# This paper for some issue and other considerations

Please be tolerant on my English 😉

I’ve developed a software starting from this Mario replication: [https://archive.codeplex.com/?p=mario]

I can use it launching from Visual Studio, but I can’t obtain a working setup for this issue:

*Application: Mario.exe*

*Framework Version: v4.0.30319*

*Description: The process was terminated due to an unhandled exception.*

*Exception Info: System.DllNotFoundException*

*at SFML.Graphics.RenderWindow.sfRenderWindow\_createUnicode(SFML.Window.VideoMode, IntPtr, SFML.Window.Styles, SFML.Window.ContextSettings ByRef)*

*at SFML.Graphics.RenderWindow..ctor(SFML.Window.VideoMode, System.String, SFML.Window.Styles, SFML.Window.ContextSettings)*

*at GameEngine.GameObject..ctor(System.String)*

*at Mario.Program.Main()*

*Faulting application name: Mario.exe, version: 1.0.0.0, time stamp: 0x5ad50054*

*Faulting module name: KERNELBASE.dll, version: 6.3.9600.18938, time stamp: 0x5a7dd8a7*

*Exception code: 0xe0434352*

*Fault offset: 0x00015ef8*

*Faulting process id: 0x158c*

*Faulting application start time: 0x01d3d5be2c159d46*

*Faulting application path: C:\Program Files (x86)\artcava\Setup\Mario.exe*

*Faulting module path: C:\Windows\SYSTEM32\KERNELBASE.dll*

*Report Id: 69ce939b-41b1-11e8-80e1-ff9cc5c9199f*

*Faulting package full name:*

*Faulting package-relative application ID:*

At this time, I cannot solve this, consider I'm not used to developing desktop applications. I've been working for years on the web.

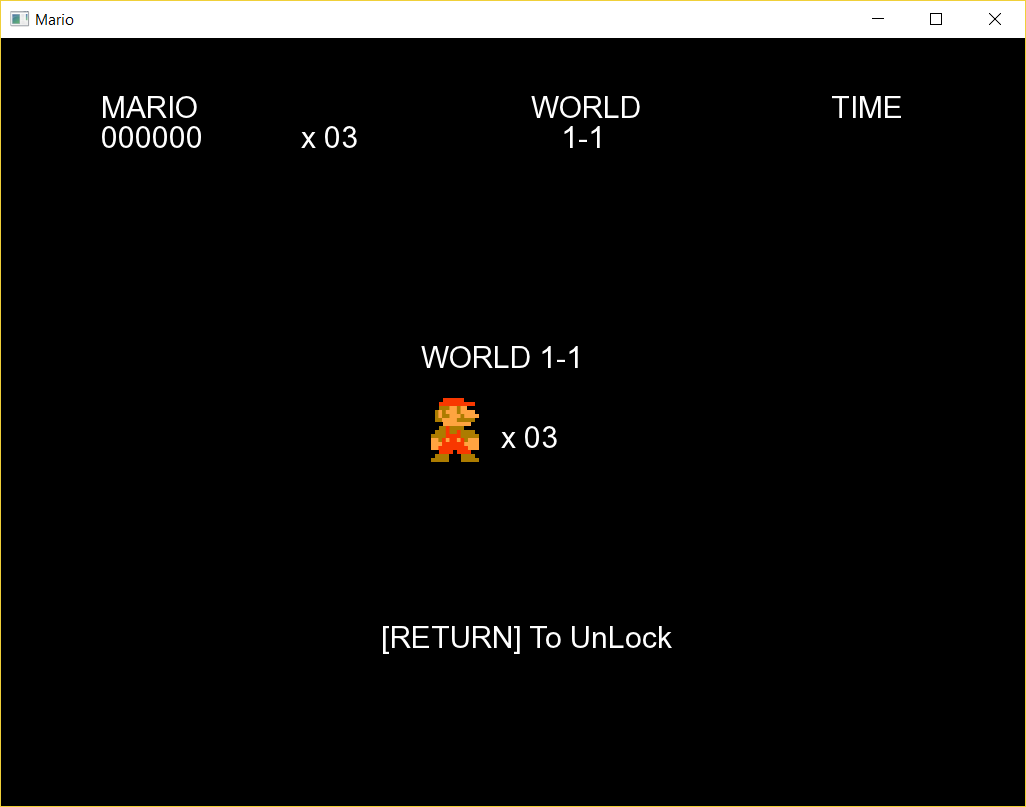
# Requirements

1. Take an open source game and modify it to be locked.
2. The user installs the game and can't play until he/she sends 100 Dime to a specific address
3. The game will constantly pull API data from blockchain checking if wallet X sent to wallet Y >100 DIME.
4. The game stops pulling data and unlocks only after the 100 DIME was sent.
5. The transaction is executed through Android app/ or any DIME wallet.

