



## INDEX

1. Concept .....	2
2. Graphics .....	3
3. Physics.....	5
4. AI.....	6
5. Music.....	7
6. Technology .....	8
7. References.....	9
8. Credits .....	10

## 1. CONCEPT

Karting Garden was not originally going to be exactly what it has turned out to be today.

Almost from the beginning we were clear that we were going to create a kart racing game, with the classic race game structure, but we wanted to give it a special touch and propose an original and striking theme, so at the beginning, Karting Garden had the intention of being Sugar Rally, a racing game of karts in a world of sugar and sweets.

Unfortunately, shortly after we realized that although the idea was well raised and we had different ideas, the visual tests we did didn't turn out as we expected, since the elements and colors could not be appreciated well due to the limitation of space and color palette.

That's when the current Karting Garden was born, a kart racing game set in a friendly and cheerful garden.

We wanted it to have a realistic appearance as much as possible, so we decided to focus on it graphically and physically so that the user could have a pleasant experience both visually and mechanically.

## 2. GRAPHICS

We designed various shapes of curves to be able to create the circuits as different and moldable as possible, obviously within the limitations of space, since we knew that our circuits could not be too big or long.

We took various characteristic elements of real life karting circuits and added the characteristic elements of a real life garden, and this is how Karting Garden arises.

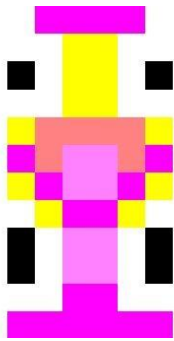
Cones, roads with red and white striped edges and signs appear on practically all circuits, while flowers and trees cannot be missing in a beautiful garden.

Creating the tileset was a bit heavy task, since it had to be continuously updated every time we needed to include a new element and we were lacking of space for tiles, but in the end we managed to fit everything.

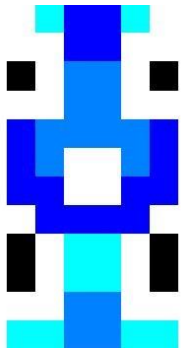
From there, designing the levels/rounds was more or less progressive, although sometimes designing maps that didn't look alike seemed like a complicated task. We also investigated real circuit shapes in order to create our own miniature circuits.



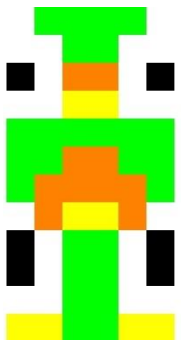
We also designed 4 different racing drivers represented by 4 different main colors, so we have:



“Pink” as the user’s driver



“Blue” as the easy enemy driver



“Green” as the normal enemy driver



“Red” as the difficult enemy driver

### 3. PHYSICS

Talking about the physics, we have tried to get the most similar ones to the reality to let the player feel that he is driving a real car. Our car can turn around it in 12 possible angles, from  $0^\circ$  to  $330^\circ$ .

Moreover, the player can accelerate and brake like a real car, also if the player is driving the car and stops stepping on the gas, a frictional force will act to stop the car progressively.

To calculate the position where the car will move we use two vectors, one for the direction of the wheels and the other for the direction of the car. Then we add this two vectors and the car moves to the new position.

To conclude, we can say that the car has only 4 possible values for the velocity and 3 for braking.

## 4. AI

The design of the AI has been one of the most hard things to think about that and to implement because our desire to create an intelligence that really challenges the player.

To achieve that goal, we used a node system which tells our AI to what position should it move. Once our AI knows where is its next objective, it has to decide if it has to turn the car to some side or if it has to keep going straight.

To do that, we use Manhattan distances which consists of calculating the next position in 3 possible cases, turning left, turning right or keep going straight. Then, decide which of these options will take the AI earlier to the objective position.

Once we decide which way should the AI take, the AI moves.

Moreover, in our game we have different types of enemies, and each of them has a different behaviour and different skills. Each of our enemies have a particular reflexes for turning the car. Also some cars skid more than others.

