**Programming Journey – Escaperoom ‘Where is my Emma?’**

**What additional things (libraries, paradigms, etc) did you learn during the project?**

The main new aspect we learned is the Pygame library. We learned how to display a frame and ‘blit’ images on it. We learned how to handle userinputs or pygame.events. Keyboardinputs as well as mouseclicks). Further because of the high interactivity we learned the use of threads in python. The concept of thread was already known (‘Einführung in die Sotwareentwicklung’), so we fastly came to the decision to use them for handling simultaneous processes.

With a (for us) new app ‘Poti poti’ we learned to design pixel-images and display them. The pixel look helped a lot by identifying the placement of the many ‘buttons’. With paint we could reduce all images to our wanted resolution in pixel\*pixel. For design aspects we also learned to use/load a .tff font and display it on our game.

We learned how to handle blocking user-input (with blocking while-loop) grafically.

Global variables. You need to state at the beginning ‘gloabal x’ if you want to change it in this function. Acess is possible by only reffering to x.

[For the scientificier part of the program I learned to connect matplotlib with pygame and display plots there]

**What challenges did you face?**

One challenge we faced, was an updating problem. We learned that if file ‘a.py’ has a global variable that has been changed, file ’b.py’ that has an import statement at the beginning: ‘from a import \*’ the variable is not automatically updated in file b.py ’ after the change.

We approached this problem by printing this variable in both files and recognized asychnority.

First we tried to import it again (after the change was made) and it worked. Later we solved that problem more elegant by implementing a getter-function (modularity and visibility reasons).

Another problem with the import was the meaningfull connection between several .py files. This was (spätestens) clear as we learned that you cannot import file b.py in a.py and vice versa (simultaneously) which is logical because then you could write everthing to one file. We approached this by splitting the progam concepts in files startscreen, endscreen and door1 door2 door3 at the beginning. After implementing and some redundancy we implemented display\_components which holds functions and status variables that are used by every file (or the majority of files). The ‘bigger’ function handle\_userinput was put in one extra file (could also be implemented in display\_components)

A further but quickly solved challenge was the simultaneous gameplay an time displaying. Because (as already stated in the beginning) the concepts was know we used threads. One (probably the main one, like in java) handles the gameplay and user-input and the other one the display of the time. First (naive) approach was that everything is handled by the mainprocess but even after qsetting the fps to 60 one second was incremented every three (or so) because that thread had so much other to handle inbetween.

**What further additions could be made?**

I think the first obivious addition could be increasing the complexity of the rooms task (like in real escaperoomgames) or add levels/new rooms. If that is not wanted, one could also add a look-around function which lets you look around in the room from different perspectives (also like in real escaperoom games). We already implemented some zooming functions so equally we could let the user ‘turn around’ to look at other things in the rooms.

We also have implemented one task, were the key to the next door is collected but neither acknowleged or displayed, so here could an itembox be added that displays it. Furthermore we had the idea of an Avatar which is seen in the loading screen or the final\_words room, but we did not really embedded it. Here is room to improvement.

* Pygame library (load and display images and pygame.rects, handle events (keyboardinput as well as mouseclicks)
* ‘threads’ ( known from ‘Einführung in die Softwareentwicklung‘ but new usage in Python)
* Paint is cool. You can reduce picture to an exact amount of pixels there 😊
* Use of an own .ttf font (pokemon.ttf) to display text
* (soundeffects probably)
* (for scientific: connecting pygame and matplotlib)
* Handle userinput (with blocking while-loop) graphically
* Paradigm: update problem if file a.py has a global variable that has been changed file b.y that has an import statement at the beginning ‘from a import \*’ after the change the variable is not automatically updated in file b.py.
* > approach: we found this out by printing the variables in both files then tried to import the file again after change (which worked) and then implemented a getter (because of modularity und visibility-standards)
* Connecting several .py file through import statements (meaningfully) + you cannot import a in file b and b in file a simultaneous
* > approach: at beginning split in files startscreen, endscreen and door1 door2 door3. After implementing and some redundancy we implemented display\_components which holds functions and status variables that are used from every file. The ‘bigger’ function handle\_userinput was put in one extra file (could also be implemeten in display\_components)
* Simultaneous gameplay an time displaying
* -> threads one (probably the main one, like in java) handles
* One click is sometimes are evaluated as more (tryied to us gameplay, other handles time displaying
* Approach: blocking functions e.g. loops or time.sleep() to prevent this)
* More rooms
* Look around in the round (not only one perspective)
* Soundeffects
* Avatar
* Item inventory

Finale Fragen an Tjark/Robin:

Schriftliche aufgaben auf deutsch okay? (kommentare würde ich englisch lassen /also generell das spiel)

Einfache getter/setter one docstring okay? Zu beginn # the follwing functions…

Es ist wie du vermutet hast eine recht lange if-else verschachtelung geworden, laut dem punkt ‘appropriate use of pythons features, don’t force things that make no sense’ fühlt sich das aber falsch an. Kannst du vielleicht nochmal dazu sagen, ob das wirklich okay ist? (für jeden knopf der gedrückt werden kann gibt es einen fall (da ist nämlich nicht wirklich nen zusammenfassen möglich da jeder nopf eine andere function auslöst)

Desweiteren haben wir die knöpfe durch ausgeben der koordinaten ermittelt (screenshot einfügen?)

Was sollen die results sein? Screenshot/video?

Größtes problem bei langen clicks werden diese als mehrere gewerten bei ‚sesiblen‘ gebraucht aber nicht. Blockierende functionen machen das spiel teils lang. Ist es okay wenn der sensible gebrauch gefordert ist?

Robin: scientific teil genung?