

**PET WARS**

June 29, 2023

Hristo Ivanov

Strypes LAB Project

1. System requirements …………………………..2
2. Installation…………………………………………...2
3. Game Overview…………………………………...3
   1. Description……………………………….…...3
   2. Units……………………………………………...4
   3. Resources……………………………………...6
   4. Farms………………………………….….….....7
   5. Adventure Map………………………….....8
   6. Controls………………………………………...8
4. Code Overview…………………….….………..…9
   1. Behind the adventure map………..…..9
   2. Hero and movement on the map...11
   3. Resources and Farms……………………12
   4. Events……………………………………….….12
5. Developer Notes…………………………………13
   1. Motivation…………………………………….13
   2. Inspirations……………………………………14
   3. Starting from scratch……………….….…15
   4. Artwork Notes……………………………….16
   5. Challenges……………………………….….…17
   6. Future development.………………….….18
6. Credits and download link……………………19

1. System requirements

The game was tested on the following configuration:

* Windows 11 22621
* Python 3.11
* Pygame 2.4.0

However, feel free to try it on Linux or other versions of OS, Python or Pygame.

2. Installation

2.1 Python - <https://www.python.org/downloads/>

2.2 Pygame - <https://www.pygame.org/wiki/GettingStarted>

2.3 PetWars - clone the GitHub repository: <https://github.com/hristogwivanov/PetWars>

3.Game Overview

3.1 Description

Pet Wars is turn based strategy game (TBSG) created using Python and Pygame. You have limited moves – in the current version 3 each turn and then it is the opponents turn. The main game is focused on exploring the adventure map and hiring an army. In your journey you can fight other armies, collect resources and take control of strategic objects on the map. The main goal is to defeat your opponent – in this early version of the game the one and only opponent is Shaq the Pug.

3.2 Units

There are 2 fractions cats and dogs, leaded by a hero which is not taking part in the battles. If the hero is losing his army, the game is lost, if the enemy hero is defeated – the game is won. Each fraction is having 3 units – basic melee unit, an archer, and a top tier fighter. There are also 2 Neutral units – a mouse and a dragon. More details on each of them below:



Nebuchadnezzar: Our hero!

The cats follow him, but he never

takes part in a fight!



Street Cat: Found on the street.

Basic melee unit. Don’t expect much!



The Persian: Loves milk!

The cats archer, however

I still can’t explain how it shoots.



The Sphynx: Guardian of the pyramids. A top tier melee unit.

Attacks twice.



Shaq: The prime evil, the leader of all dogs. Controlled by AI. Defeat him and you win the game!



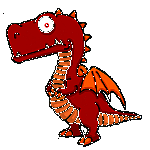
PUG: Slow speed, weak melee attack. Deteriorating vision. I suspect he is relative to Shaq.



Malinois: The dogs ranged unit. Artificially created. Attacks for no reason.

Shiba Innu: The top tier dog unit. Can teleport on the first turn on the battlefield. Sent by Elon Musk on expedition to the moon.





What is a computer game without a dragon? Don’t poke him! Kills every stack on the battlefield with one strike only!



It’s a mouse. Annoying, usually guards something that no one wants. 1 Attack, 1 Defense, 1 Health.

3.3 Resources

In Pet Wars we have 6 resources – MeowCoin, Milk and Fish for the Cats and respectively DogeCoin, Bones and Meat for the Dogs. Detailed description below:



Meow Coin is what the cats are using instead of gold. It is so bad that I use Monero image instead.



Milk. 3.6% of Fat.

Produced by a happy cow.



Fish is caught by the cats during their spare time. Get control of the fish pound to get it daily.



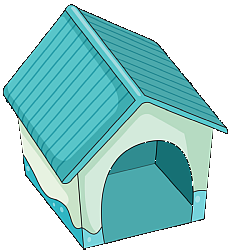
Meat is getting production bonus In every Bulgarian village the weeks before Christmas.