As we showed before, there are 3 types of enemies:

“Blue”: low reflexes

“Green”: medium reflexes

“Red”: high reflexes

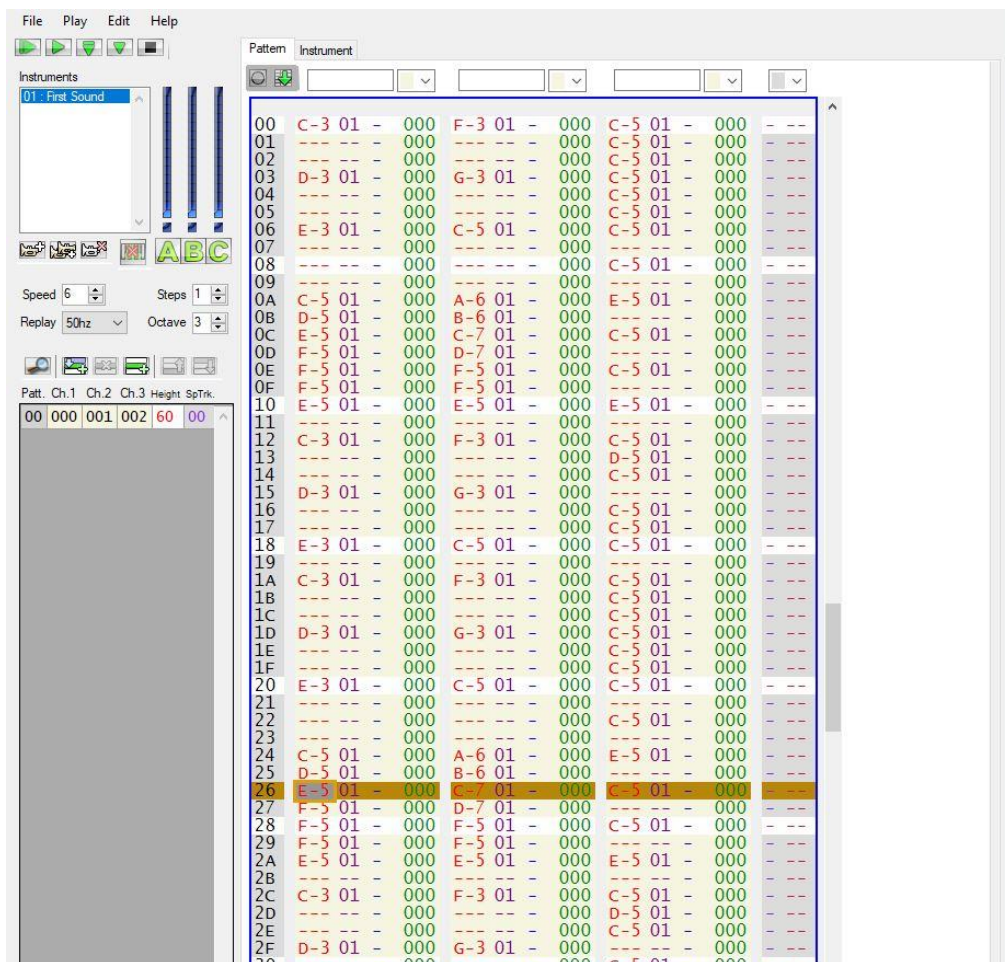
Our difficulty rises in an undulating way. You will face to a few easy-medium enemies and after that, “Red” will appear, and it will be a bigger challenge for you. If you win you will advance to new types of circuits.

# 5. MUSIC

To create the music for our game, the idea was to have a cheerful main melody but not much, a simple and short melody to provoke a fun feeling in the player.

The main theme appears in the menu, but during the game we thought it would be better to let the player focus on steering the kart as well as possible without crashing (which is difficult enough) without having the distraction (or annoyance) of music while playing, which could result in a big headache.

Using arkos tracker 1 we were able to create that main melody along with the win melodies (which goes from low to high) and lose (from high to low).



## 6. TECHNOLOGY

Technology used:

- CPCtelera: Game engine for Amstrad used during development.
- WinAPE: Amstrad CPC emulator.
- GitHub: Development platform for managing the project.
- Sublime Text 3: Text editor for coding.
- Visual Code: Text editor for coding.
- Arkos Tracker 1: Musical software for producing the music.
- Adobe Photoshop: Image editor software for designing and creating sprites, screens etc.
- Tiled: Map editor for creating maps/levels.



## 7. REFERENCES

**Prince Of Persia by Broderbund (1990):** Round 3 has Jaffar as the enemy racer and the Round's number sign decorated with one of Prince Of Persia's wall pattern.



## 8. CREDITS

- **Game authors:**



Adrián Mulero Alarcón

Julia Lozano González

Nuria Carmona Guijosa.