

ASK – LABORATORIUM

SPRAWOZDANIE

ZADANIE 4

1. Zadanie

Tematem zadania jest napisanie aplikacji, będącej symulatorem stanowiska dyspozytorskiego „linii produkcyjnej”. Program ten ma zawierać elementy diagnostyki nadzorowanego „procesu produkcji” jak i autodiagnostyki operatora czuwającego nad prawidłowym przebiegiem „produkcji”.

Wykorzystując dostępne informacje na temat parametrów pracy komputera PC (np. temperatury rdzenia procesora, stopnia wykorzystania procesora, prędkości obrotowych wentylatorów itp.) oraz generatory liczb losowych i timery, zasymulować parametry kontrolowanego „procesu produkcyjnego”. Należy przewidzieć obsługę pojawiających się losowo awarii oraz przekroczeń granicznych wartości wybranych parametrów procesu - np. po przekroczeniu granicznej temperatury obudowy silnika należy włączyć dodatkowy wentylator lub zwolnić tempo pracy linii produkcyjnej itp. O wszystkich wyjątkowych zdarzeniach i wymaganych działaniach, operator musi być informowany za pośrednictwem odpowiedniego zestawu komunikatów. Program powinien zawierać okno logowania do aplikacji i na bieżąco badać obecność oraz „przytomność” operatora. Element autodiagnostyczny powinien polegać na okresowym pojawianiu się komunikatu informującego o konieczności potwierdzenia obecności przez wciśnięcie wybranego klawisza. W przypadku braku potwierdzenia, np. przez co najmniej 30 sekund, powinno nastąpić uruchomienie alarmu i wylogowanie operatora z systemu.

W celu urozmaicenia zadania, zamiast fabryki, produkcji i taśmy produkcyjnej, w naszej aplikacji wykorzystaliśmy podobne mechaniki z inną szatą graficzną – w ten sposób powstał autorski ‘symulator życia’. Aplikacja została napisana w języku python.

2. Techniczne założenia projektowe

- aplikacja powinna zawierać okno logowania
- wykorzystywanie timerów i generatorów liczb pseudolosowych
- losowe awarie i przekraczanie granicznych wartości parametrów procesów
- informowanie użytkownika o zdarzeniach za pomocą wyskakujących komunikatów
- badanie aktywności użytkownika i wylogowanie go w przypadku jego braku

3. Autorskie założenia projektowe

Gra 'symulator życia' jest aplikacją, w której użytkownik wciela się w przeciętną osobę i stara się przeżyć przez dbanie o potrzeby takie jak: energia, głód, zabawa, higiena czy komfort.

