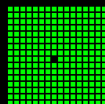


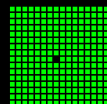
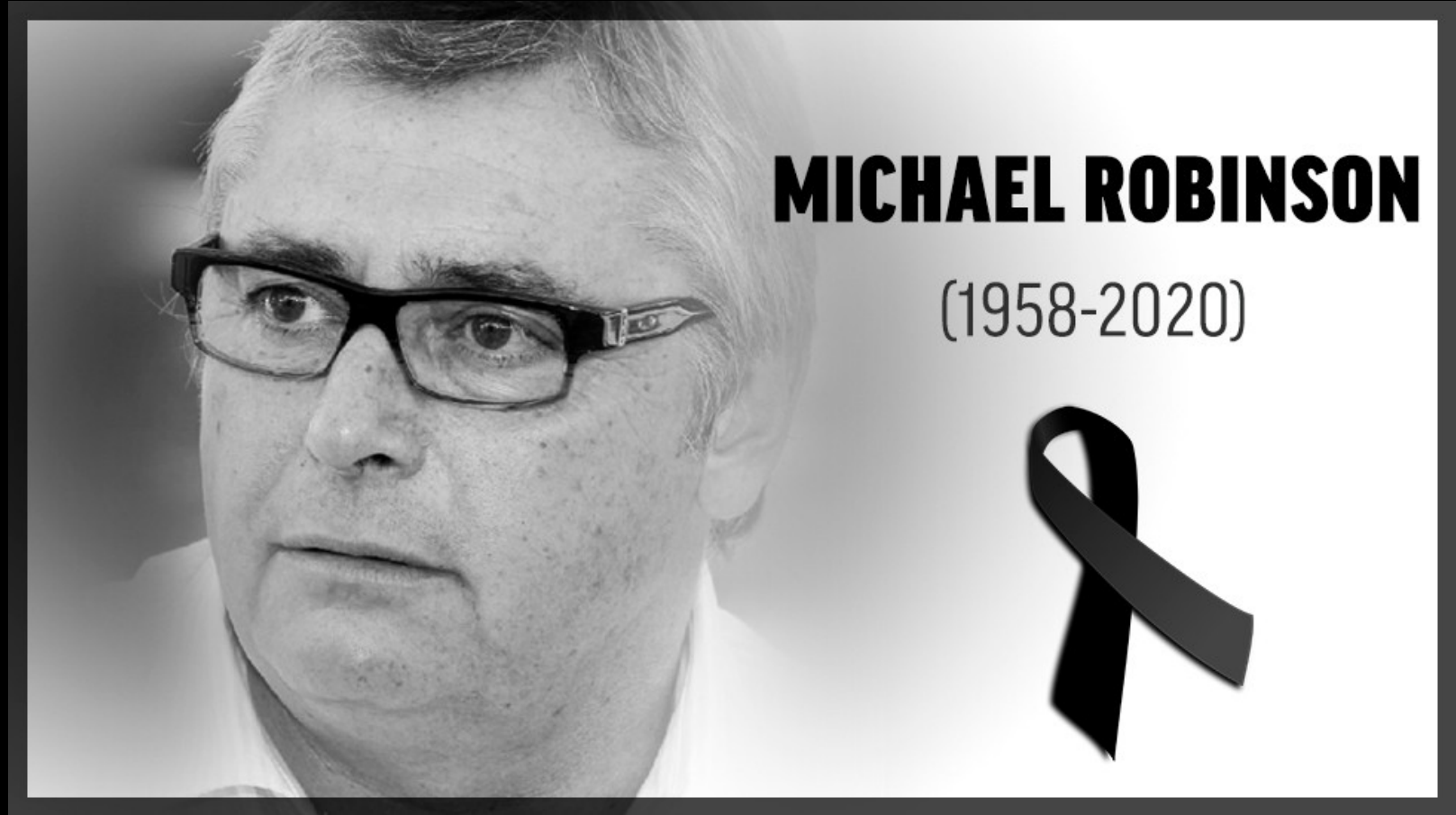
# Hacking on PKF files

My tribute to Michael Robinson and the PC Fútbol Community  
(Mr. Math, SPISE MISU ApS)

