
Programowanie współbieżne i rozproszone 2019/20

Dominik Kołodziej, Kamil Skomro
3 rok, Informatyka EAIIB, grupa 1a

Data oddania: 17.01.2020r.

System sterowania żaluzjami

Cel programu	2
Opis i schemat struktury zadaniowej programu	2
Struktura zadaniowa programu:	2
Diagram zależności	3
Krótką instrukcja obsługi	3
Wymagania	3
Instalacja i uruchomienie	3
Przykład użycia	4
Możliwe rozszerzenia programu	5
Ograniczenia programu	6

Cel programu

Celem programu jest symulacja systemu sterowania żaluzjami, poprzez użycie rozproszonych procesów udostępnianych przez język Erlang.

Opis i schemat struktury zadaniowej programu

Struktura zadaniowa programu:

moduł(zadanie) blinder - jest odpowiedzialny za stan pojedynczej żaluzji. Każda żaluzja ma swój identyfikator, aktualny poziom rozwinięcia rolety, który ma być wyrenderowany, oraz poziom docelowy rozwinięcia do którego będzie zmierzać z każdą iteracją;

moduł controller - jest odpowiedzialny za kontrolowanie rolet. Umożliwia on zamknięcie wszystkich rolet na raz oraz zapamiętanie poziomów rolet przed wykonaniem tej operacji, tak aby po otwarciu wróciły one do stanu przed zamknięciem;

moduł print - posiada funkcje umożliwiające rysowanie po siatce terminala jak i funkcję wyświetlającą menu;

moduł render - jest odpowiedzialny za wyświetlanie rolet jak i bieżącej temperatury słońca;

moduł sensor - jest odpowiedzialny za zamykanie wszystkich rolet w sytuacji, gdy temperatura słońca będzie wyższa niż zdefiniowana w stałej `TEMP_TRESHOLD`, oraz powtórному otwarciu jeśli temperatura spadnie poniżej tego progu;

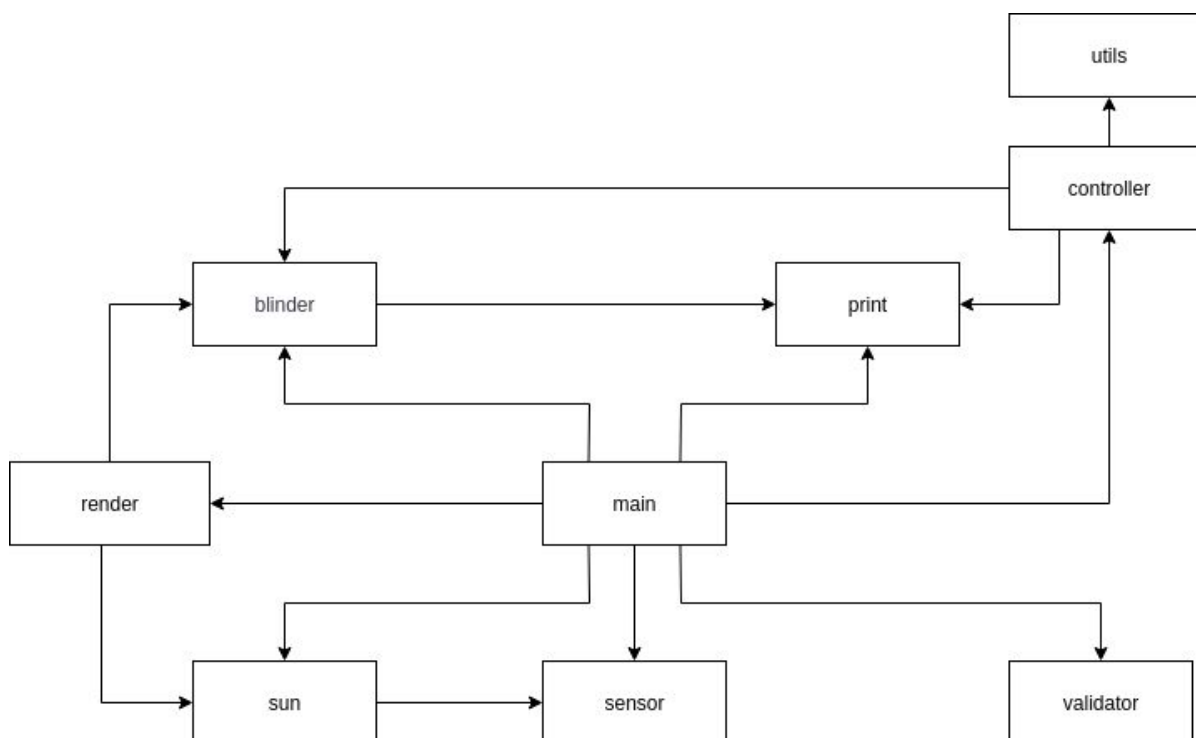
moduł sun - jest odpowiedzialny za symulację słońca. Wartość temperatury jest losowana z przedziału 0 .. 100. Słońce zmienia swoją temperaturę z interwałem zdefiniowanym w zmiennej `SUN_INTERVAL_SECONDS`;

moduł utils - posiada funkcje pomocnicze;

moduł validator - posiada funkcje pomocne przy przetwarzaniu komend użytkownika

moduł main - jest odpowiedzialny za skompilowanie innych modułów, oraz uruchomienie odpowiednich procesów. Na początku mamy możliwość podania ile rolet chcemy symulować.

Diagram zależności



Krótką instrukcja obsługi

Wymagania

- System operacyjny Unix
- Zainstalowane środowisko Erlang - <https://www.erlang.org/>

Instalacja i uruchomienie

1. Sklonowanie repozytorium lokalnie z linku: <https://github.com/skomkami/blinders>
Można również ściągnąć archiwum .zip, a następnie je rozpakować.
2. Otworzenie terminalu w folderze src/
3. Uruchomienie konsoli erlanga komendą
`erl`
4. Skompilowanie pliku main.erl
`c(main).`
5. Uruchomienie funkcji start
`main:start().`

[illegible]

Możliwe rozszerzenia programu

Interfejs użytkownika w terminalu jest mało wygodny w użyciu, zapewnia jedynie wyświetlanie danych w płaszczyźnie po współrzędnej X i Y. W związku z odświeżaniem terminala co sekundę i rysowaniem panelu na nowo powstają artefakty i wpisywany input pojawia się w randomowych miejscach na ekranie.

Rozwiązaniem byłby podział programu na back-end w Erlangu i front-end w dowolnym frontendowym frameworku np. React. Wtedy back-end wystawiałby endpoint, pod którym znalazłyby się informacje potrzebne do rysowania rolet poprzez front-end.

Ograniczenia programu

Program ze względu użycia komend to wyświetlania właściwych tylko dla terminalu systemów Unixowych (Ubuntu, Mac OS).