Waternet Multiplatform Retro Game

Waternet is a multiplatform puzzle game written for old consoles and handhelds like Game Boy (Color), Game Gear, Master System, Analogue Pocket and Mega Duck using the gbdk sdk.

Buy me a "koffie" if you feel like supporting

I do everything in my spare time for free, if you feel something aided you and you want to support me, you can always buy me a "koffie" as we say in dutch, no obligations whatsoever...

Controls

Button Nintendo	Button Sega	Action
A / Start	1 / Start	Confirm in menu and level selector, rotate or slide action while playing
B	2	Back in menu, level selector and game

Game Modes

The aim of the game, in any game mode is always to connect all pipes so that water can flow through them from the water point source. How you can accomplish this depends on the game mode. The game has a help section in the main menu where you can always check up on the rules of each game mode.

Rotate Mode

You need to connect all the pipes so water flows through them, by pressing the A or 1 button (depending on the system) on a pipe, to rotate the single pipe.

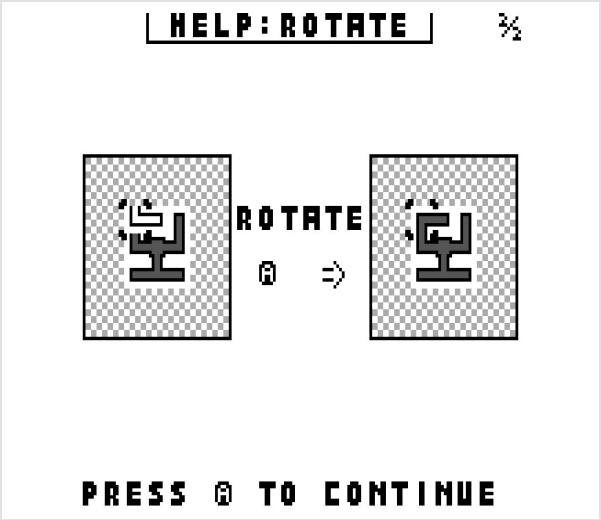
Slide Mode

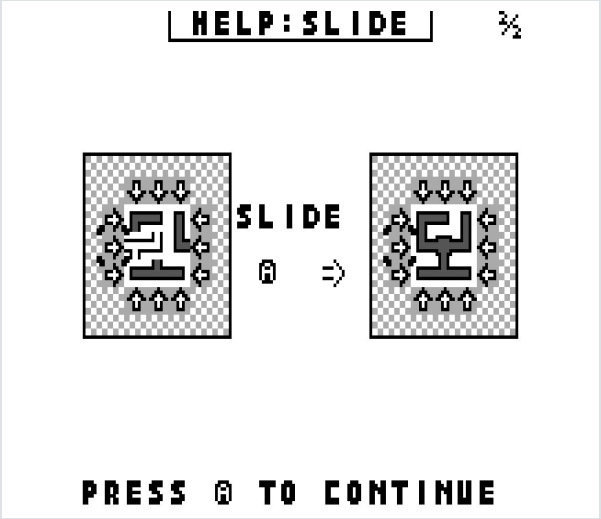
You need to connect all the pipes so water flows through them, by pressing the A or 1 button (depending on the system) on the arrows of a row or column. The row or column will move all pipes in the direction the arrow is pointing at.

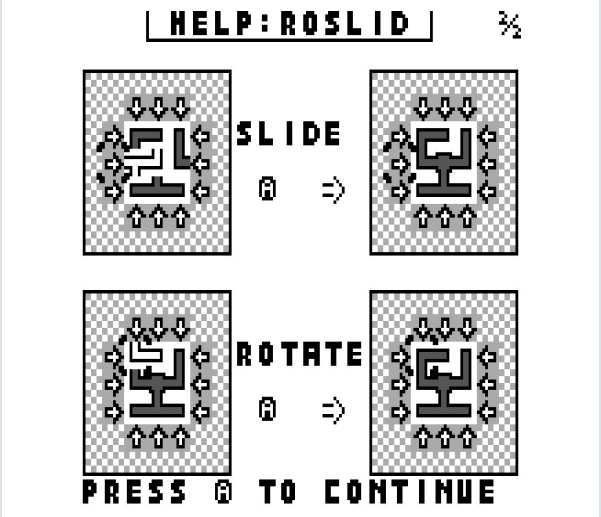
Roslid Mode

You need to connect all the pipes so water flows through them, by pressing the A or 1 button (depending on the system) on the arrows of a row or column. The row or column will move all pipes in the direction the arrow is pointing at. You can also press the A or 1 button (depending on the system) on a pipe, to rotate the single pipe. This is a combination Rotate and Slide mode.

Ingame help slides







Graphics

Graphics (tiles), i created in Game Boy Tile Designer and the titlescreen graphic is based on a modified title screen image from my waternet game i made for gp2x and windows. I designed the title screen in the Gimp Image Editor to just have the word waternet, the waterdrop and the menu box to fit it all on the Game Boy

screen. Afterwards i used Game Boy Png Converter to convert the titlescreen to an optimized tilemap and tiles.