The bones are usually found in the garden. Unknown origin.

Doge Coin helped a lot of people to retire. However, it’s not always used with good intentions.

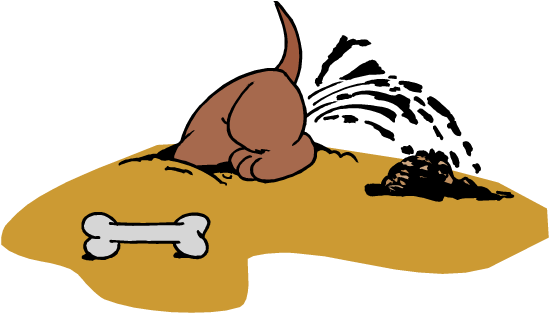
3.4 Farms and houses

The resources are not only found on the map, you can also generate them if you have the control of the below strategic objects:



Cat House: collects 1000 Meow Coins daily from cat’s taxes

Dog House: collects 1000 Doge Coins daily from dog’s taxes



Fish Pound: produces 1 fish daily

Bone Yard: produces 2 bones daily

Pig Farm: produces 1 meat daily

Milk Farm: produces 2 milk daily



3.5 Adventure Map

The adventure map is consisting of 140 – 14x10 tiles – each is having the size of 100x100px. You can explore it and depending of the tile that you reach you trigger various events.

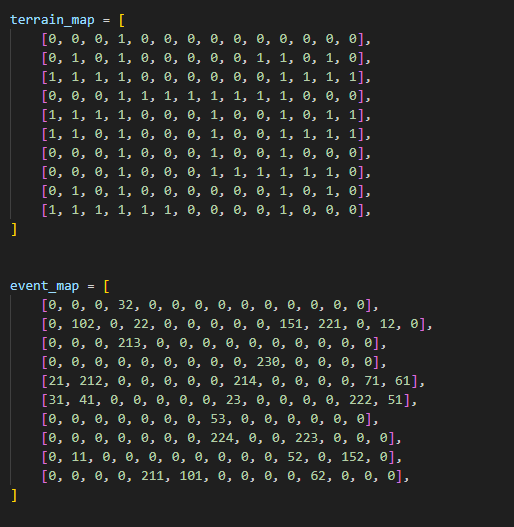
3.5 Controls

The control is from the keyboard and the mouse. The submarine Logitech F710 is not supported yet. The keyboard arrows are moving the hero one hour in the direction of the arrow, or you can click with the mouse in a neighboring tile to move it. For the end of turn you just press ‘E’ or you can click the designated interface button in the bottom right corner.

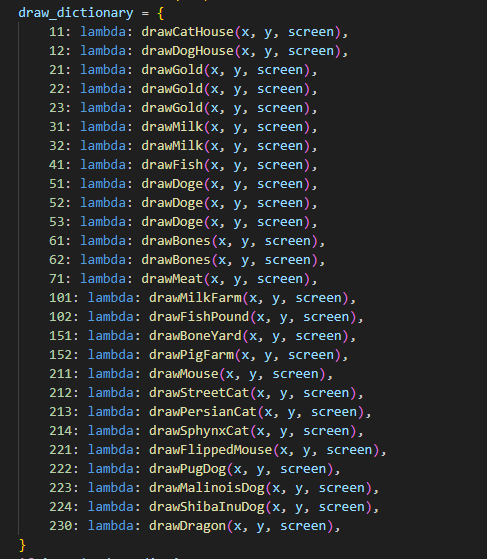
4.Code Overview

4.1 Behind the adventure map

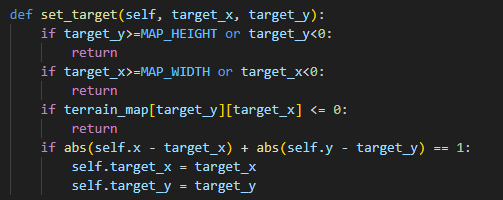
The adventure map consists of 2 layers – the terrain map and the event map. They are represented as list of lists of numbers – the map is a list of rows which are lists of the value for each tile:

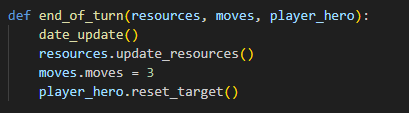


The terrain map is a constant and is never changed. The visual representation is just a single png file created with the HoMM 3 map editor. The event map dynamic. It is changing depending on the game mechanics. Each single number represents objects which is drawn on the map based on the draw dictionary:



4.2 Hero and movement on the map

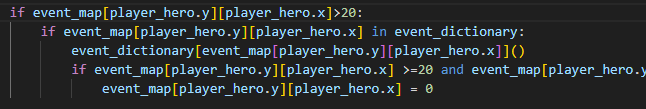
The hero is a instance of a Hero class. Same class is used for the enemy hero. It is initialized getting the position and the image from the main loop. It has draw, set\_target, reset\_target and update methods. It is getting the target from the main loop, which is currently allowed to be only within the neighbor tile. Of course, this is planned to be expanded in future. Also, it is not allowing to pass through obstacles on the terrain map: 

And finally, the movement logic in the main loop – we check if we have quit event, then we check for keyboard input, then for mouse input and act accordingly. When the moves counter goes to 0, we have to trigger the end of turn:

4.3 Resources and farms

There is resources class, an instance of it is containing the current values for each resource and the production rates. Both resources and farms are presented on the event map. Visiting the resources adds to the quantity and changes the event map value to 0. Visiting a farm is increasing the production value. In the end of each turn the production values for each resource are added to the total quantity. All this is handled by simple functions in gamedata.py, there is separate function for drawing and for the game mechanics. The initial idea was each resource and farm to be instances of a class, but that was dumped to avoid over complication following the YAGNI principle. However, for future development I might get back to the class mechanics which will give some advantages in creating bigger maps with more players, but there are issues that need to be addressed first.

4.3 Events

A dictionary is used for the events handling. It checks the tile on the event map and then triggers the corresponding function: 

If the code belongs to a resource (events 20-100) the field on the event map goes to 0 in order to remove the taken resource from the map.

