possible.

This objective can be broken down into the three key points given at the time of my proposal. Those are as following:

- Create and polish a racetrack.
- Populate said racetrack using different objects taking into account level design principles.
- Create an exciting experience.

During the development time of this project I created 2 different racetracks (5 if we take into account *mirror* circuits) using 2 different environments; a nursery and an american-style diner. All three levels have been thoughroughly populated, both reusing assets and including exclusive ones.

Members of the team and people who have played the game since then have manifested having an "exciting experience".

This checks all three boxes that were used as markers for our objectives.

4. Methodology

The team I have known directly and worked with on a daily basis is formed by one programmer, one artist, and one designer. Other people or teams have been in charge of music and sound design, and some of the modeling, but it was not necessary for me to meet them since our workflow was completely different.

4.1. Tools

Due to the lack of a physical office, every member from the team has been working from their home. In order to communicate with each other, Skype has been the application of choice, though Discord was debated at least a couple of times.

HackNPlan is the tool we used to organize the work and keep track of the different tasks every member of the group was covering at a certain point. It allows the user to create notes in a board that represent the different tasks each member of the team has to fulfill, as well as the amount of time they are supposed to take. This helps keeping track of the time and shows which areas need more improvement or more work put in them. A revision of each task was done daily as well as weekly, similar as how it is done in agile methodologies such as scrum.

Figure 21 HacknPlan

A version control system is very useful for projects as large as this one. We used Git, and Sourcetree was the softfare used paired with it due to its intuitive graphic interface. Finally, the engine used for making the game is Unity3D. All the tools I could need for level design are included in it, plus the ones that are specific to Super Toy Cars 2 had already been coded by the programmer when I started at my position, and worked just as intended, so there were not many improvements to make.

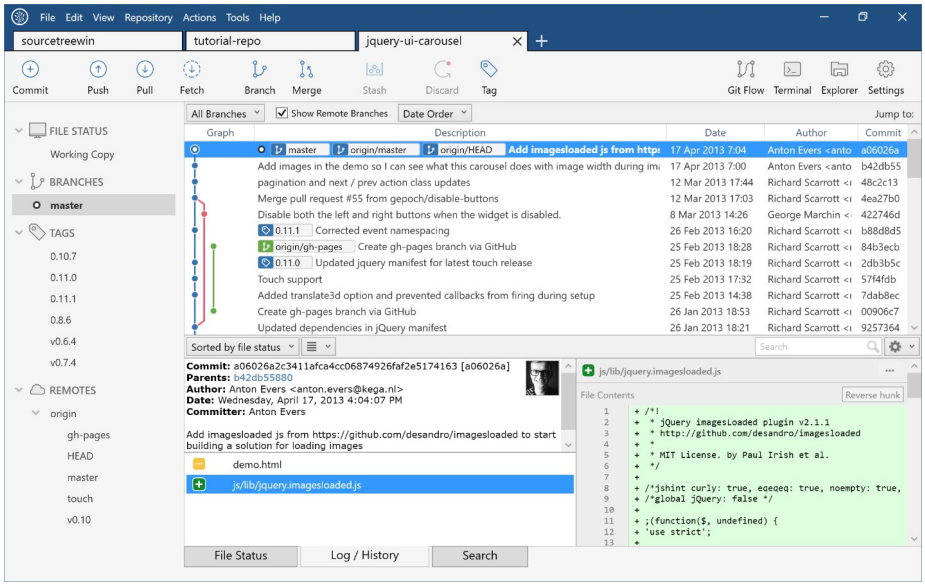


Figure 22: Sourcetree

The most relevant tool for developing *Super Toy Cars 2* has been Unity, the game engine. My role as a level designer has been deeply affected by the various tools within Unity developed by the team, that allowed me to create and work with roads in the most simple of ways.

One that has definitely made my life easier has been the Sidewall Tool. This tool allows me as the designer to place side walls onto the racetracks seamlessly and in perfect consonation with the tracks' tumbles and turns. Placing them one by one would be an umphathomable task.

5. Development

Video games are an iterative process, and *STC2* is no exception. The epigraphs following this introduction are a streamlined version of how these circuits came to life. From turns to lights to bounding boxes, every asset on the game has been placed, tested, moved, removed, placed, tested, and tested again.

5.1. Pre-level designing

Before getting hands on the level design, I had to get familiar with the product, run some races, and ask some questions about what was needed of me and how I could approach my tasks. My first tasks had nothing to do with level design *per se* but knowing about what was done before me made me understand more the aesthetics and feeling of the game.

I ran several times around the different tracks that were already in the game, investing some time working my way through the different menus and pointing out things that didn't seem as intuitive.

5.2. Circuits

When making video games, one of the first principles always used is that "everything has to be as reusable as possible". This saves plenty of work and, if done well, is almost always unnoticed by the players. A great example of this is *Diablo III*, which uses an assortment of obstacles and a great variety of assets in order to create different maps while maintaining the same layout.

Figure 23: Diablo III procedurally generated maps

The way we reused assets and create more content for the game while shaving some time off was by creating a level and then doing a reverse track with that same layout but changing some details along the way. This was precisely my first task: taking Tony's Drive-in and making a second track that follows its structure.

5.2.1. First circuit: Psychobilly freak-out

Being the first task, this was without any doubt the safest we could go about risk versus reward.

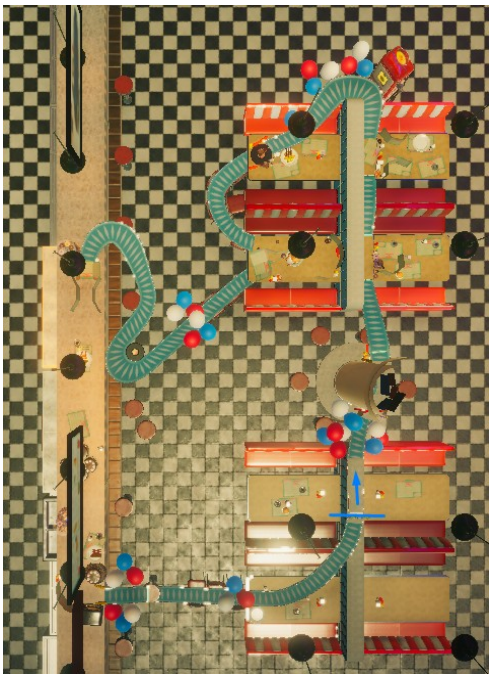


Figure 24: Tony's Diner

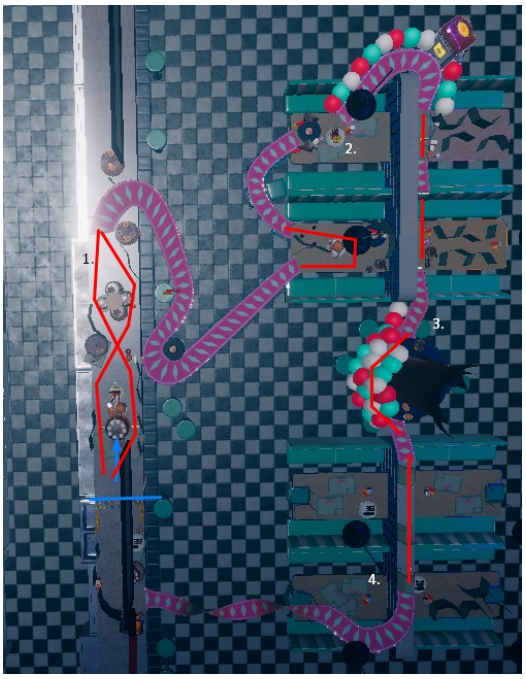


Figure 25: Psychobilly freak-out

Tracks are built using mainly the Road object. Roads are built using a series of connected nodes. Each node is linked to its next node or nodes, forming a circular queue of nodes. Nodes that form the road connect using Bezier curves. This means that each node has 2 handles, one on each side, that adjust both rotation and size of the road. Parts of the road that describe more turns are more dense on nodes, while straight parts use only a couple or three.

