Politechnika Śląska

Wydział Automatyki Elektroniki i Informatyki

Kierunek Automatyka i Robotyka

Projekt Inżynierski

Planowanie i symulacja wybranego systemu produkcyjnego

Autor: Wojciech Zysk

Kierujący pracą: dr inż. Jolanta Krystek

Spis treści

1. Wstęp

W dzisiejszych czasach praktycznie każdy sprzęt wokół przeciętnego człowieka jest wytwarzany masowo. Programy takie jak “Jak to jest zrobione” pozwalają przyjrzeć się bliżej procesowi wytwarzania produktów. Programy te niestety nie pokazują procesu produkcji z perspektywy maksymalizacji ilości wytwarzanych produktów a co za tym idzie, maksymalizacji zysku, a jest to przecież główny cel fabryk czy hal produkcyjnych. Studenci kierunku Automatyka i Robotyka zaczynają uczyć się optymalizacji produkcji już na piątym semestrze na przedmiocie Zautomatyzowane Systemy Wytwarzania. Przed studentami są tam stawiane problemy z którymi spotykają się fabryki (np. Problem identyfikacji “wąskiego gardła”) oraz studenci uczeni są jak te problemy rozwiązać, stosując zależne od sytuacji algorytmy (np. Algorytm McNaughton`a).

Niniejsza praca porusza dwa zagadnienia. Pierwszym z nich jest tworzenie oprogramowania w języku Java, używając Framework`a JavaFX oraz oprogramowania ułatwiającego tworzenie elementów graficznych: Scene Builder. Jest to program typu “Drag & Drop” umożliwiający stworzenie widoku aplikacji w osobnym pliku z rozszerzeniem fxml. Został on stworzony specjalnie dla Frameworku JavaFX, co umożliwia automatyczne tworzenie kodu w pliku fxml, a nie w klasie java, jak jest to w przypadku poprzednich Frameworków np. Java swing.

Drugim zagadnieniem jest sama optymalizacja procesu produkcyjnego, w którym problemem jest tzw. „wąskie gardło”. Skutkiem występowania “wąskiego gardła” są zastoje produkcyjne. Przykładem takiej sytuacji może być np. wykwalifikowany pracownik, który zachoruje na kilka dni.

2. Cel i zakres pracy

Celem niniejszej pracy było stworzenie aplikacji desktopowej w formie gry produkcyjnej, umożliwiającej planowanie i optymalizację pewnego procesu produkcyjnego. Gra została stworzona w języku Java.