5.Developer Notes

5.1 Motivation

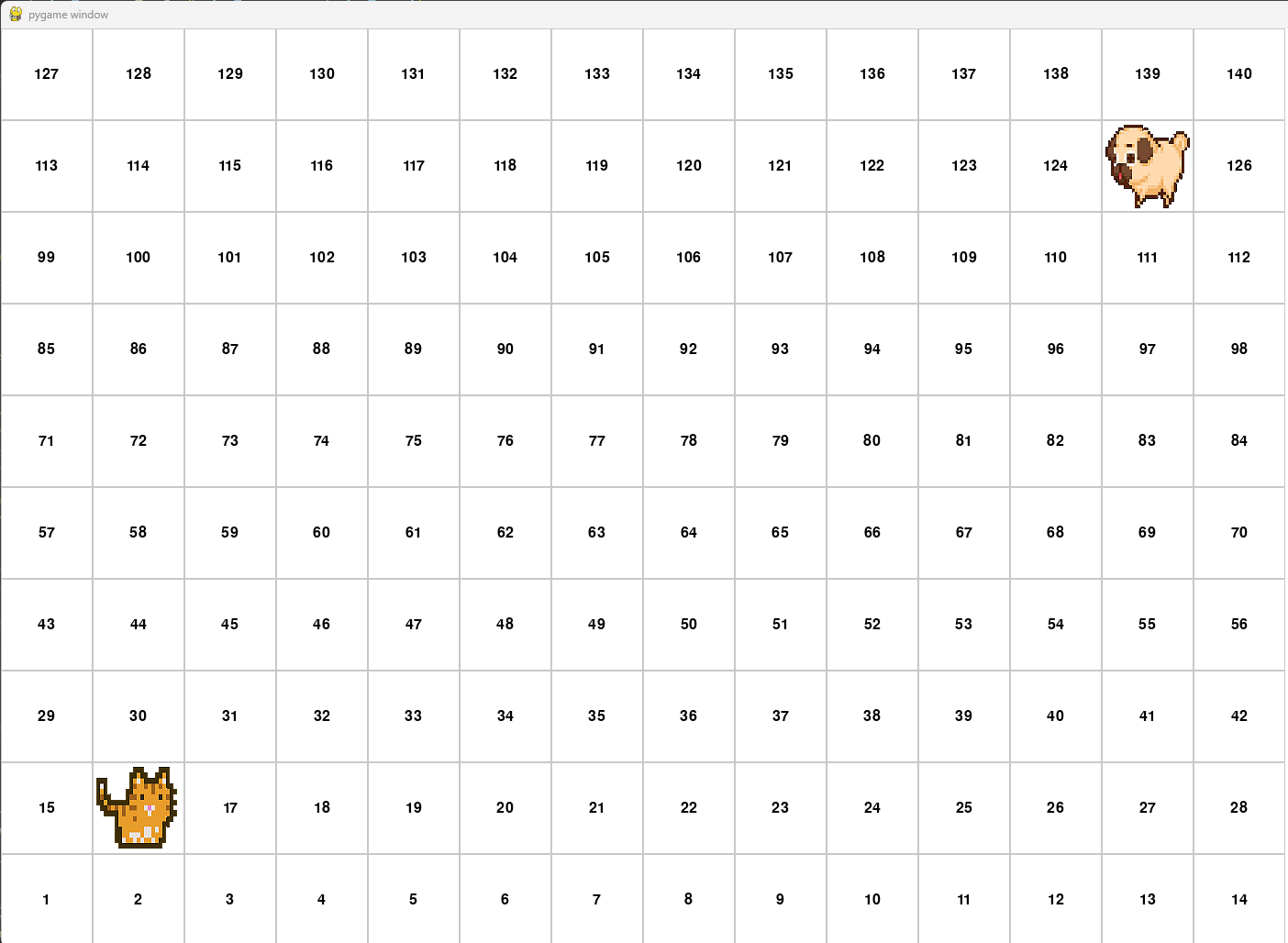
I always wanted to know how hard it is to create similar type of game and why it is required to have a team of 20+ developers. After my expectations to have the terrain and everything on the map ready on day 2 were ruined, I got the answers on both questions. This target was achieved on day 5 and many others targets remained just a good wish. However, the process was a good experience and I learned some new concepts. As a side effect I was surprised that my 5yo beta tester actually enjoyed it:

5.2 Inspiration

The game was heavily inspired by the Heroes of Might and Magic series, as I spent a lot of hours in childhood and as a teenager playing it. I am not happy with the state that the series and whole genre of turn based strategy games are in the moment. A game from 1999 to be the most popular in the genre nowadays while the community is still huge is meaning that the AAA dev companies are underestimating the potential of such project.

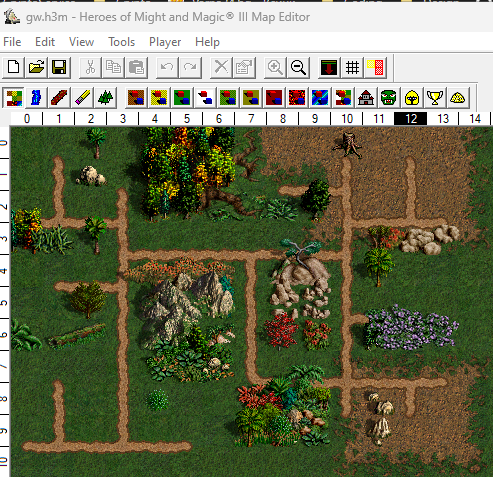
Another inspiration is the Disciples series, which came later and even I liked them a lot, I didn’t implement any of the unique ideas in making of Pet Wars.