Thank to the programmers, the process of making the road go the other way was just a click of a button, so I could focus on creating shapes and curves that would make the circuit flow without having too much else on mind. That way, I decided for the circuit to start atop of the bar. It also happens to be lovated at nighttime.

Figure 26: Starting line

The player then has to describe an eight around 2 different cake stands before arriving at a slope and the first right turn of the circuit. This helped me wrap my head around the fact that a node can have more than two follow-up nodes, and that both follop-up nodes can converge at the same node. This eight-shaped structure allows the player (and the AI) to take different paths to the same destiny. It generates both options and chokepoints, which make the circuit feel more exciting and dynamic.

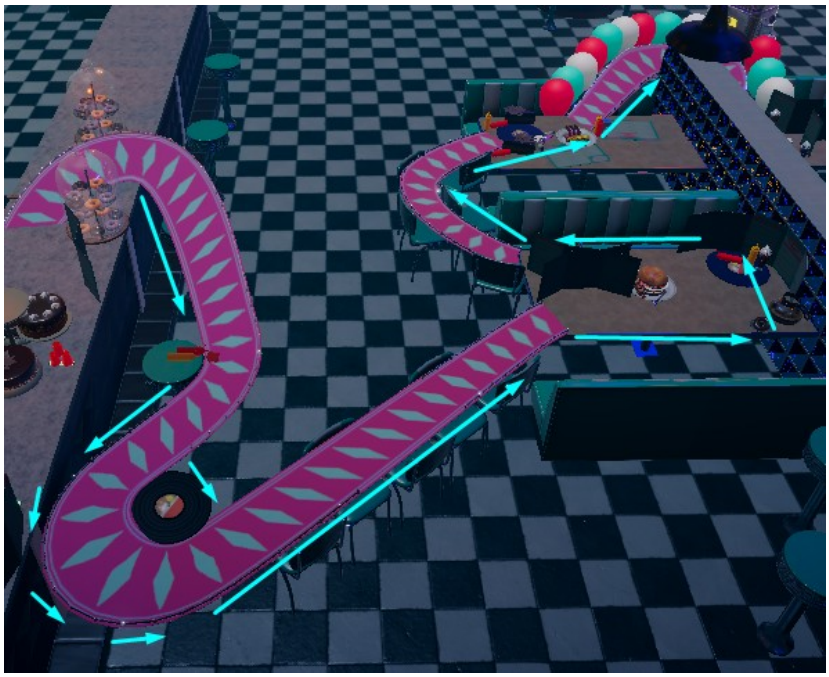


Figure 27: Vinyl turn

After turning right, the player has to go down and left on the first vinyl, go straight and across the table after making a left, and getting up the balloon slope after taking a right between the

dining cubicles.

Figure 28: Turning left and jumping

After a left tight turn right at the gas deposit, players get a long straight and a boosted jump right before describing a slight left turn and venturing up into the balloon tunnel.

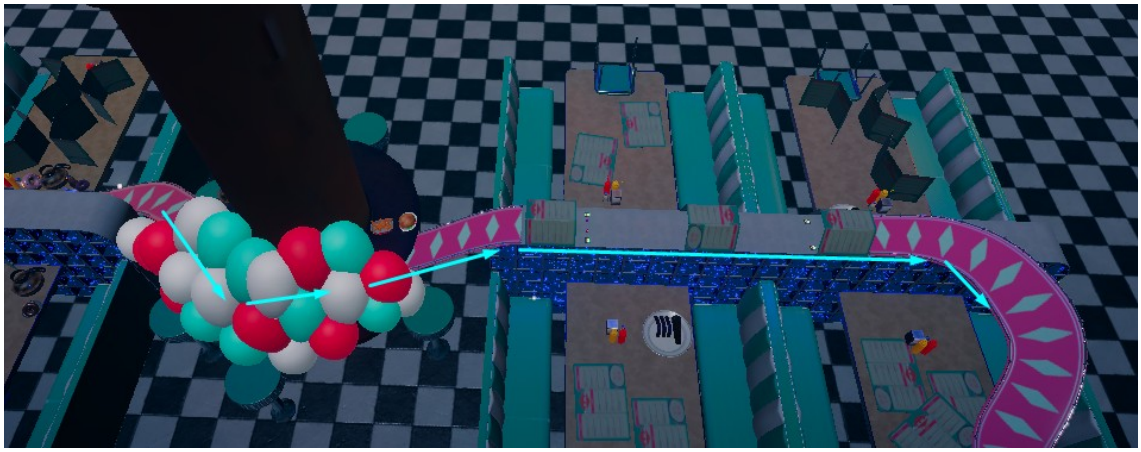


Figure 29: Last straight of the circuit

The balloon tunnel describes a curve around the pillar, and it is followed by the last straight of the circuit. Right as the cars leave the path described by the separator, the road widens and goes down as it describes its last left turn. The player then can clearly see what is left ahead of them; a corkscrew.

