DEFOLD™

About me

- Japan Riddles (1998), Discord Times (2004), Floris Reliquiae (2007, frozen), Mahjong Legacy (2009, frozen), Legends of Eisenwald (2015), Family Age (2017).
- Between 1996 and 2017 I wrote 10 engines. From DOS+VESA in 1996 to Win10+DX11 in 2017.

About our team

- The team was put together from skratch
- The team never made farm-games
- But this is not an obstacle if there are professionals

How Defold was chosen

- What were the alternatives? Unity - unbridled and risky, Unreal is too cumbersome (could get an open-world RPG), C++ is an ancient demonic language, you can unintentionally *call_Satan(*self)*. *// I did not check*
- In Defold, you can make the game in half an hour.
- Defold in its structure is very similar to the ideal engine.
- The authors of Defold came from the Swedish team Avalanche Studios, which made Mad Max on its own engine.

STARK GAMES

How to make the biggest game on Defold in 1 year

EASY TEAM®

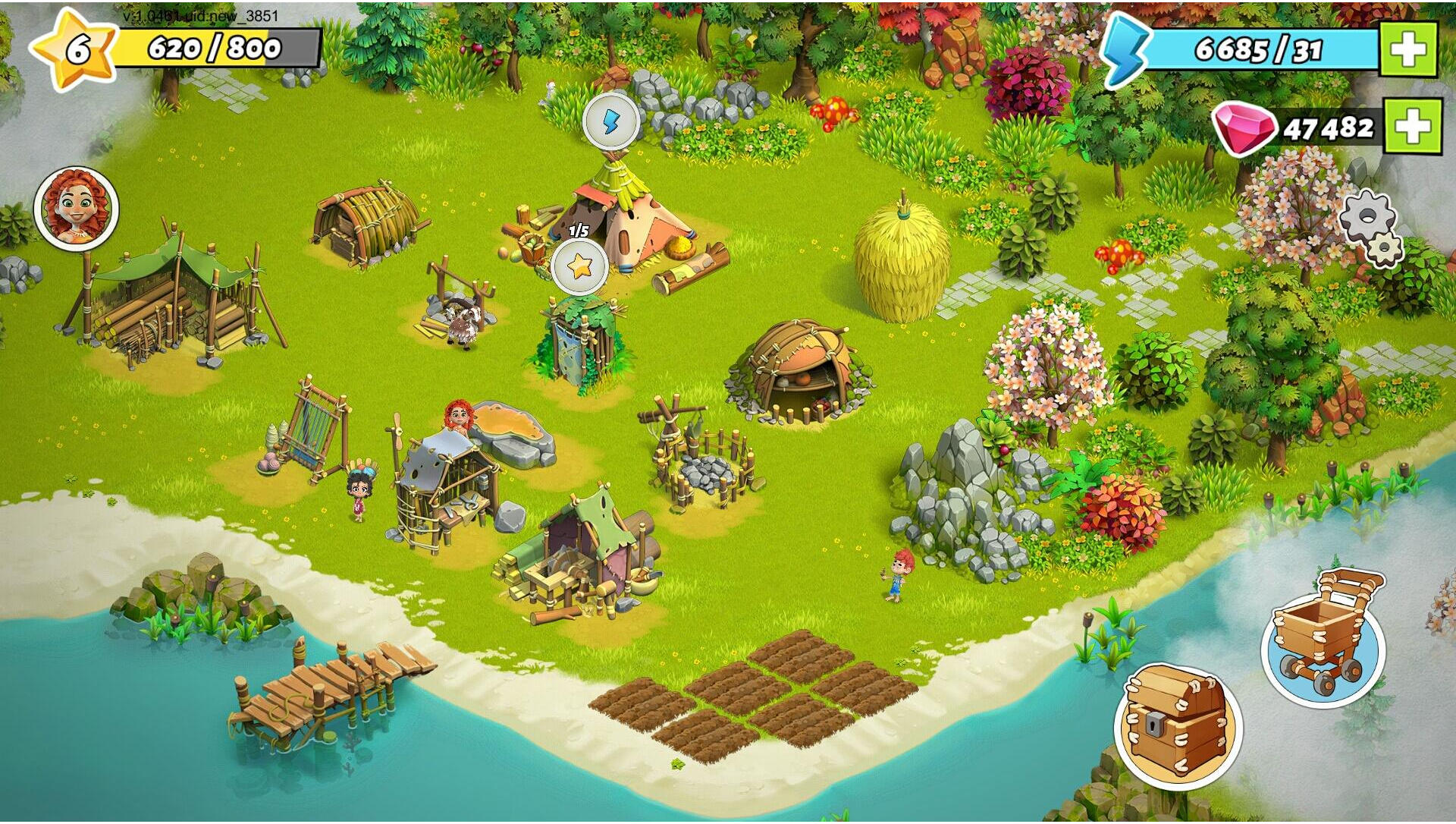
Our game - Family Age

- Farm for mobile devices from industry professionals.
- 1 year of work. 20 months of programming. 150 months of work of the rest of the team.
- 40000 lines of code
- Volume of art (1GB RAW TGA), code size (1.5MB LUA, more than 100 modules), game databases (1MB JSON, more than 150 files), sound (more than 30MB WAV + OGG).

STARK GAMES

How to make the biggest game on Defold in 1 year

EASY TEAM®



Technical production pipeline

- Chaos Development methodology (seems like indie development).
- First 1 programmer - does tests and primary R&D.
- Then 2 programmers - 1 makes the game code, 1 makes the editor.
- Then 5 programmers - 1 interface, 1 editor and export, 2 game code, 1 client and statistics.

Statistical measurements

- FPS on different devices (ZTE = 7...10, Note3 = 15...25, iPhone7 = 60, iPad Air = 30...40, MiPad = 20...30).
- A typical number of game objects is about 4000.
- Typical execution time of LUA is 20~40% of the total frame time.
- The memory occupied by the game is from 200MB to 500MB depending on the device.