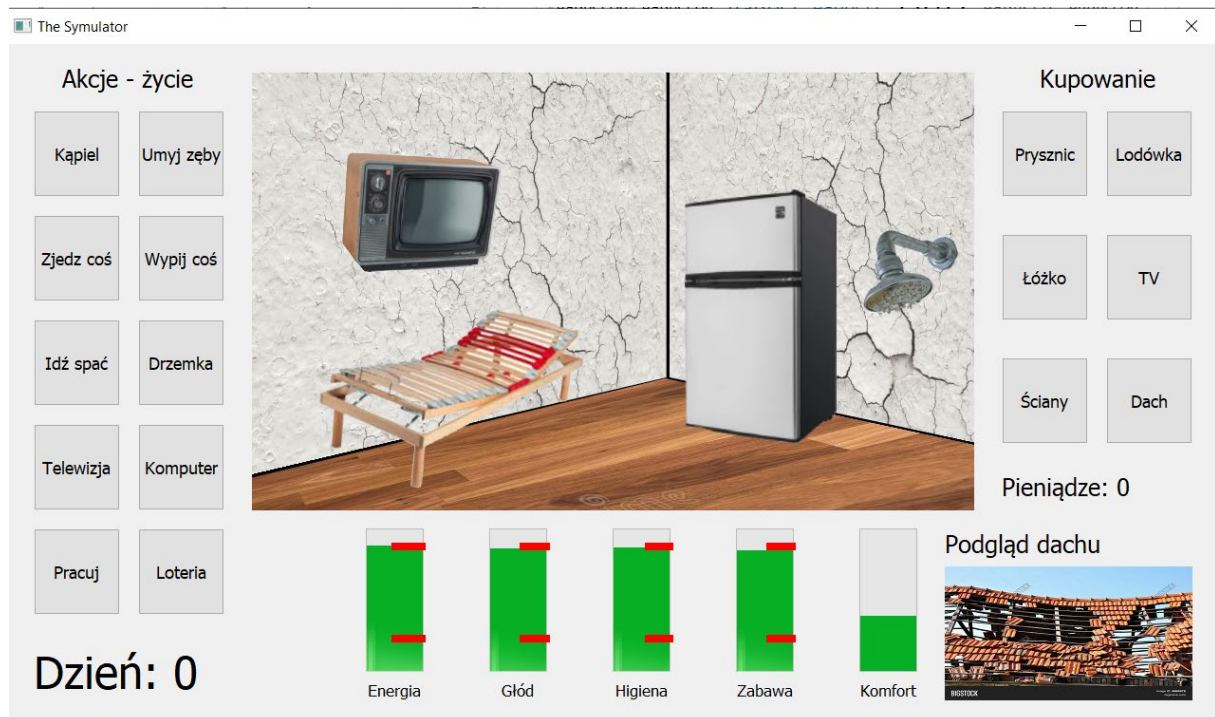
- losowe awarie przedstawiają losowe zdarzenia w życiu (w tym wypadku powódź lub kradzież)
- przekraczanie granicznych wartości parametrów (zbyt niskie potrzeby)
- informowanie użytkownika (użytkownik dostanie wiadomość o wymienionych wyżej zdarzeniach)
- badanie aktywności użytkownika (sprawdzone w postaci potrzeby płacenia rachunków co 30 dni – zmiana dni ustawiana jako timer inkrementujący daną zmienną int liczby dni co kilka sekund)

3. Działanie programu



Rys.1 Okno logowania programu

Na starcie programu zostaje inicjalizowane okienko logowania z dwoma polami tekstowymi oraz przyciskiem 'Zaloguj mnie!'. Po wpisaniu nazwy użytkownika i hasła, użytkownik przechodzi do symulatora, w przeciwnym razie program mówi o braku lub niepoprawnym hasle lub nazwie.