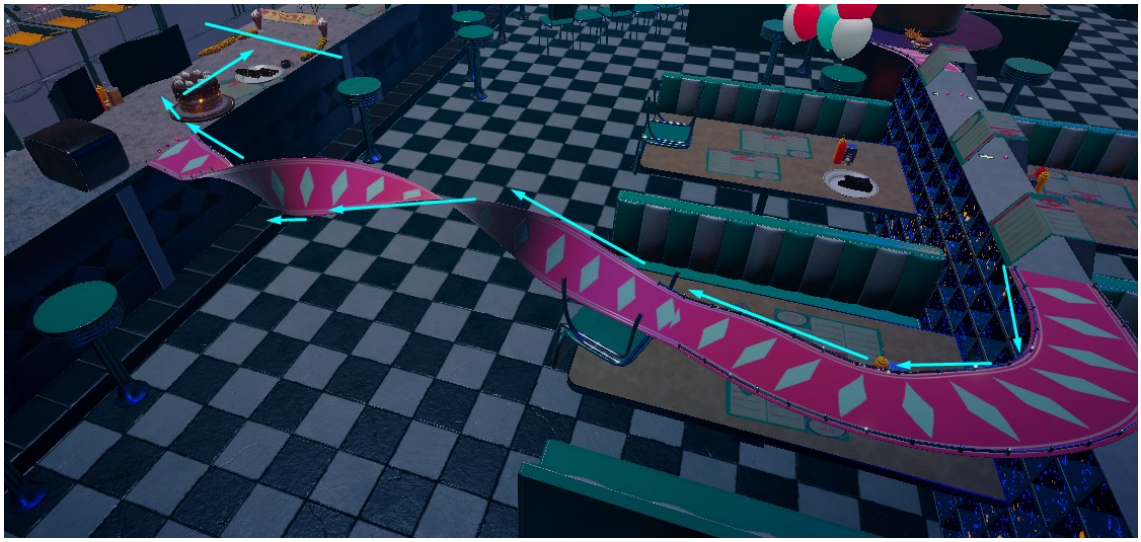


Figure 30: Corckscrew right before the finish line

When playing the game and testing the circuits that were made until this point, the fact that none of them used a corckscrew stood out to me. Thus, I decided to place one on my first attempt at building a track. I clearly did not think this through enough, as it was more difficult to place than I expected and the result is not as good.

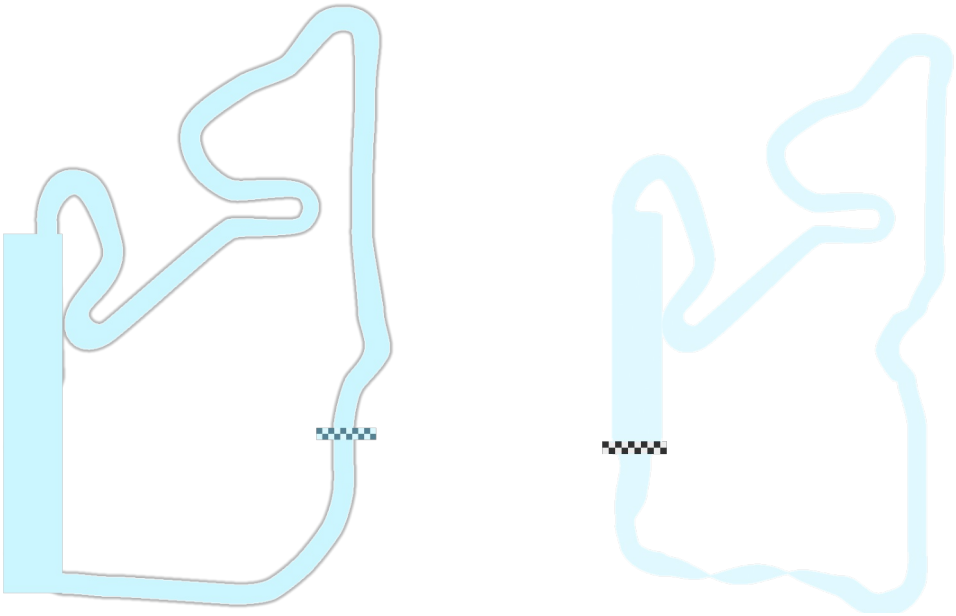


Figure 31: Minimap comparison (Psychobilly freak-out on the right)

The minimap is the 2D representation of the circuit's aereal view. Located at the left-bottom part of the screen, it guides the player through the course. In this case, these also allow us to evaluate how much the circuit has changed from its normal version to its mirror version (the one made by me). There is not much change in its outline, but every slope and every turn had to be adjusted.

5.2.2. Second circuit: Toy Train Station x Spin Top Rampage

The second task I had to tackle was a complete set of circuits, having to make both the original and the reverse version of a couple of circuits located at the nursery. This time I started by taking notes on what the team needed me to do with this one. These had to be different from the set of circuits already based at this location, meaning that it has to be longer and make use of different assets than the pre-existing nursery circuits.

After making these two things clear and seeing what we are left with, we decide to go for the train and the spinning-top asset. They have both appeared before in *Super Toy Cars,* so they are already existing assets ready to place in. I then proceeded to gather information. I revised every *Mario Kart* race track that contains wagons, trams, subway, or any sort of transportation. As for the spinning tops, I did a bit of research about bey-blades in order to see what could be done with them.

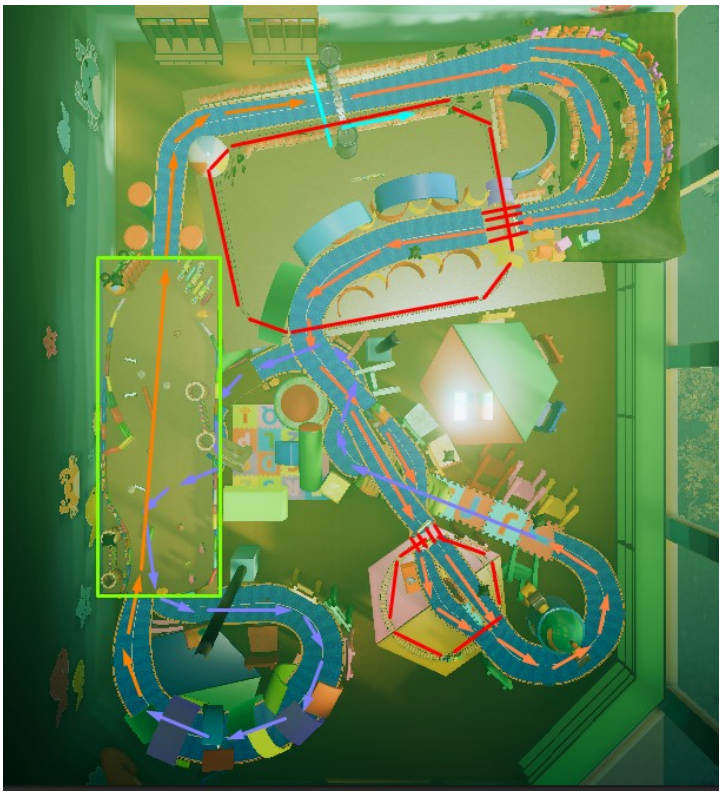


Figure 32: Toy Train Station Layout

I did a bit of writing (included at the annex) to jot down every idea gathered. The train was finally selected to go right besides the track and also across it at some points. Players and AIs can also take a path that goes over it. For the spinning-tops we decided on making an arena and having them travel across it, creating moving obstacles.

Having taken a look at all the different assets, I was also offered to use the crocodile and the planes, which had not been used before. I also ended up resizing a robot and making it an integral part of the circuit.

The personal challenge I wanted to pursue while designing this circuit was including the use of verticality. I wanted I to start atop, go underground, and then come back for the finish line. At Figure 32, orange lines are abover the table and purple lines are those parts of the track that run underneath. Having assets both on the table and on the floor makes it seem more like an actual nursery after kids are done playing but toys have not yet been reorganized.

Two trains describe circles throughout the level. The first one, on top of the table, starts its laps at the same time as the cars, and can run them over after the first turn. Players have a jump and a second train right before describing a turn around the robot. Then the track goes down across the puzzle pieces and into the crocodile's mouth. After describing a couple of turns under the table, it goes up again desribing a wide turn that leads into the spin-top arena (green rectangle). Players then have to evade the obstacles as spinning tops dance around them. After leaving the table and turning right, there will be just another lap left.