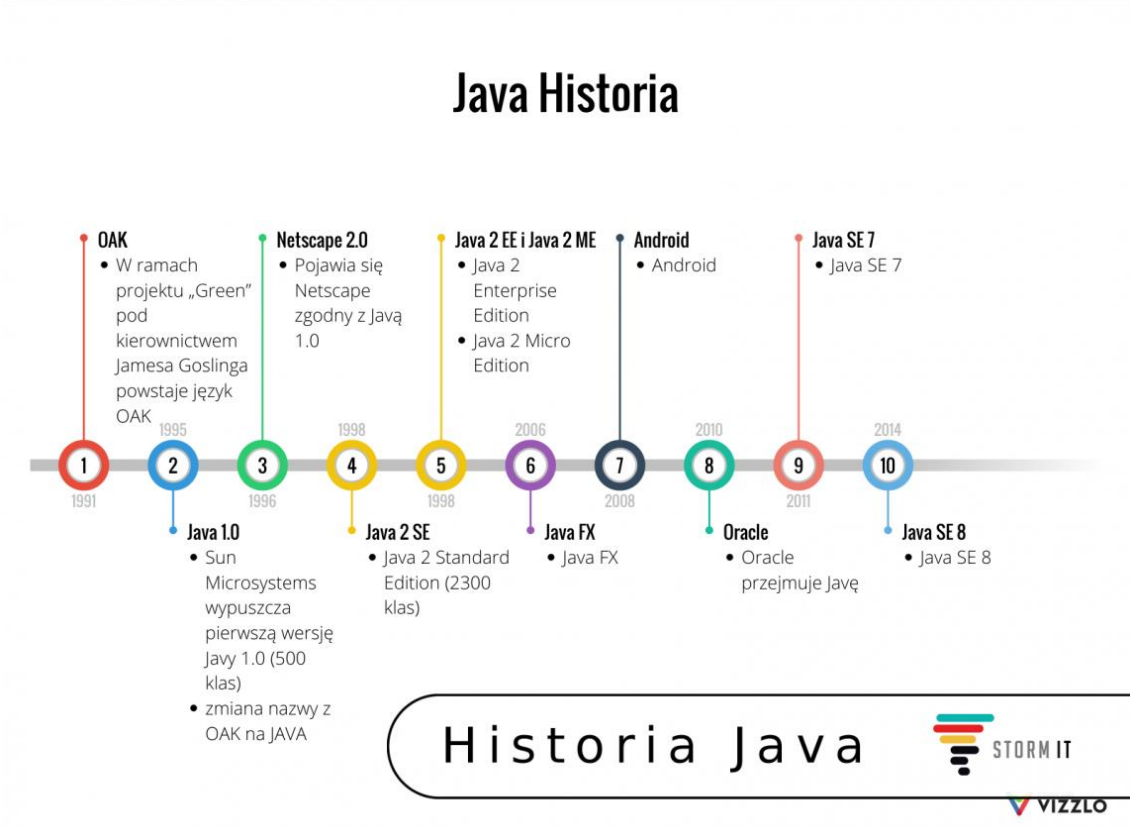
Zakres pracy obejmował:

- Przegląd literatury na temat tworzenia aplikacji desktopowych w języku Java oraz Frameworku JavaFX,
- Przegląd literatury na temat procesu produkcyjnego oraz problemu identyfikacji wąskiego gardła,
- Zaprojektowanie gry (wybór liczby maszyn, liczby stacji, czasów przezbrajania, czasów produkcji i kosztów materiałów) i celów gry,
- Stworzenie aplikacji według wybranych celów,
- Testy aplikacji.

Praca zawiera 6 rozdziałów merytorycznych. Rozdział trzeci opisuje krótko język Java oraz użyte narzędzia programistyczne. W następnym rozdziale poruszana jest tematyka procesu produkcyjnego, systemów przepływowych oraz Teorii Ograniczeń. Rozdział piąty opisuje stworzoną w ramach pracy inżynierskiej aplikację, proces produkcyjny który aplikacja przedstawia, interfejs aplikacji oraz stronę programistyczną pracy. Rozdział szósty przedstawia przykładowe strategie, które można przyjąć w rozgrywce. W rozdziale siódmym podsumowano prace. Rozdział ósmy jest spisem wykorzystanych źródeł literaturowych i internetowych.

3. Wykorzystane narzędzia programistyczne

Java jest w pełni obiektowym, wysokopoziomowym językiem programowania. Jest najczęściej używana w backendowych systemach aplikacji internetowych. Jest więc odpowiedzialna za wszystko co dzieje się “w tle”. Można jej też użyć do tworzenia oprogramowania desktopowego (JavaFX czy swing) jak i również gier (LibGDX). System operacyjny Android również działa w środowisku Javy, co umożliwia tworzenie aplikacji oraz gier na telefony z systemem Android.



Rysunek 3.1 Historia i rozwój języka Java [2]

Początek języka Java można określić na rok 1991. Wtedy firma Sun z Patrickiem Naughtonem oraz Jamesem Goslingiem na czele, postanowili stworzyć prosty i niewielki język, który mógłby być uruchamiany na wielu platformach z różnymi parametrami. Projekt zatytułowano Green. Pomysł rozwijano i doprowadzono do stworzenia języka, który poddawany był kompilacji do kodu pośredniego a dopiero ten

był uruchamiany. Pozwoliło to na uruchamianie jednego kodu na wielu urządzeniach i systemach operacyjnych wyposażonych w interpreter języka - wirtualną maszynę Javy. Choć nie był to pierwszy język, który działał na takiej zasadzie, to został on entuzjastycznie przyjęty przez społeczność programistów. [1]