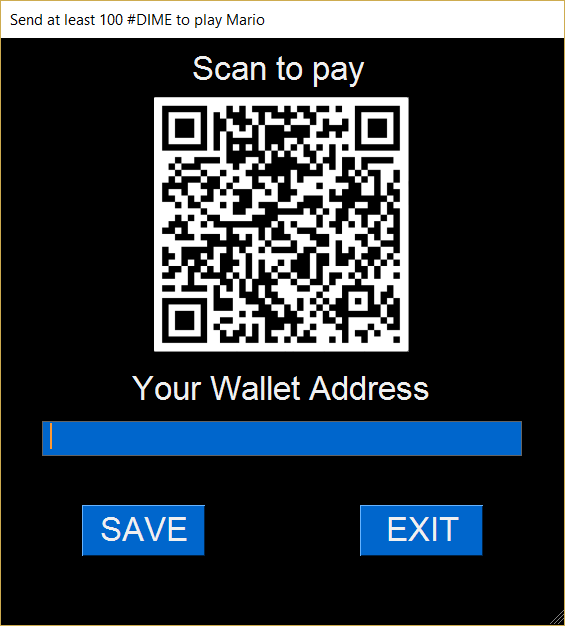
# Application

The application starts locked like in the pic below:

User have to press RETURN key to unlock

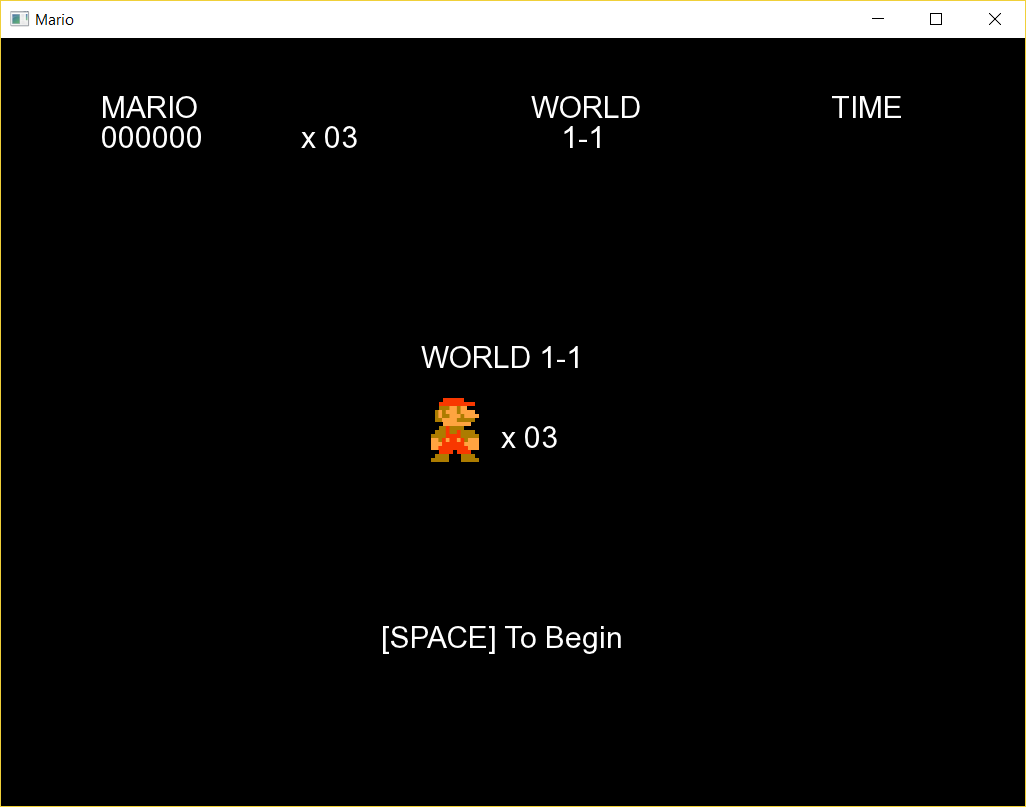


When user press RETURN this window is open:



User have to write down his/her wallet (the one used for unlock transaction) and save. Then user scan QRCode from wallet app to send at least 100 DIME.

When the game match transaction between wallet the game is unlocked:



# Security issues

* Application do not check sender wallet ownership

Anyone paying first time can share sender wallet to other players. A solution, as suggested by George Seward, is to use wallet unlocking inside the game.

So we can store some other infos over blockchain that have to be verified in order to unlock the game or whatever.

* Receiving wallet address is written down an external file

We can hardcode wallet address, even if it is not so scalable…

Perhaps we can consider that transactions can be checked by a remote service, so destination wallet cannot be changed by user. [Ex: API Rest system, json based]

* Anyway, a remote service could be bypassed

I'm thinking, for example, how old Adobe’s versions were cracked.

Solution is not so simple… when hacker identify code’s lines where API is called, substitution is quite simple.

Perhaps we can use some private/public key exchange during API calls? Quite like OpenID protocol or similar.

# Structural issues

* We do not handle API calls querying blockchain

At any time <https://prohashing.com> can change entrypoint functions or make it unusable, they are not required to tell us.

Perhaps we have to develop ourselves all the system to query the blockchain.

# Question

May I prepare some service to produce dotted QRcode like the one shared on TG?