Figure 33: Spin Top Rampage Layout

As for *Spin Top Rampage*, this track reverse version, not much changes occur. Cars start inside of the crocodile this time, and jumps are reversed. Turns are modified so they feel comfortable for players when starting them from this other side, and some props are rearranged in order to make it feel a bit more unique.

5.2.3. Third circuit: Nostalgia Road x Cheese Royale Road

Nostalgia Road is built around the whole concept that gives it its name. It takes place during the late hours of the day at the corner of a classic diner. There is a birthday celebration going on at one of the tables. The jukebox won't stop playing, and all the pinball machines are out of place because of the constant pushing and pulling.

Going into my third set of circuits, I decided to try and get the most out of *Super Toy Cars 2* main mechanic, its drifting system. This circuit is purposely full of twists and turns to force the player to drift.

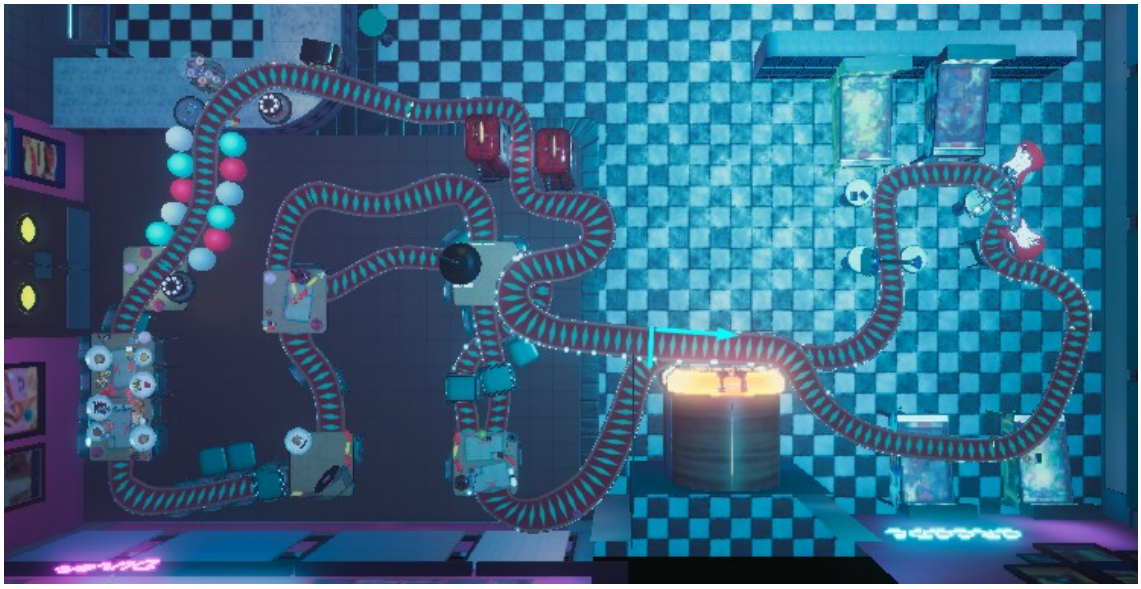


Figure 34: Nostalgia Road layout

The circuit is clearly differentiated into two halves. The first one the player encounters when playing this race is the one on the right. It is designed to contain big turns that, if played correctly, allow the racers to drift and boost themselves for the whole duration of this half.

Figure 35: Nostalgia Road first half

Figure 36: Nostalgia Road second half

The second half, also know as the birthday half, is more kart-oriented. Being the grounds of a celebration, it makes sense that things feels disorganized and all over the place. This half is organized around chicanes, sharp turns that force cars to run a bit slower.

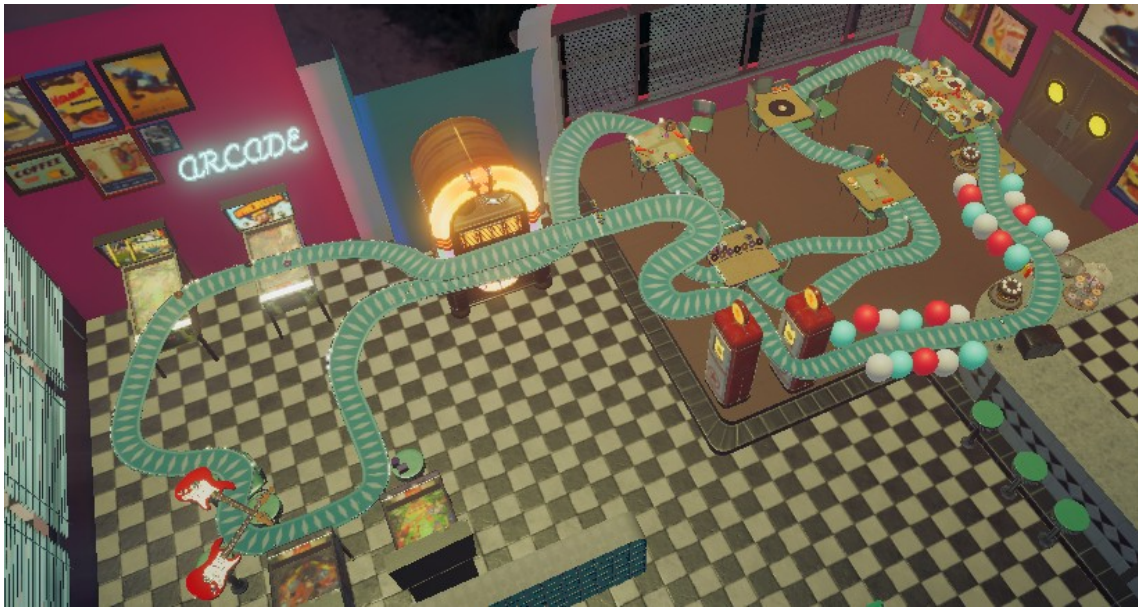


Figure 37: Cheese Royale Road layout

Being a track that features a single jump, its *mirror* version was simple to tackle, and the circuit remains mostly untouched. Turns are carefully adapted to this new way, and the circuit now flows upwards instead of downwards.

5.2.4. About Road, Aitrack, & Logictrack

The core part of any *Super Toy Cars 2* track is the junction of the three elements listed on the title of this epigraph.

The Road, being the main one, is the one that the level designer has to work the most with. It alse generates the other two components of this triad. I have already covered how it works for a bit during the Psychobilly Freakout breakdown.

The Road script generates a mesh with a custom shape mandated by the nodes that conform it. It takes 2 materials; one for the road itself and another one for the sidelines. Sizes for these two elements can be modified too. Nodes can be scaled up or down, so the road also adapts to width changes, and can divide itself in order to create 2 emerging paths. As a designer, I can also decide not to render the road when using some other prop as the road, so it remains physically unexistant for the desired length. Banks can both be made by hand or precalculated with the help of a tool.