3.1. JavaFX

JavaFX została stworzona jako następca swinga i jest ona rekomendowaną technologią do tworzenia GUI (*graphical user interface)* przez firmę Oracle. Technologia ta daje możliwość stworzenia interfejsu zarówno w języku Java, jak i przy wykorzystaniu plików FXML. Użycie plików FXML pozwala na znacznie wygodniejsze stworzenie statycznego interfejsu, choć oczywiście stworzenie dynamicznych elementów dalej będzie odbywało się w klasach języka Java. JavaFX pozwala również na łatwą obsługę języka CSS, dzięki czemu można osiągnąć nowoczesny wygląd aplikacji. Jest to Framwork wprowadzony w Javie 8 jako podstawa, jednak pierwsze wersje pojawiły się już w roku 2006 i bazowały na języku skryptowym JavaFX Script.

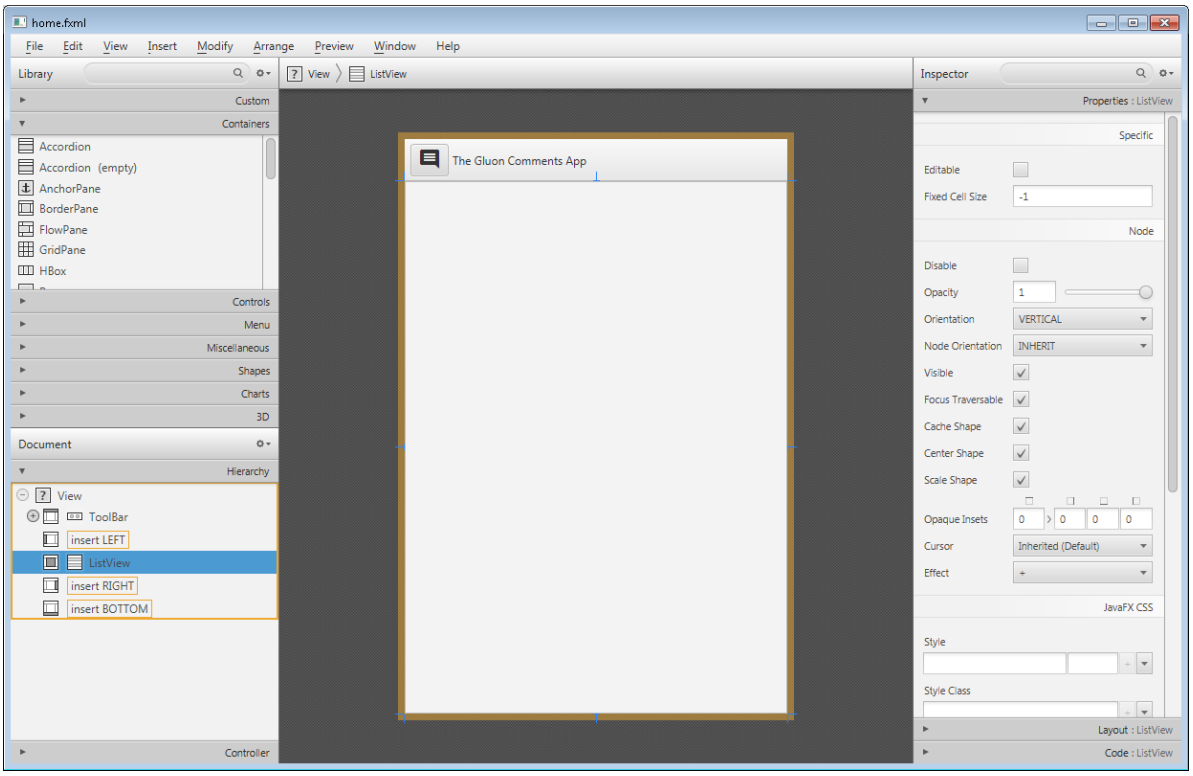
JavaFX — najważniejsze informacje:

- Umożliwia programistom integrację grafiki wektorowej, animacji, sieciowych zasobów dźwiękowych oraz wideo w procesie tworzenia bogatych, interaktywnych i złożonych aplikacji.
- Rozszerza technologię Java, umożliwiając korzystanie z dowolnej biblioteki Java w aplikacjach JavaFX.
- Zapewnia wydajny przepływ prac na linii projektant-programista z wykorzystaniem narzędzia Project Nile: projektanci mogą pracować w dowolnych narzędziach, jednocześnie współpracując z programistami. [3]

3.2. Scene Builder

Scene builder jest oprogramowaniem służącym do prostego tworzenia statycznych elementów interfejsu, który zapisywany jest w osobnym pliku fxml. Jest to program typu “Drag and Drop” co znaczy, że wybrane elementy zostają przeniesione na

interfejs, a program sam pisze odpowiedni kod (np. rozmieszczenie, rozmiar obiektu, czy margines). Scene Builder umożliwia również edytowanie elementów poprzez użycie komend języka CSS. Został on stworzony przez firmę Oracle, a później porzucony i przejęty przez firmę Gluon, która do teraz zajmuje się jego rozwojem.



Rysunek 3.2 Okno Scene Builder`a

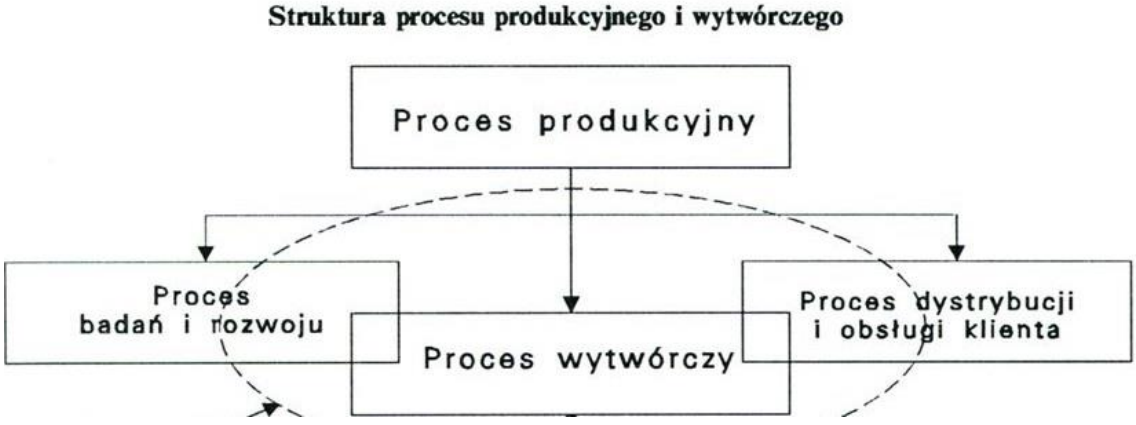
3.3. Eclipse

Eclipse jest platformą napisaną w 2004 roku w Javie i służy do tworzenia aplikacji. Na bazie Eclipse powstało środowisko programistyczne do tworzenia programów w Javie, które jest rozpowszechniane razem z platformą. Eclipse został stworzony przez firmę IBM, a następnie przekazany na zasadach Open Source (otwarte programowanie), co oznacza między innymi, że każdy użytkownik ma dostęp do kodu źródłowego. Powoduje to np. szybsze rozwijanie oprogramowania, ponieważ programiści z całego świata mogą uczestniczyć w procesie optymalizowania i ulepszania kodu. Eclipse obecnie rozwijany jest przez Fundację Eclipse [4]. Eclipse oferują obsługę wtyczek,

ponieważ sama platforma nie oferuje narzędzi do tworzenia kodu oraz budowania aplikacji. Wtyczki jednak umożliwiają m.in. tworzenie kodu w językach Java, C/C++, PHP czy współprace z serwerami aplikacji.