2020-08-15 @ BornHack



# My tribute to Michael Robinson (RIP)

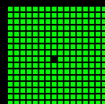


# Overview

- About me (very shortly)
- Background
- PCx-Utills
  - Before and after going public
- Demo (short)

**Note:** Slides are released under the CC BY-SA license

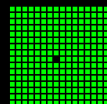
- Creative Commons Attribution-ShareAlike (“copyleft”)



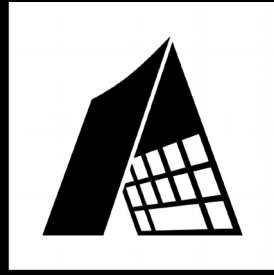
# About me (very shortly)



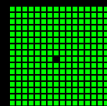
- **Mr. Ramón Soto Mathiesen** (*Spaniard* + *Dane*)
- MSc. Computer Science and minors in Mathematics
- **CompSci** @ SPISE MISU ApS
  - Trying to solve EU GDPR with a scientific approach (<https://uniprocess.org>)
    - Permissive copyleft license (LGPL-3.0)
  - Mostly with **Haskell** and to a lesser extend **Elm** (**PureScript**)
- Member of the Free Software Foundation (FSF) since **November 2007**
- Founder of Meetup F#unctional Copenhageners EST. **November 2013**
- Blog: <http://blog.stermon.com/> (slides under /talks/)



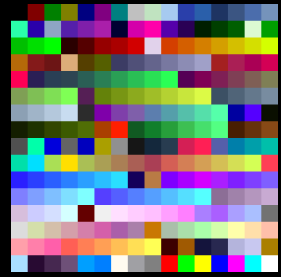
# Matching of expectations



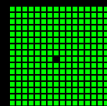
- I will show that when using tools having support for **parser combinators**, **algebraic datatypes** (AST) and **type classes**, can be really useful for projects like this as they can help to **discover patterns** but also to ensure: **simplicity**, **structure** and **correctness**
- **Notice** that all the **discoveries** are made without any **debugger** or **disassembler**. My setup is just good old Emacs (**hexl-mode**) and a lot of **WTF** moments. It's important to understand that I had no knowledge of the file format before doing this project even though I played the games many years ago  
**Remark:** There will be **shown code**, but it's **not necessary** to know **how to code** as it (hopefully) will remind of plain **English**



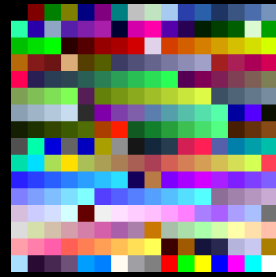
# Background



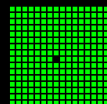
- **Two years ago**, it was the **25th anniversary** of the first edition of PC Fútbol (\*):
  - Pablo Ruiz interview (Spanish)
  - Gaby Ruiz interview (Spanish)
- Thanks to Wine, I was able to install it on my **\*nix** box
- There is a **Community** (PCFutbolManía) that tries to keep the games **up-to-date** with the **newest players** and **teams +25 years later**
  - it takes **a lot of effort**, as it's **mostly a manual process**
  - it takes **about a day to update a team** (there are **+480 teams**)
- (\*) - A football manager game. It's **initial version** was actually **2.0** and it was **sold as a magazine** as the **VAT** was **lower** and therefore resulted in a **cheaper price**



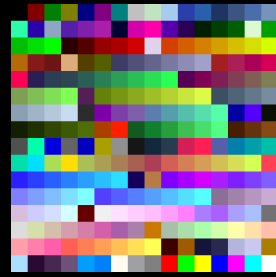
# PCx-Utills (before → ...)



- Main goal is to **automate as much as possible** the task of **updating data**
  - I knew about the **EA Sports** FUT/FIFA API (**free** and **no registration required**)
    - Sadly **it isn't updated anymore**, but still available though
  - Found **alternative** FUTBEST (still **free** and **no registration required**)
    - Some fields aren't populated with **FUT/FIFA** data (maybe only delivered with a **premium** account?)
  - Other goals are:
    - **Open project**, copyleft licensed, that would allow **people to contribute** but mostly **learn** in order to **spawn** other **similar projects** (\*) for the **benefits of the Community**
    - **Edit data** by the **end-users** preferred **text editors** which would **allow most people to use the tools**
- (\*) My **focus** has been **only** on **PC Fútbol 6.0** as “it is the best football manager in history”