This node structure is later used as information for the LogicTrack. The LogicTrack stores the first and last nodes of a road. With them, it assigns an order and direction to the circuit. This way players are able to tell if they are going backwards.

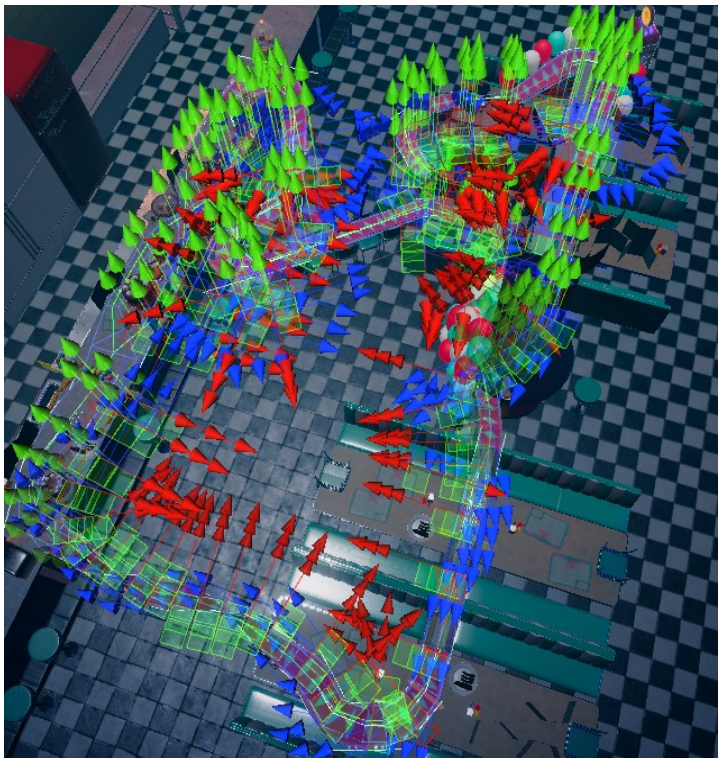


Figure 38: Psychobilly Freak-out LogicTrack

Road's structure is also used to build the Aitrack, used by non-player cars to lap around the circuit. This track is mostly a more defined Road. Psychobilly Rampage's Road is composed of 39 nodes. Its Aitrack is formed by 97 nodes. When creating this track, I had to take into account all variable paths an ai could take. They can either grab boxes, or not, go across speed boosters, or not, and take a path or another depending on the lap.

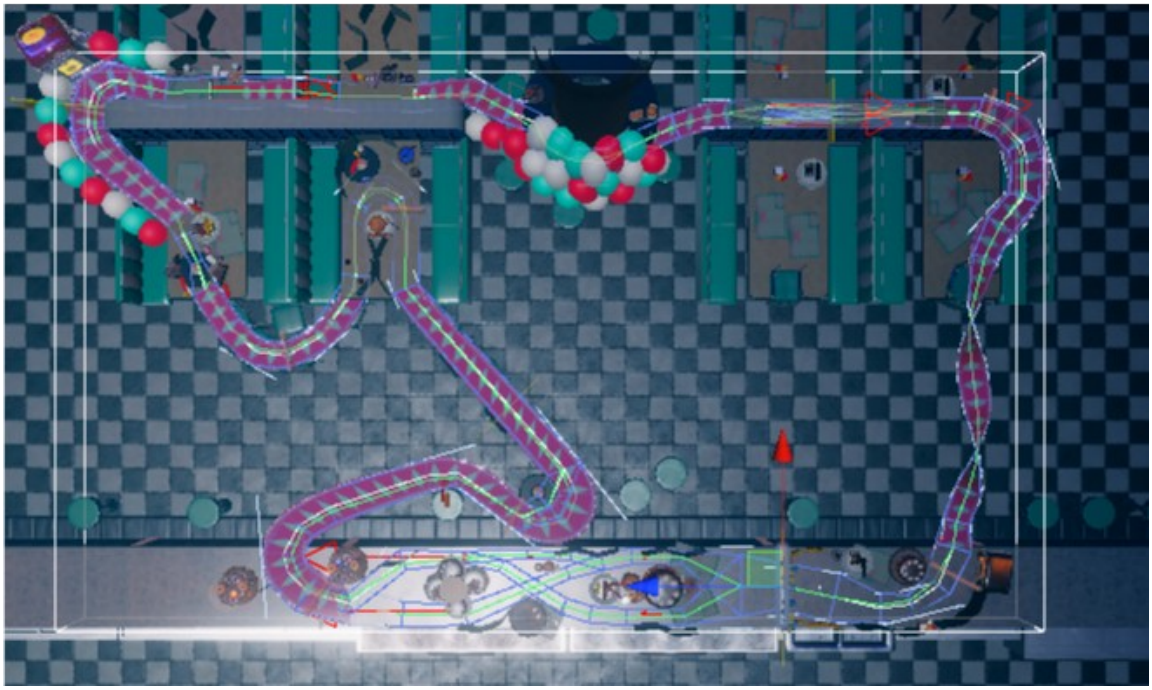


Figure 39: Psychobilly freak-out AiTrack

5.2.5. Miscellanea

This miscellanea epigraph is going to cover all the things that are included in every circuit but are not exclusive of the circuit itself. These objects are power-ups, speed boosters,and sidewalls among others.

Power-ups are contained in boxes that the player can find at different locationis throughout the circuit. There are 8 different power-ups that the player can obtain when passing through one of the boxes. These vary from a shield, health, a speed boost, an eightball, a missile, a trap, a shockwave, and a magnet. Each of them have a different effect, but players are not able to know which one they are getting.

This makes us place boxes in places that may have a use for most items, if not all. For instance, speed boosts are definitely more useful when obtained at the start of a long straight. Certain