Music

Music was made using Online Sequencer, i created single channel, non mixed music files and later converted this music to an array storing the frequencies of notes to be used on the gameboy. I got this idea from the sheep it up game developed by Dr. Ludos. He stored the frequencies to be used for the music notes in a first array, containing the registers for the gameboy for channel 2. Later he referenced this array in another array to create the music itself with these notes. So i used the same system and all i had todo was convert the music from onlinesequencer i made to such array (by hand) as well.or the Sega Hardware (Game Gear, Master System) i did something similar but related to the sound chip used (PSG). I just had to use a different array for the values used for frequencies and create a similar function as the one i used on the Game Boy.

You can find the music files i used for the game below:

- Game Music: https://onlinesequencer.net/2485064
- Title Music: https://onlinesequencer.net/2484977
- Level Done Tune: https://onlinesequencer.net/2484974

Testing on Real hardware

To test my rom on real hardware i used a flash cartridges like for example the EZ-Flash Jr for Game Boy (Color).
It allowed me to put my compiled rom on a sd card and then load that rom from the system itself.
I'm guessing these flash cardridges flash a certain chip on the fly with the provided rom, but it allowed me to test my game's on a real hardware.
I highly advise anyone making similar games to do the same as initially everything looked fine on an emulator on pc but as soon as i tested on the real hardware like the original Game Boy i noticed some tiles were not really visible and i had to adapt the tiles i initially created.

Making your own Waternet (gameboy) cartridge

The game for GB / GBC requires a MBC5 type cartridge with at least 32KB Rom + 8k RAM + BATTERY to save your progress. I personally used flashgbx with a chinese bootleg cartridge of Penta Dragon and the GBxCart RW cartridge rom dumper / flasher. The batteryless save versions require specific bootleg cartridges and allows you to save your progress without the need of having a battery attached to your cartridge, it will (re)write the sram at the end of where the rom is saved while playing the game.

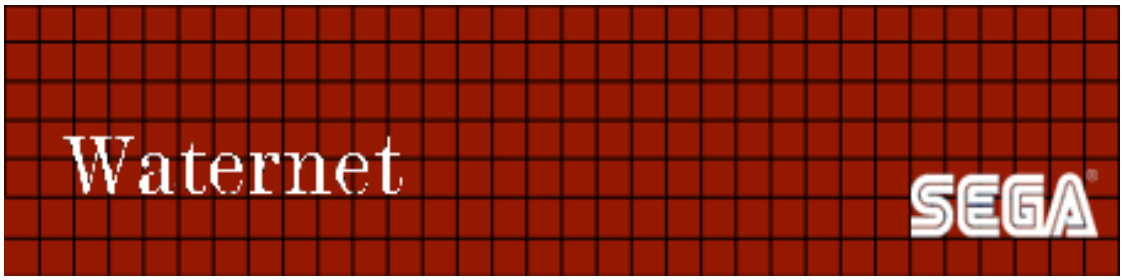
Cartridge Graphics

To create the cartridge graphics i used a template made by armando92 to start with in Gimp and then added different pictures i found on cleanpng to create the Cartridge Label and the logo i created using a free logo creator site. You can see the end result below.

Game Boy (Color) - 4.2 x 3.7 cm.

Game Gear

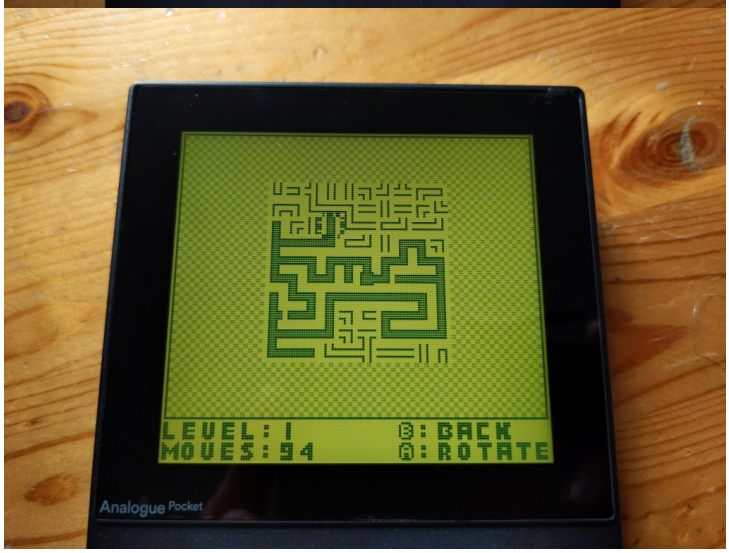
Master System - 10.3 x 2 cm.