4. Proces Produkcyjny

Proces produkcyjny jest uporządkowanym ciągiem działań zaprojektowanym i zorganizowanym do realizacji określonego, pojedynczego wyrobu lub grupy wyrobów. Proces produkcyjny przebiega według zaprojektowanej reguły działania w określonych przedziałach czasu, zmieniać się jednak mogą charakterystyki ilościowe i jakościowe, zasilenia materiałowe, energetyczne i informacyjne. Świadczy to o dynamicznym charakterze przebiegu procesu. Proces produkcyjny powinien być opłacalny, czyli realizowany przy maksymalizacji efektu produkcyjnego w określonych warunkach. Warunkiem koniecznym realizacji procesu produkcyjnego jest przepływ materiałów, informacji, czynników energetycznych czy ludzi (w tym wypadku ruch). Na proces produkcyjny składają się takie procesy jak: proces badań i rozwoju (realizowane są tam prace badawcze, projektowe oraz wdrożeniowe), proces dystrybucji i obsługi klient oraz proces wytwórczy. [5]



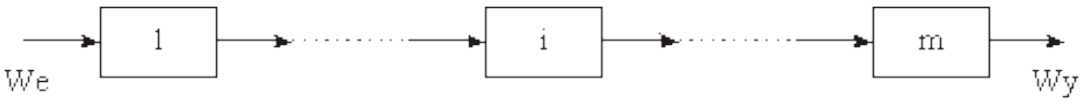
Rysunek 4.1 Struktura procesu produkcyjnego [6]

4.1. Proces wytwórczy

Struktura procesu wytwórczego jest układem faz i procesów: magazynowych, transportowych, technologicznych oraz kontrolnych. Procesy te występują w ramach faz, a także operacji produkcyjnych wraz z powiązaniami materiałowymi, energetycznymi i informacyjnymi, umożliwiającymi przepływ materiałów i półwyrobów przez stanowiska obróbkowe i montażowe, aby możliwe było otrzymanie produktu gotowego. [7]

4.2. System przepływowy

W procesie wytwórczym typu przepływowego (flow shop) wytworzenie produktu wymaga wykonania sekwencji na stacjach uporządkowanych w ciąg zgodny z realizowanym procesem. W procesie produkcyjnym tego typu występują ograniczenia kolejnościowe. Ograniczenia te można przedstawić za pomocą grafu w postaci łańcucha operacji. Kolejność wykonywania operacji jest zgodna z kolejnością ustawienia maszyn na linii produkcyjnej.



Rysunek 4.1. Graf prostego systemu przepływowego

W przypadkach prostych, jak ten przestawiony na rysunku 4.1., kolejność wykonywania zadań jest jedna i wynika ona z ograniczeń kolejnościowych. W przypadkach bardziej skomplikowanych potrzebne jest rozpatrzenie problemu przepływowego. Polega on na ustaleniu kolejności wykonywania N różnych zadań złożonych ze skończonej liczby sekwencji operacji na M stacjach/maszynach tak, aby zminimalizować:

- Czas uszeregowania, czyli czas od chwili rozpoczęcia obsługi pierwszego zadania na pierwszej stacji/maszynie do chwili zakończenia ostatniego zadania na ostatniej maszynie/stacji,
- Czas przestoju maszyn [8]

4.3. Problem “wąskiego gardła”

Każde przedsiębiorstwo, porównując je do łańcucha posiada najsłabsze ogniwo. Jest to czynnik ograniczający przepustowość, czyli liczbę “wyrobów finalnych”, które możemy wyprodukować w określonym czasie. Wyroby finalne niekoniecznie oznaczają fizyczne produkty. Przykładem tego może być szpital, gdzie celem jest liczba wyleczonych pacjentów. Najsłabsze ogniwo ogranicza osiąganie zysku, co jest podstawowym celem procesu produkcyjnego. Ograniczenia te mogą przyjmować jedną z trzech podstawowych postaci:

- Ograniczenia techniczne – wynikają one z niedostatecznych zdolności produkcyjnych środków trwałych przedsiębiorstw. Występują one w postaci wąskich gardeł, czyli zasobów, których zdolności produkcyjne są mniejsze od zapotrzebowania.
- Ograniczenia rynkowe – brak zapotrzebowania na produkty lub usługi.
- Ograniczenia ludzkie – ograniczenia tkwiące w ludziach, w tworzonych przez nich procedurach, politykach czy strategiach postępowania.