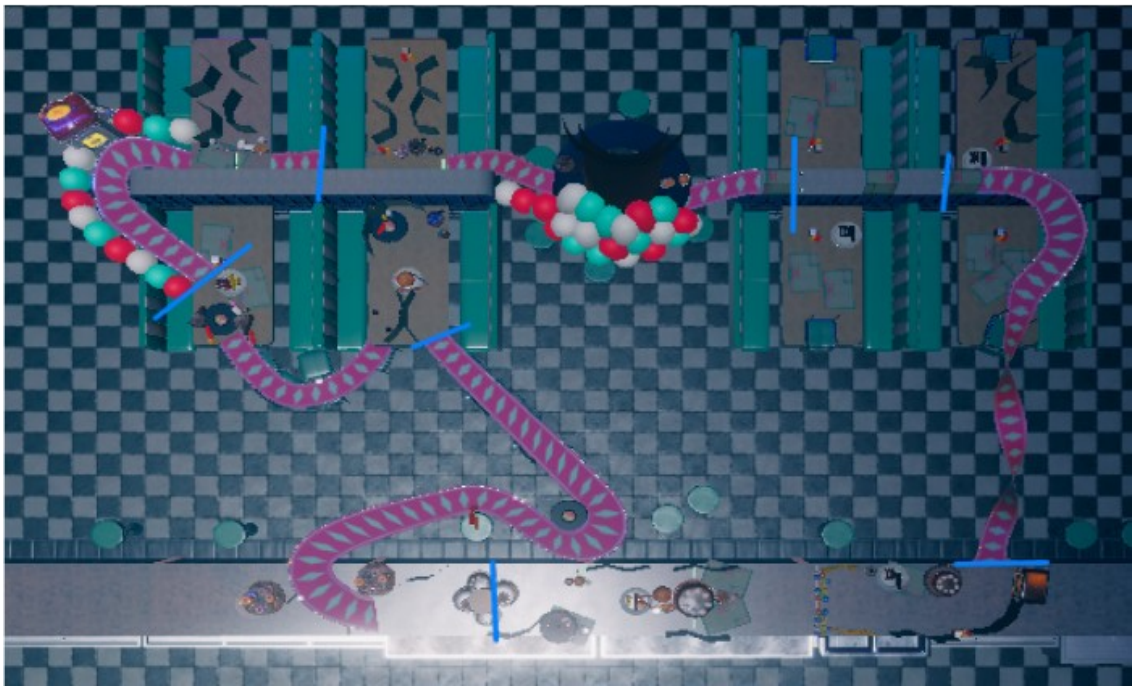


Figure 40: Power-up location at Psychobilly freak-out

boxes serve a double purpose. Maintaining the camera's line of sight centered around the track's next turn or slope is not always possible. Having boxes at the top of slopes or during turns help guide the player towards their goal.

Speed boosters, as their name say, momentarily boost the speed of cars that pass over them. In *Super Toy Cars 2* we have them in three sizes. Wide ones, almost impossible to miss, long ones that give a longer boost, and small ones, which can make the difference between players that are driving toe to toe. They are typically placed on top of jumps, during straights that need more action, and at the outer edges of curves. Speed boosters give a sense of rhythm, and create another layer of complexity by altering the ideal path.

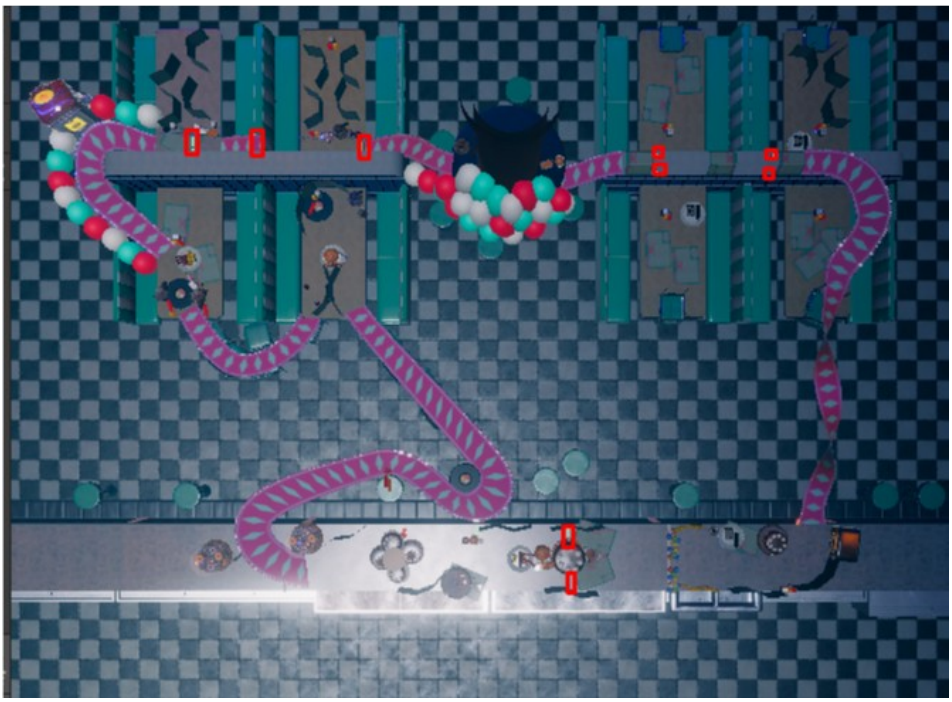


Figure 41: Speed-booster location at Psychobilly freak-out

Sidewall is the name given to the walls which are placed at the sides of the road. Apart from giving a sense of safety to players, and being useful as a bump for particularly complicated turns, sidewalls also give players a vertical reference for the circuit's layout, stating in a clear way how the path continues from the current point.

In *STC2*, sidewalls are placed using a propietary tool, that allows designers to place a defined number of them along one or both sides of the road. The most common places to find them are the outer side of turns, and roads that are too steep or have a change of slope.

Figure 42: Sideline location at Psychobilly freak-out

But what happens when a player manages to leave the road or falls off of it? That is when respawn boxes come in to help. Every circuit is surrounded by this boxes and spheres, that, on hit, make the player respawn on top of the road. For security reasons, the floor beneath the circuit is also fully covered by one. Better safe than sorry.

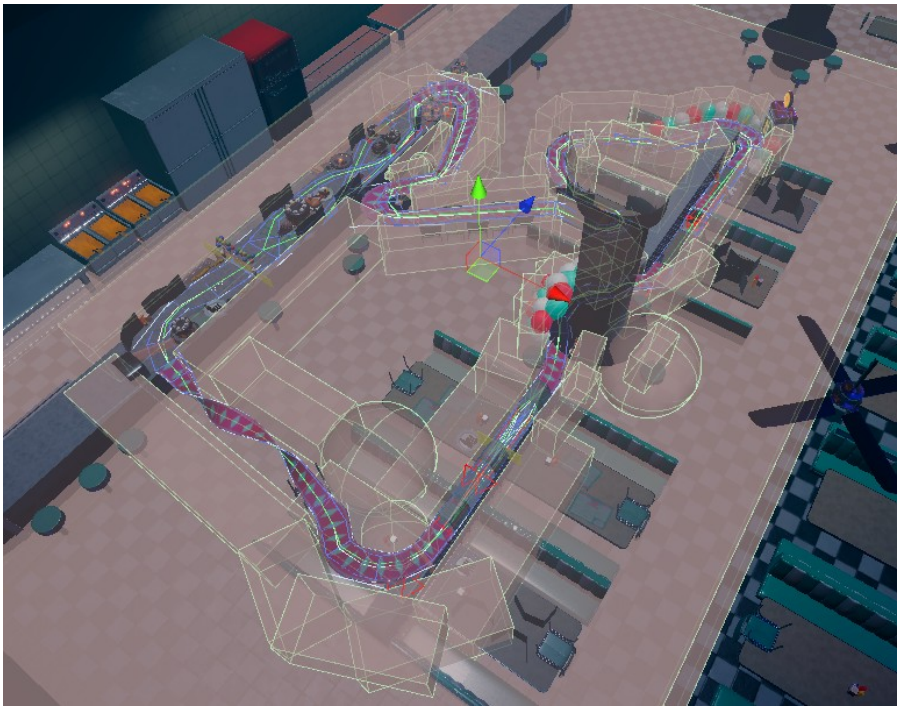


Figure 43: Respawn box location at Psychobilly freak-out

One of the most important things to track at a racing game is the number of laps a player has made around the circuit. There have been instances of player finding different ways to cheat on this checkboxes, such as the famous Grumpy Volcano cheat on *Mario Kart Wii*. In order to avoid this kind of strategy, several boxes are scattered throughout the circuit in a given order. A player that completes a lap must have crossed all of them, since everytime they cross one, both that one an the last one are checked.