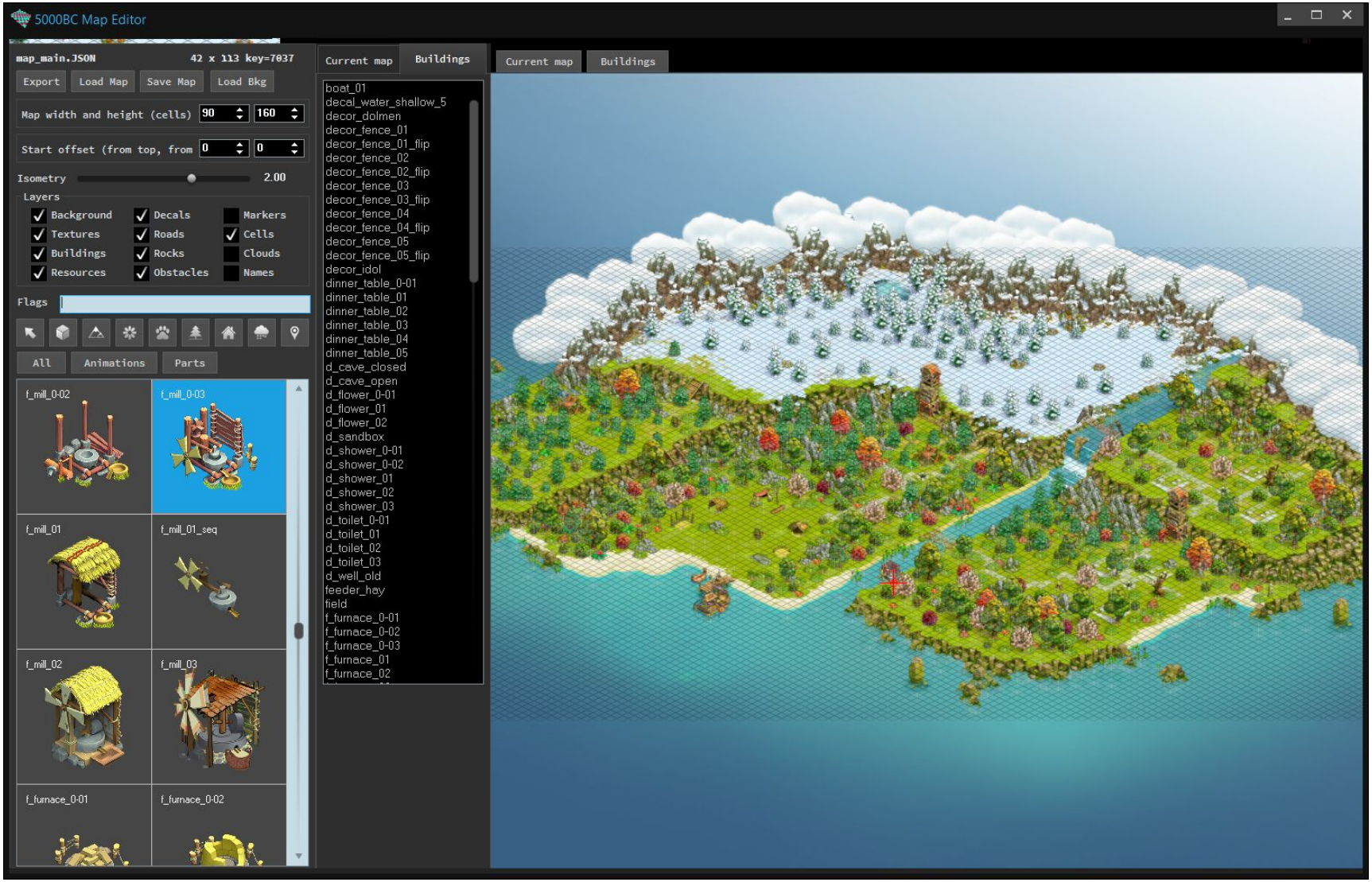
Defeated problems

- Optimization of JSON main map size – from 7MB to 125KB.
- Loading progress bar.
- Dynamic loading of game buildings.
- Path search clusters.
- Fears of FPS.
- Drawcalls - from 1800 to 100.

Used solutions

- Defold editor (№2)
- Atom (as LUA IDE with many plugins)
- Our own Content Editor
- Build-machine on the our server

How to make the biggest game on Defold in 1 year



Our own Content Editor

- An absolute must for a big game on the defold
- Provides comfortable editing of game levels and objects
- Collects all resources from regular folders on Windows
- Processes sprites (trimming alpha, distribution by atlases)
- Prepares all links between resources (creates Defold objects)
- Analyzes game design data and reports errors
- Generates game json by special rules
- Can upload partial updates of the game to our server

it's time to sum up

Advantages of Defold

- Ease of obtaining the final product
- Good optimization of rendering 2D
- Easy to write complex code (due to Lua)
- Sufficiently rapid reaction of developers to requests
- Direct access to shaders is enough
- Support for native code

Cons of Defold

- Persistent bugs with spine
- Problems with editing the very big codebase
- The organization of resources is predefined at the time the build was started
- The development environment is very "raw“
- Low performance of Lua

Let's look at the development at an unexpected angle!

Nuances outside the topic

- Does your engine add risks?
- Legacy code in third-party engines, closed sources, bugs and crutches
- Why all these dances around the engines, if your engine takes up little?

Contacts

- Skype: nikolay.armonik
- E-mail: nikolay.armonik@gmail.com
- Facebook: /nikolay.armonik
- Twitter: @Morgerion
- LinkedIn: /in/morgerion