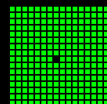
# PCx-Utills (before → PKF)



- **PKF** files are data containers for the **PCx (PC Fútbol and PC Basket)** games developed by **Dynamics Multimedia**
- The structure of a **PKF** file is defined by several **set of 32 data pointers** containing **unique id, offset** and **size of a data chunk**

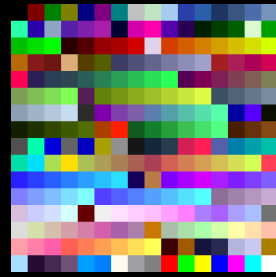


- : Empty bytes
- ▣: Set of Pointers
- : Data chunks
- : Pointer terminator



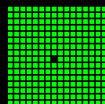


# PCx-Utills (before → PKF)

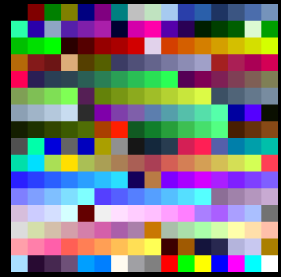


- First approach
  - **Emacs + hexl-mode** and keep staring at the **bytes**
  - **No knowledge of pointers** and used the **Dynamics Multimedia Copyright** notice text as **separator**
  - **Many errors** caught by parser (\*)

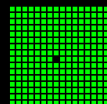
(\*) **Analyzing** a sequence of **bytes** conforming to a **set of rules** and stored in an **AST**



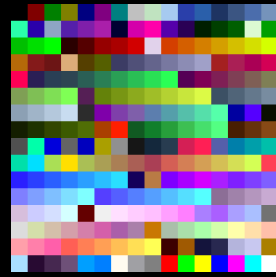
# PCx-Utills (before → PKF)



- Second approach
  - **Understand** that there are **sets of 32 pointers** before and between data chunks
  - **Few errors** (only three)
- Third and final approach
  - Not all **PKF** files use the same **obfuscation for the unique ids**, therefore, just **store bytes** instead of transforming to unique identifier



# PCx-Utills (before → DBC)

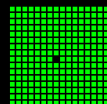


- In order understand **data chunks**, **bytes** need to be **rotated** with **Dynamics Multimedia** obfuscator. Notice that: `rot (rot bytes) = bytes`

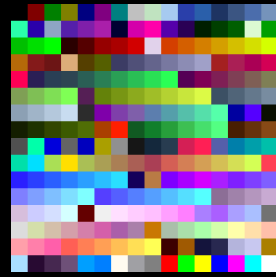
```
bijjective
  :: Word8
  -> Word8
```

```
-----
bijjective b
```

```
| 032 > b = if b .&. 01 == 0 then b + 97 else b + 95
| 064 > b = if b .&. 01 == 0 then b + 33 else b + 31
| 096 > b = if b .&. 01 == 0 then b - 31 else b - 33
| 128 > b = if b .&. 01 == 0 then b - 95 else b - 97
| 160 > b = if b .&. 01 == 0 then b + 97 else b + 95
| 192 > b = if b .&. 01 == 0 then b + 33 else b + 31
| 224 > b = if b .&. 01 == 0 then b - 31 else b - 33
| otherwise = if b .&. 01 == 0 then b - 95 else b - 97
```



# PCx-Utills (before → DBC)



- After **data** is **parsed** into an **AST**, the **next step** is to **transform** it back into a **binary data** container that can be used by the applications
  - Each **datatype** only has to **implement** and **instance** of the **ByteStream** typeclass

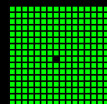
```
instance ByteStream Bool where
  bytes True  = [0x01]
  bytes False = [0x00]

instance ByteStream Word8 where
  bytes = tbs 1

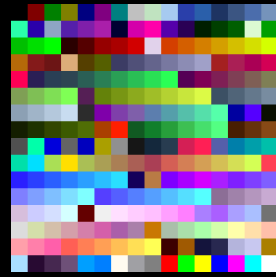
instance ByteStream Word16 where
  bytes = tbs 2

instance ByteStream Word32 where
  bytes = tbs 4

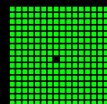
instance ByteStream String where
  bytes x =
    tbs 2 n ++ map r2b x
  where
    n = length x
```