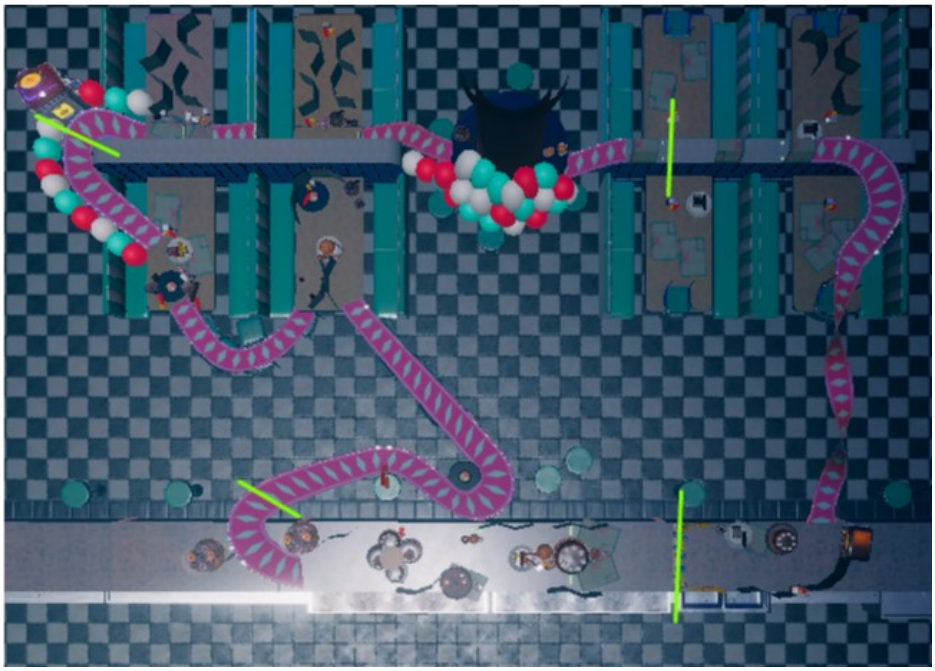


Figure 44: Progress-tracking boxes at Psychobilly freak-out

These are all the extra objects that make an *STC2* circuit work.

6. Economic study

During the development of this project I received a monthly small payment for my job, correspondent to my abilities, workload, and responsabilities. My work on the game allowed for another member of the team to lighten up their workload and dedicate their time to other systems and parts of the game that were also their responsibility.

Working 8 hours/day, this averages at 504 hours worked during the three-month period.

Task	Time (hours)
Planning	48
Building	324
Testing	110
Meetings	22
Total	504

Junior video game designers make around 23k€ per year on average. For that salary and the 30% a company has to pay to the state this sums up a cost of almost 30k€ in one year. A year has 1.780 working hours, so cost/hour is 16.85€. The total cost of my work totals 8.494,3€.

That has to be added to the price of the Unity Pro license, which costs 115€/month. The minimum amount of time you can purchase this license for is a year, which means adding up 1.380€.

The total cost comes up to 9.874,3€.

7. Results

The objective of this work was developing an understanding of the role of a level designer. Being design not a deliverable but an iterative process, it has been relatively easy to analyze which things worked and which things did not.

The result of this work is a collection of circuits that can be infinitely iterated on in order to create different layouts that can generate an interesting and fun experience that adapts to whichever environment is built around it.

Having to rate my satisfaction with the job is a difficult effort. On design tasks, one of the most complicated things is not seeing spots that are not perfect once you come and revisit them. Even so, I am pretty satisfied with my role.

Ultimately, my work helped *STC2* make it it onto PC, Nintendo Switch, and PlayStation4, having physical releases at the two latter platforms.

That being said, perhaps the most important result has been the knowledge I have absorved from developing these levels.

8. Conclussion

The levels built for this project all form part of *Super Toy Cars 2*, an Eclipse Games video game. I have work as part of a team of developers in order to get to publish this video game. My aim has mostly been centered around player experience, making circuits as fun and replayable as possible.

Being a part of this has already have an impact on my life, as it has been an invaluable experience that has taught me a lot of how to work in a professional environment, self-discipline, and of course, general knowledge on video game making.

The work presented here satisfies the objectives set, and has advanced the development of the game.

One thing I still have left to do is create a level all by myself. Knowing how to use all the tools needed for road creation, and being able to create environments and props by myself, getting to it is just a matter of time.

9. Bibliografía

[1] MetalJesusRocks. (7th May 2011). History of Racing Video Games (and subsequent) https://www.youtube.com/watch?v=mtbg5PLH5Es

[2] Masters of Doom, by David Kushner (6th May 2003). https://books.google.com.ar/books?id=ltQ6LYHpmQEC&pg=PT124&lpg=PT124&dq=Donna+Jackson+doom&source=bl&ots=Ob5_qX_J3l&sig=1Fg7mQX091-7tX9AW-10DS3QXnM&hl=es-419&sa=X&ved=0ahUKEwjykoOGh63LAhVEj5AKHdA3BTgQ6AEITjAG#v=onepage&q&f=false

[3] The Ultimate Story of car racing video games. https://www.conceptcarcredit.co.uk/car-racing-video-games-the-ultimate-history/

[4] GameDeveloper, The pure advantage: advanced racing game AI, by Eduardo Jimenez. (3rd February 2009). https://www.gamedeveloper.com/design/the-pure-advantage-advanced-racing-game-ai#comments

[5] Wikipedia, The golden age of arcade video games. https://en.wikipedia.org/wiki/Golden_age_of_arcade_video_games

[6] IGN, Historia visual de los juegos de carreras, by Juan García (30th March 2015). https://es.ign.com/reportaje/92281/feature/historia-visual-de-los-juegos-de-carreras

[7] Eurogamer, Out Ran: meeting Yu Suzuki, by Martin Robinson (22nd March 2015). https://www.eurogamer.net/articles/2015-03-22-out-ran-meeting-yu-suzuki

[8] Gamasutra, The history of Atari: 1971-1977, by Steve Fulton (). https://www.gamasutra.com/view/feature/2000/the_history_of_atari_19711977.php?print=1

[9] Wikipedia, Modo 7. https://es.wikipedia.org/wiki/Modo_7

[10] Vice, An abridged history of segas awesome arcade racers, by Even Hosie (23rd February, 2017). https://www.vice.com/en/article/ezqzp4/an-abridged-history-of-segas-awesome-arcade-racers

[11] Wikipedia, Racing game. https://en.wikipedia.org/wiki/Racing_game

[12] Videojuegos fandom, Videojuego de carreras. https://videojuegos.fandom.com/es/wiki/Videojuego_de_carreras

[13] Eurogamer, Arcade racers are having a moment, by Martin Robinson (1st September 2020). https://www.eurogamer.net/articles/2020-09-01-arcade-racers-are-having-a-bit-of-a-moment

[14] Wikipedia, Kart racing game. https://en.wikipedia.org/wiki/Kart_racing_game

[15] GameElitist, The kart racing genre, by Shola Adedeji (14th June 2012). https://web.archive.org/web/20141211113252/http://www.gamelitist.com/2012/06/the-kart-racing-genre/

[16] CRC Press, An architectural approach to level design, by Christopher W. Totten. ISBN 978-1-4665-8541-6.