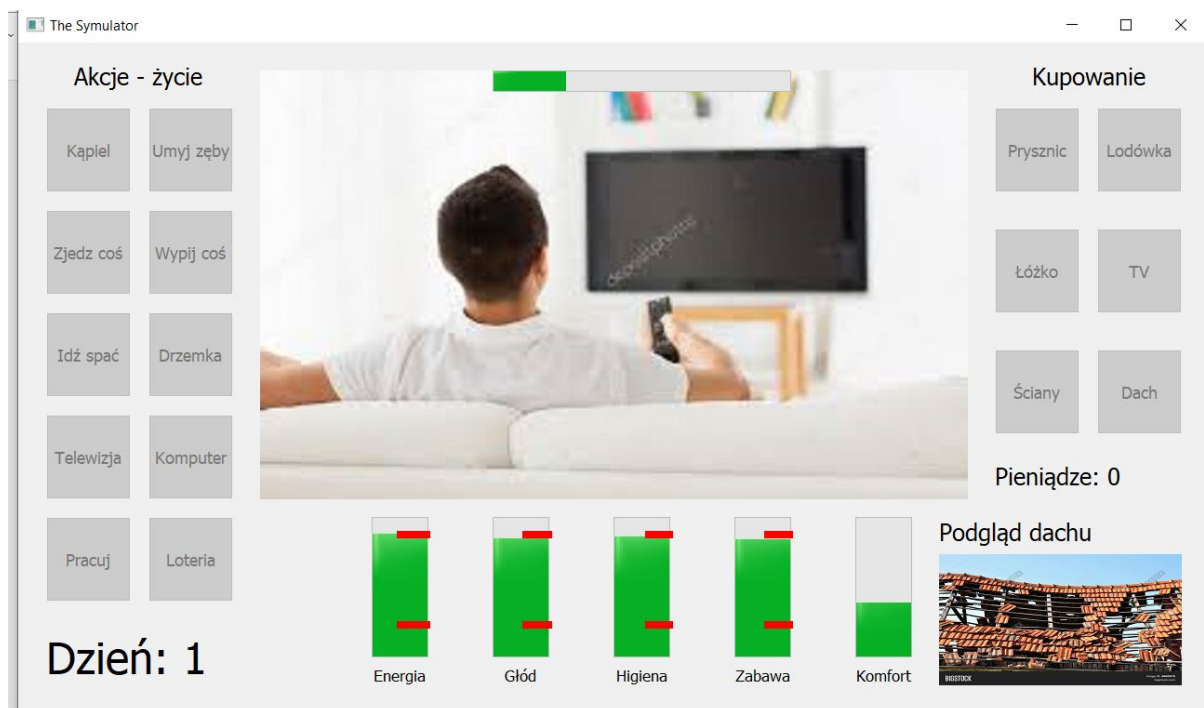


Rys.2 Właściwe okno symulatora

Główne okno – po lewej i po prawej stronie widać przyciski typu 'Push Button', etykiety z informacjami o częściach okna oraz obrazki i na dole (potrzeby) Progress Bary – każdy działa na innym wątku.

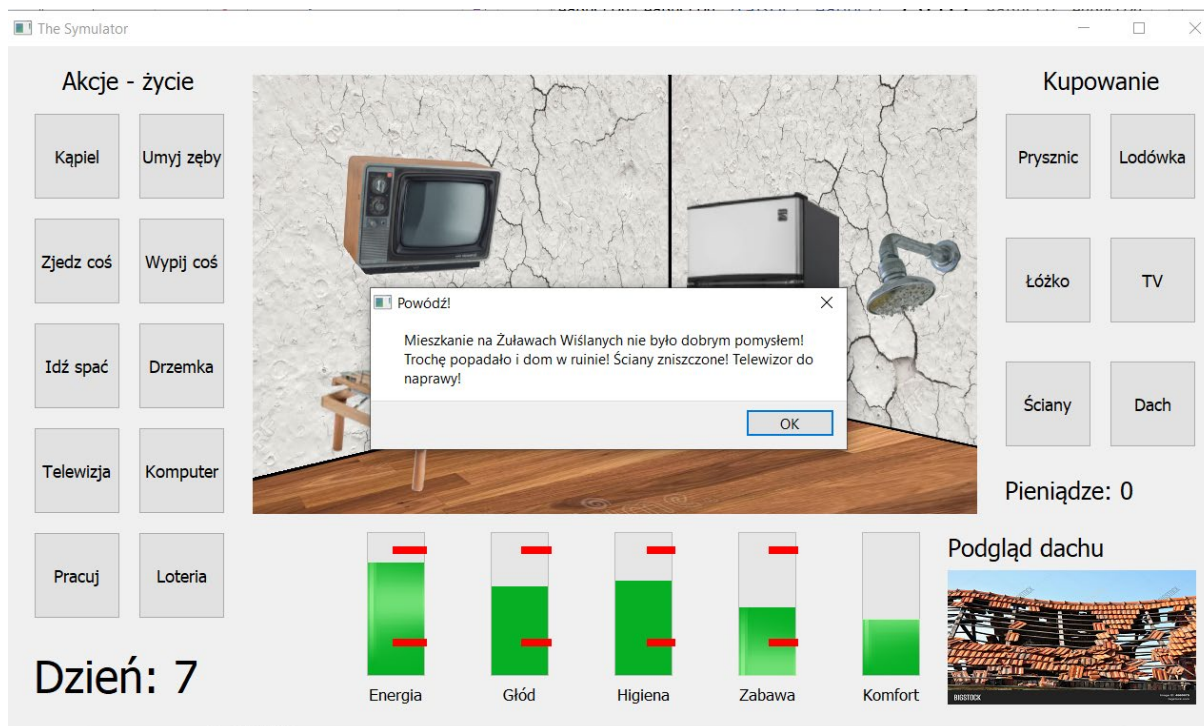
Po pokazaniu głównego okna – potrzeby, czyli Progress Bary zaczynają się zmniejszać, każdy z inną, ustaloną częstotliwością. Wyjątkiem jest potrzeba 'Komfort', jest ona wyliczana za pomocą zależności pomiędzy paroma czynnikami; każda potrzeba powyżej dolnego czerwonego znacznika dodaje 10% do komfortu, dodatkowo ulepszanie każdej części domu za pomocą przycisków po prawej stronie dodaje kolejne 10%. Im wyższy komfort, tym wolniej spadają inne potrzeby, dlatego ulepszanie domu i dbanie o potrzeby jest ważne.