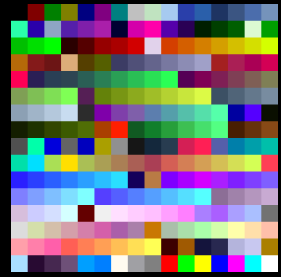
# PCx-Utills (before → DBC)



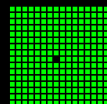
- Since **Dynamics Multimedia** used (delta) **DBC** files to do this, we just **mimicked** that as well
- By **removing database information** from **PKF** data chunks, a **very small size** can be achieved which is import when it comes to distribution
- **Remark: DBC** and **PKF** data chunks are very similar. Only a **few bytes change**



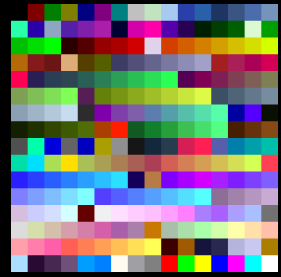
# PCx-Utills (before → API)



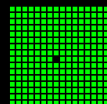
- So now we know how to create **DBC** files based on the **AST**
- Next step will be to **generate AST elements** from an external source
- As mentioned, we will be using the **FUT/FIFA API** from FUTBEST



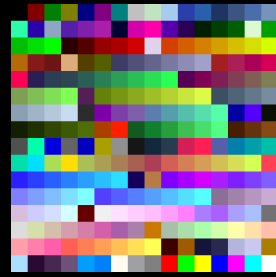
# PCx-Utills (before → FUT)



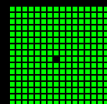
- All data is contained in **JSON** and images as **PNG**
- In order to retrieve the data and images both `curl` and `wget` are used
- Once the data is **local**, we can **transform** the data **into AST elements** with `jq`



# PCx-Utills (before → IM)



- In order to transform the images from **PNG** to the expected **BMP** format with the custom **Dynamics Multimedia** indexed colour table, ImageMagick as well as some home brewed tool (**pcx-colourpalette**) are used
- I noticed that the **template** used by the community is **BMP v.3** which uses **4 bytes** per colour (**RGBA**). I use the **BMP v.2** as it only uses **3 bytes** per colour (**RGB**) which **saves at least 256 bytes per image**. Since there are +10.000 image files, it helps when distributing files

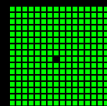




# PCx-Utills (before → ZIP)



- The result is distributed in a **13 MB** file and hosted by our own **Tykring** (\*) at hushfile.it
  - Latest pcf0061-eq022022-v.0.11.3.zip
- (\*) Probably owe you an **alcohol-Tsunami** at the bar

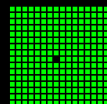


# Made PCx-Utils repo public

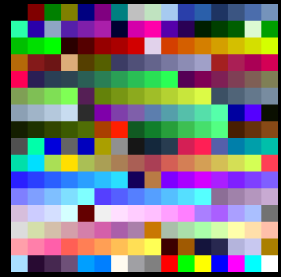


<https://gitlab.com/spisemisu/pcx-utils>

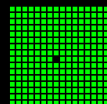
2020-07-16 @ HackMadrid %27



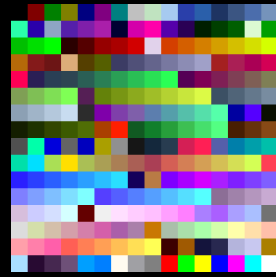
# PCx-Utills (after → ...)






- Main goal is to **automate as much as possible** the task of **updating data**
  - I knew about the **EA Sports** FUT/FIFA API (**free** and **no registration required**)
    - Sadly **it isn't updated anymore**, but still available though
  - Found **alternative** FUTBEST (still **free** and **no registration required**)
    - Some fields aren't populated with **FUT/FIFA** data (maybe only delivered with a **premium** account?)
  - Other goals are:
    - **Open project**, copyleft licensed, that would allow **people to contribute** but mostly **learn** in order to **spawn** other **similar projects** (\*) for the **benefits of the Community**
    - **Edit data** by the **end-users** preferred **text editors** which would **allow most people to use the tools**
- (\*) My **focus** has been **only** on **PC Fútbol 6.0** as “it is the best football manager in history”