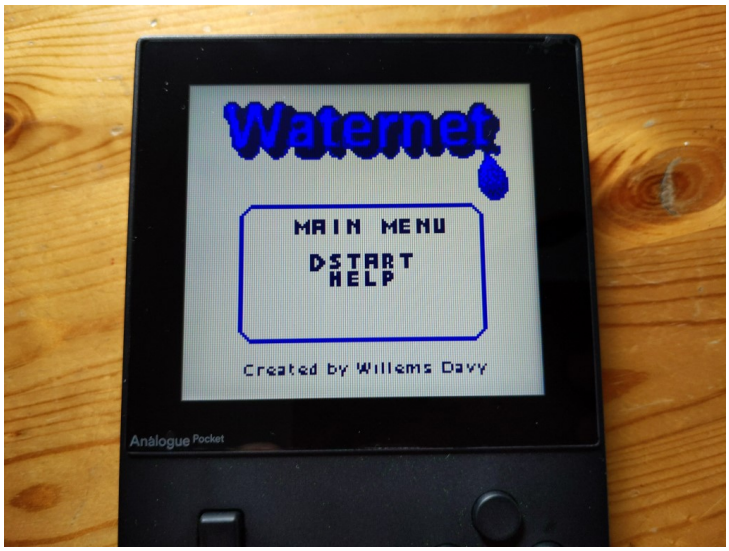


Photo's of the game running on real hardware

Analogue Pocket (thanks to bbbbbr)

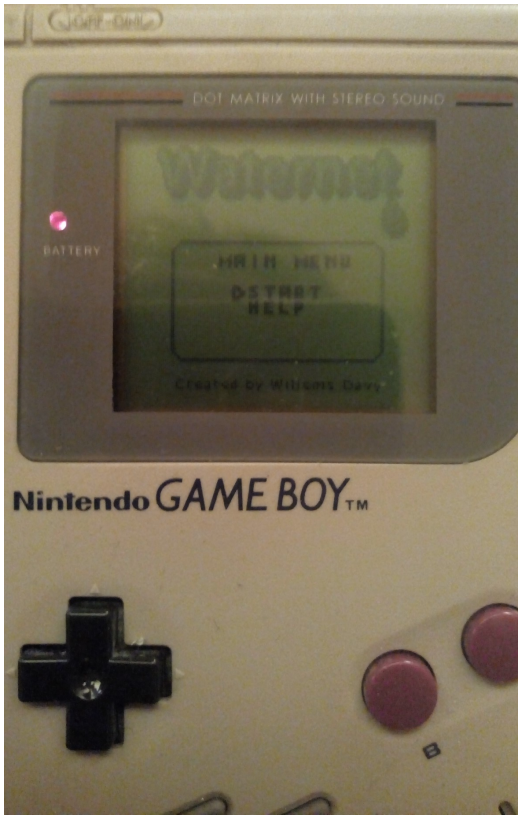

Waternet
MAIN MENU
START
HELP
Created by Willems Davy
Analogue Pocket


LEVEL:1
MOVES:94
B:BACK
A:ROTATE
Analogue Pocket


Waternet
MAIN MENU
START
HELP
Created by Willems Davy
Analogue Pocket


LEVEL: 1
MOVES: 54
B: BACK
A: ROTATE
Analogue Pocket

Game Boy


DOT MATRIX WITH STEREO SOUND
BATTERY
MAIN MENU
START
HELP
Nintendo GAME BOY™

Game Boy Color (thanks to SelvinPL)

Gb Boy Colour

Credits

Waternet game concept is based on the Net and Netslide game from the Simon Tatham's Portable Puzzle Collection,
it's my own implementation of these 2 game concepts

Tutorials used

- GamingMonsters "How to create a gameboy game" video tutorials on youtube

Code Credits

- Fading function for Game Boy Color is based on ZGB's implementation - https://github.com/Zal0/ZGB
- Super Gameboy Border implementation is taken from GBDK-2020 SGB Border example - https://github.com/gbdk-2020/gbdk-2020
- Music implementation based on system used in the sheep it up gameboy game made by Dr. Ludos
- Music on Sega hardwarde wouldn't have been possible without the help from SelvinPL's Sega Sound example
- Batteryless save implementation is Toxa's implementation, only changed the commands for my (reproduction) cartridges - https://github.com/untoxa/batteryless_save
- Flash chip commands for batteryless save taken from FlashGBX / GBXCartRW - https://github.com/lesserkuma/FlashGBX

Cartridge Graphcis used

- Game Boy Cartridge Template - armando92
- Game Gear Cartridge Template - SegaSonicFan Designs
- Master System Cartridge Template - borracho2x
- Water droplets - Baure
- Plumber Guy - Endara
- Plastic Pipes - Jaumo

Tools used:

- GBDK 2020 - https://github.com/Zal0/gbdk-2020/
- Visual Studio Code - https://code.visualstudio.com/
- BGB - https://bgb.bircd.org/
- Emulicious - https://emulicious.net/
- Game Boy Tile Designer - http://www.devrs.com/gb/hmgd/gbtd.html
- Game Boy Map Builder - http://www.devrs.com/gb/hmgd/gbmb.html
- Game Boy Png Converter - https://github.com/gingemonster/GameBoyPngConverter
- Gimp Image Editor - https://www.gimp.org/
- Online Sequencer - https://onlinesequencer.net/
- Music notes to Game Boy Frequencies chart - http://www.devrs.com/gb/files/sndtab.html