Przyciski z lewej strony służą do odnawiania potrzeb. Zasada działania jest prosta, przykładowo po kliknięciu przycisku 'Telewizja', pokaże się obrazek człowieka oglądającego telewizję oraz progress bar odliczający 4 sekundy, a następnie 'Zabawa' wzrośnie znacznie. Można też wybrać krótszą akcję, w tym wypadku 'Komputer', która zajmie 2 sekundy, ale tym razem 'Zabawa' wzrośnie o mniejszą wartość.



Rys.3 Okno po kliknięciu akcji 'Telewizja'

Istnieją też akcje 'Pracuj' oraz 'Loteria', które nie zwiększają potrzeb, natomiast dodają odpowiednio 2000 i 500 pieniędzy za wykonaną akcję. Dzięki nim można ulepszać dom za pomocą przycisków z prawej strony okienka.



Rys.4 Przykładowe wyskakujące okienko

Wyskakujące okienka odgrywają ważną rolę w programie – informują użytkownika o stanie gry i losowych zdarzeniach. Na rys.4 pokazano zdarzenie ‘Powódź’, które psuje ulepszony telewizor i ściany (sprowadza je do pierwotnego stanu w ten sposób obniżając komfort). Innym losowym zdarzeniem jest kradzież. Zostajemy ograbieni z naszych oszczędności, pieniądze spadają o 500, jest to również ważna mechanika, ponieważ co 30 dni musimy płacić rachunki o wartości 2000, bez nich przegrywamy i zostajemy wylogowani.

Jeśli potrzeby spadną poniżej dolnych czerwonych znaczników, również wyskoczy okienko informujące nas o stanie danej potrzeby.

Mechaniki gry i wykorzystane elementy są proste i nie wymagają ilustracji w postaci kodu.

4. Wnioski

Aplikacja spełnia założenia przyjęte w punkcie nr 1 i 2 sprawozdania. Do utworzenia okienek aplikacji została użyta biblioteka PySide2-Qt, Qt jest standardem dla wielu nowoczesnych aplikacji i posiada wsparcie dla wielu języków programowania.

Aplikacja składa się z prostych elementów, ale mimo to zajmuje użytkownika na dłuższy czas.

Kod jest podzielony na trzy skrypty oraz używa dziedziczenia klas – jeden od Progress Barów i rozdzielania ich na wątki, skrypt graficzny GUI oraz okno właściwe z mechanikami.

Do wad aplikacji można zaliczyć poziom trudności, zdecydowanie należy dopracować prędkości zmniejszania wartości potrzeb. Przy dalszym rozwijaniu aplikacji warto też zastanowić się nad większą ilością losowych zdarzeń.