Report Investman

Gijs Koenders, Julian van Bree and Selim Berntsen

ANR: u998822, u556493 and u303699

Tilburg University

Pre-master DSBG/CSAI

Basic Programming

Group assignment

17-01-19

1. **Background.** Learning to invest seems like a daunting challenge, especially for young investors willing to give it a try. However, investing often requires substantial knowledge and years of experience to be remotely successful. A recent study by Blockchain Capital estimated that 30 per cent of millennials became interested in investing due to the significant rush of capital into cryptocurrency markets in 2017 and started investing without prior experience (Arnold, 2018). Therefore, a challenge of creating investing games for millennials to learn how to invest in fun and competitive ways became apparent. Even though this is not a societal issue, but rather an issue of the inability to learn investing in interesting and competitive ways, the group decided this would be an interesting project to be done in Python. Currently, Python code is often applied to build algorithms for automatic trading, stock market prediction and stock market analysis in the field of investing, however, it is rarely used to create games. While several investing and trading applications exist, very few can be considered competitive and ‘fun’ in nature. Additionally, others have created applications in Python for a data-driven approach to cryptocurrency speculation (Triest, 2017), personal trading assistance (Sentdex, 2014), trading simulators (James, 2018) and trading bots (Github, 2017). This assignment offered a unique opportunity to learn complex python programming while creating a fun and interactive application.
2. **Methods.** This project aims to teach investing by simplifying interactions and creating an interesting no-risk environment to learn how to predict markets, gain knowledge about cryptocurrency and defeating your friends. Several methods were used to create this application. Initially, the chosen GUI module for the application was Tkinter, due to its ubiquity and simplicity to implement (Shipman, 2013). Additionally, to be able to plot updating graphs in the interface, the module Matplotlib was studied and implemented, due to its simplicity and well-tested, cross-platform graphics capabilities and the ability to be supplied with data gathered by regular expressions from a website (Cryptowatch, 2019). Furthermore, an SQL database is used to save user information and track user progress, allowing users to login and continue playing at later times without losing progress.
3. **Results.** The choice to use these novel methods (to us) was a greater challenge than expected. Specifically, the coupling of the SQL database and studying to fully understand the capabilities of Tkinter was extremely challenging. Furthermore, initially the objective was to use an API to receive currency information, however, due to the complexity and difficulties implementing, all cryptocurrency information for plotting and trading is done through regular expressions. Considering the setbacks, we have managed to create a game that does what we wanted it to be able to do, with the exception of some (seemingly inevitable) bugs.
4. **Discussion:** We feel like we have conceived a unique approach to cryptocurrency investing and believe that, especially for younger learners, fun and interactive ways to learn are better than diving into theory and statistics, while most likely losing money in the process. Additionally, a game like this can easily mimic real life trading situations as it uses real-time currency data for the player to make informed decisions. Even though the game can be considered rather simplistic for now, future additions could greatly improve the game. Specifically, additions like (1) retrieving more detailed information about different markets and individual cryptocurrencies, (2) adding the option to trade in different cryptocurrencies or possibly company stocks, (3) improving server management and the ability to support more (possibly online) players, (4) upgrading to the use of an API instead of depending on regular expressions and (5) allowing for more interactivity with graphs are examples of future improvements and additions.

**References**

* Arnold, A. (2018, January 7). *30% Of Millennials Would Rather Invest In Cryptocurrency: Here Are 3 Tips To Help You Do It Smarter*. Retrieved on January 14, 2019, from Forbes: https://www.forbes.com/sites/andrewarnold/2018/01/07/30-of-millennials-invest-in-cryptocurrency-here-are-3-tips-to-help-you-do-it-smarter/#7094b82c7861
* Cryptowatch. (2019). Retrieved on January 14, 2019, from Cryptowat.ch: https://cryptowat.ch/
* Github. (2017). *Trading bot*. Retrieved on January 14, 2019, from Github.com: https://github.com/topics/trading-bot?l=python&o=desc&s=updated
* James, F. J. (2018, March 24). *A simple crypto trading simulator*. Retrieved on January 14, 2019, from hackernoon.com: https://hackernoon.com/how-to-build-a-simple-crypto-trading-simulator-part-1-4ccdddcd6b76
* Sentdex. (2014, October 26). *TkInter Tutorial Python 3*. Retrieved on January 14, 2019, from Youtube.com: https://www.youtube.com/watch?v=A0gaXfM1UN0
* Shipman, J. W. (2013). *Tkinter 8.5 reference: a GUI for Python.* New Mexico: New Mexico Tech. Retrieved on January 14, 2019, from http://infohost.nmt.edu/tcc/help/pubs/tkinter/tkinter.pdf
* Triest, P. (2017, August 20). *Analyzing cryptocurrencies python*. Retreived on January 14, 2019, from Blog.patricktriest.com: https://blog.patricktriest.com/analyzing-cryptocurrencies-python/