[17] Wiley, Level Up!, by Scott Rogers. ISBN 9780470688670.

[18] Eclipse Games website. https://www.eclipse-games.net/

[19] Mario Kart Wii, atajo volcán gruñón. https://www.youtube.com/watch?v=TGZMNS0HV-g

10. Annexes

10.1. Racing games genre evolution.

1959 Mini Drive. Mechanical game.

1974 first arcade racing game Gran Track 10

1975 Hi-Way introduces the recognizable scrolling highway.

1976 Night Driver, first person view

1980 Rally X top-down view

1979 Grand Prix. Extremely fast, competing cars like never seen before

1981 Turbo by SEGA, uses sprite scaling in a way that provides illusion of motion and depth

1983 Pole Position by NAMCO, highest-grossing arcade hit. Perspective and style set the standard for games to come. Perfected Turbo with his "Super Scaler" arcade board. It follows the trajectory of the famous Fuji Speedway, spawning the idea of video game as a simulation of the real driving experience.

1983 EPYX Pitstop adds, as the titles suggests, pit stops to the equation. Slows down the gameplay but adds strategic elements.

1983 TX-1 added more realistic driving characteristics, which forced the player to brake before corners or shift gears at opportune moments.

1984 REVS Shift of paradigm – realistic feeling SIMULATION Geoff Crammond

1984 Buggy Boy for Commodore 64 - KARTS

1985 Test Drive by Accolade (Test Drive II turned the series into a classic)

1986 Out Run by Yu Suzuki (SEGA). – OPEN WORLD (third paradigm)

1989 Indianapolis 500. High standards in terms of configuration. HYPER REALISM. "the company's games were so good that professional racers allegedly used them in training. Right from the start, the ability to customize your car's setup in minute ways set the game apart."

1992 Super Mario Kart. Computing "Mode 7"

1994 Nascar racing. DAMAGE MODELLING + races took the same time as actual Nascar Tracks to complete.

1997 PlayStation Gran Turismo – perfect simulation, photorealistic, complex. NEW CHAPTER

Here onward, refinement, not revolution.

10.2. Notes on Nursery_03

Vital assets for the track.

- Train.

- Spinning-tops.

What can I achieve with them?

Train: moves, following a path shaped with the railings. (Railings can be part of the road, so cars can go on top of them at their own risk). It can be the main theme of this track.

- (PRESENTATION & TAGGING ALONG) Can move besides the road, but taking a little space from the margin, thus, making the track narrower (maneuvering difficulty increased, cars can push each other against the train)

- (SHORTCUT & INTERRUPTION) Can cross through the road, breaking the pace of the race (more useful if the train is divided into different wagons, just as in (1)). It makes the road more dynamic, as it is going to be constantly changing (adds more maneuvers available to the cars, can also be used as a shortcut).

- (JUMPING) Can cross through the road, generating a vertical space that the cars can jump. We can have a bridge that generates a jump right where the railings go through.

Spinning-tops: move randomly, spinning around, making cars bounce off them. Their random-ish movement can be a con, but we can create a controlled environment for them. They are obviously a hazard for the cars.

- (ARENA) One option would be creating a battle arena that they cannot leave (2) Can be used as a roundabout, the same way as the octopus asset.

- (ROTATORY CENTER) The middle part of the arena can be rotated, creating a different effect on spinning-tops and on cars. This way we can have a roundabout without anything at the middle and spinning-tops may collide with each other more this way.

- (INTERRUPTION) Cross through the car's path, making them navigate around them. They can also push each other against the spinning-tops.

Additional thoughts.

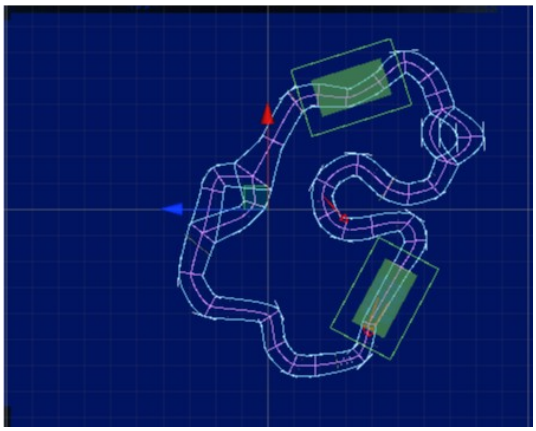
Maybe I can divide the circuit into 3 different areas. An introduction, where the railings are on our side and we can get to see the train. Maybe then having the train cross our path and having to evade it (or follow it to get into the shortcut). After that, a jump to the spinning-top

arena. When you cross the arena, a series of jumps above the railings. You then arrive to the finish line.

I'm also thinking about including the planes, which can go on top of a wagon, and can also be used as a separate hazard, having them come out of spaces signaled by a light. I would have a straight path with holes at the sides, with a light on top of them. When the light is turned on, the plane will come out of the hole, making the player be on guard for it, ready to evade the hazard. (Perhaps to similar to the train).

Playtesting nursery01 and nursery_night. (Steam version).

Nursery01 fastest lap was 33.58 seconds. Nursery_night fastest lap was 38.59 seconds. Maybe I should aim for a longer lap time. The concept that this new circuit is trying to aim for doesn't resemble any of those circuits (apart from the background assets).



There is a lot of extra room I can use to make a longer track, that hits at least the 60 seconds mark on a perfectly driven lap. From the space I see I can maybe make the track twice as long.

Doubts about circuit building.

Mario Kart tracks that use wagons/trams/trains or moving parts as a mechanic

All Mario Kart tracks: https://www.youtube.com/watch?v=w9VdEq1LetM&t=1s

5:40 -> Toad Harbor (tram)

47:42 -> Wario's Gold Mine (wagons)

53:05 -> Ribbon Road (toys move from side to side)

54:26 -> Super Bell Subway (subway, duh)

References.

(1) Gameplay Wario Gold Mine: https://www.youtube.com/watch?v=_pMmSB6YSVc

(2) BeyBlade Arena: https://www.youtube.com/watch?v=8_54kwe7WoI

(3) BeyBlade Arena Rotation: https://www.youtube.com/watch?v=focUXb7dDx0

My circuit planning

Node001 as the start.

First straight: train presentation -> The train comes from the right and stays at the middle of the road. The next 2 times you see the train, you jump over it. The final straight is reserved for spinning-tops.

Red square: same level.

Green square: going down.

Pink square: going up.