5.3 Starting from scratch

 That moment that you are staying in the front of the empty screen is always uncomfortable. First, I decided the tiles to be with QUAD shape, which is for sure worse user experience than the HEX tiles, but much easier to implement and more appropriate considering it was my first ever PyGame project. Then with some help from internet I managed to draw the tiles and to put numbers on them:

Afterwards I implemented the terrain map, and It started to become easier and more straightforward after every feature added.

5.4 Artwork Notes

All assets are transparent png images randomly found on google. I tried using AI for generating the artwork or ready tilesets, but the result was unsatisfactory. For the map, my first option was Tiled. There are a lot of specifics which were stopping me to use pre-ready tile sets. Due to the limited time and the complicity of Tiled I dumped the idea. However, it still could be a good base for future issues! My second choice was game maker. I have used it many years ago but wasn’t expecting it to evolve so much. Finally considering the limited time for developing I decided to move to the next option. HoMM 3 map editor had everything I needed and was super easy to use. Even if it not meant for this purpose, the result was great. However, this needs to be changed to one of the previous 2 options if I want to further develop the project.

5.5 Challenges

The biggest challenge was the time management. The target was set too high considering the available time and my current experience with Python and PyGame. I had to prioritize and cut off a lot of ideas, leaving them for later implementation.

The artwork took much more time than I expected, even if it is just ripped from google. I had to try a lot of different approaches for the map creation which resulted in loss of time, but also gave me valuable experience.

After the first code review I followed the advices and refactored big pieces of the code. Few days lost, but I found out how easy is to maintain the code when using dictionaries instead of if else structure or switch/match case.

Looking at the empty screen is always hard time for the junior developers. I wrote enough earlier, so I am passing to the next item.

The battle system – this is like game in the game. I am currently working on it, so fingers crossed this is going to be completed soon at least in some basic shape.

Game lagging – nothing special here, just getting random images and putting them in the game is never a good idea. The lagging was so bad that It was taking about a second for the hero to move. At some point I spent the time to edit every single sprite (with 1 exception) so this is now resolved.

5.6 Future Development

The battle system is the first thing on the list as it is currently not existing. Even If I have clear vision how it should look like I have to write it in code.

The farms are currently having no visual indication of who is the owner. This could be easily implemented but again it needs some art work.

AI for the enemy hero is from critical importance to make the game playable. For a starter I would implement the maze algorithm that we did during the course, of course with slight modifications.

The moving mechanics needs to be changed in order to make the user experience much more enjoyable. What I mean is to actually draw to road to the target, and then the moving will start with the second click. Also, the cursor needs to change if there is event on the target accordingly.

Quad tiles are easy. However easy in the most cases means worse UX. Hex is the correct type of tile for this type of games, giving much more depth on the adventure map and the battlefield. Also, it represents the spaces much more accurate considering the diagonal movements. However, I can see a lot of modern developers taking this shortcut and using quad tiles in high budget projects which is a shame.

Drawing own artwork could be part of the plan only if there is a team including a designer. Still, it is important area for improvement if the project continues development.

Full screen mode is another one, I believe everyone is feeling more comfortable to play in full screen. The resolution needs to be reworked as well to fit at least 16:9 aspect ratio.

The project can be expanded in a lot of areas like maps, units, towns, fractions, campaign mode etc. The options here are unlimited.

I also believe PyGame is great for learning experience, but not preferred environment for big projects. So if the scale of the projects becomes unmanageable by PyGame, then I will have to transfer is to different environment and probably different language.

6.Credits and download link

I will just leave some links in case you want to contact me or follow my other projects:

Get the game from: <https://github.com/hristogwivanov/PetWars>

GitHub Profile: <https://github.com/hristogwivanov>

LinkedIn Profile: <https://www.linkedin.com/in/hristogwivanov/>