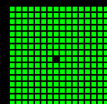
# PCx-Utills (after → Feedback)



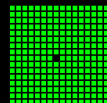
Edición				Marcar este foro como leído	
Tema / Autor	Respuestas	Vistas	Puntuación [asc]	Último mensaje	
 [Ayuda] Patrones en EQ022022.PKF JuanDeLaCierva	4	2.861	★★★★★	19-08-2018, 07:40 PM Último mensaje: JuanDeLaCierva	
 [Ayuda] Conversor de formato PC Fútbol a entendible y de formato entendible a PC Fútbol 1 2 3 4 ... 8 JuanDeLaCierva	109	8.521	★★★★★	23-07-2020, 05:55 PM Último mensaje: Tronic	
 editor Neuroboy01	4	5.127	★★★★☆	29-05-2018, 07:52 PM Último mensaje: LucasAriel 98	

Mou ( 7 ) - Última actualización 20-07-2020, 08:58 AM  
Reputación recibida por el mensaje de JuanDeLaCierva en PCx-Utills: Charla (Twitch.tv) + Proyecto (código)

Positivo (+1): una de las mayores aportaciones al campo de la edición en muchos años



# PCx-Utills (after → Contribution)



# PCx-Utills (after → Contribution)

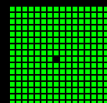


```
-- 2020-07-17: Scale code provided by @eckankar

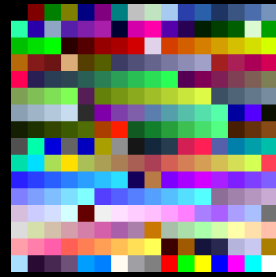
type Stats =
  ( Word
  , Word
  , Word
  , Word
  )

scaleStats
  :: Word
  -> Word
  -> Stats
  -> Stats
scaleStats old new (pace, stamina, aggression, skill) =
  ( scale pace
  , scale stamina
  , scale aggression
  , scale skill
  )
  where
    scale v = (v * new) `div` old

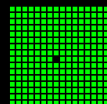
maxSkill :: Stats -> Word
maxSkill (pace, stamina, aggression, skill) =
  maximum
  [ pace
  , stamina
  , aggression
  , skill
  ]
```



# PCx-Utills (after → Patches)

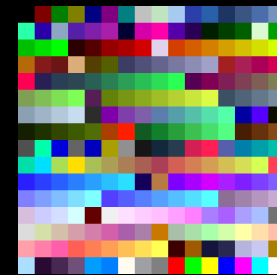


- Patch bytes in the executable (**MANAGER.EXE**):
  - Change calendar years from 1997-98 to 2019-20
  - Change birthday limit from 1900-2000 to 00001-65535
    - Use **zero** for when the application needs to **generate a random birthday**
    - Update all **1900-00-00** to **0000-00-00** after applying patch



# Demo (short)

Cpu: 013% 008% 038% 011% | Temp: 066°C 060°C 054°C 057°C | Mem: 008% |



**PC FÚTBOL 6.0**  
TEMPORADA 97-98

**dinamic multimedia**

eXtensión 1

eXtensión 2

**PCCALCIO PCPREMIER**  
ARGENTINA 1970/2014 '98

**PCFÚTBOL PCFRANCE**  
FRANCIA 1970/2014 '98

**HISTORIA**

**BASE DE DATOS**

**LOS GOLES DE LA LIGA**

**SEGUIMIENTO**

**PARTIDO AMISTOSO**

**DESAFIO AMÉRICA-EUROPA**

**LIGA MANAGER**

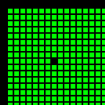
**LIGA PROMANAGER**

**PROQUINIELAS 2.0**

**LFP**

**Instrucciones** **Infútbol** **Créditos** **SALIR**

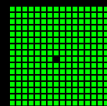
F2.12





# Summary

- I hope others realize that tools having support for **parser combinators**, **algebraic datatypes** (AST) and **type classes**, are useful for projects like this as they can help to **discover patterns** but also to ensure: **simplicity**, **structure** and **correctness**.



# Q&A



*DINAMIC* MULTIMEDIA