Nauką pozwalającą na analizę systemu oraz znajdowanie i usuwanie tych ograniczeń jest Teoria Ograniczeń. Została ona stworzona w latach 70-tych XX. wieku przez dr. Eliyahu M. Goldratt’a, izraelskiego fizyka i autora licznych bestsellerów o tematyce biznesowej. Ograniczenia zidentyfikowane w procesie produkcyjnym nazywane są “wąskimi gardłami”. Można wyróżnić krótkoterminowe “wąskie gardła” oraz długoterminowe. Przykładem krótko terminowego “wąskiego gardła” może być podany wcześniej przykład pracownika korzystającego z urlopu. Długoterminowe są poważnym problemem, ponieważ działają cały czas i znacznie utrudniają produkcję. Przykładem tego może być niewydajna maszyna, na której odbywa się kilka lub kilkanaście operacji. Właśnie taki przykład “wąskiego gardła” został zaprezentowany w przygotowanej w ramach pracy inżynierskiej aplikacji. Identyfikacja “wąskiego gardła” jest kluczowa, kiedy próbuje się poprawić wydajność linii produkcyjnej. [7][9]

4.4. Sposoby identyfikacji i likwidacji wąskiego gardła

Teoria ograniczeń powołuje się na proces myślowy, systematyzujący działania mające na celu identyfikację i eliminację wąskich gardeł. Proces nazywa się pięcioma krokami skupienia.

1. Identyfikacja ograniczeń systemu – Pierwszym krokiem jest zidentyfikowanie ograniczenia. Wiąże się to również z nadaniem priorytetów ograniczeniom, w celu uniknięcia rozwiązywania mało istotnych problemów, zostawiając poważne.
2. Podjęcie decyzji, w jaki sposób ograniczenie ma być eksploatowane – Zidentyfikowany problem ogranicza w pewnym stopniu przepustowość systemu, należy zapewnić takie warunki, aby możliwości ograniczenia były maksymalnie wykorzystywane.
3. Podporządkowanie całego systemu ograniczeniu – System to połączone ze sobą elementy, zatem każda zmiana w jednym elemencie wpływa na pozostałe. By wspierać zidentyfikowane ograniczenie, pozostałe elementy muszą się temu ograniczeniu podporządkować. Na tym etapie istnieje możliwość, że zidentyfikowany problem nie będzie już dłużej ograniczeniem. Można wtedy przejść do kroku ostatniego.
4. Wzmocnienie ograniczenia – Jeśli istniejące ograniczenie dalej jest krytyczne, tylko radykalne zmiany w jego eksploatacji zwiększą wydajność. Jeśli jest to ograniczenie fizyczne, w większości przypadków wiąże się to z zakupem dodatkowej maszyny lub zatrudnieniem kolejnego operatora.
5. Powrót do kroku 1 – Kiedy ograniczenie zostanie wyeliminowane, zaczynamy proces dla kolejnego w kolejności ograniczenia. [7][9]

4.5. Algorytm Werbel – bufor – lina

Rozwiązanie jest stosowane w przypadku ograniczeń zidentyfikowanych w procesie produkcyjnym. Określane są trzy podstawowe elementy: werbel, bufor i lina. Werbel jest częścią będącą wąskim gardłem, determinującą rytm w jakim pracuje ograniczenie. Drugim elementem jest bufor, który odpowiedzialny jest za tłumienie zakłóceń w przepływie produkcji. To bufor umożliwia ciągłość produkcji. Ostatnim elementem jest lina, która synchronizuje dopływ materiałów do produkcji. Algorytm ten działa dobrze, gdy produkcja jest stabilna i przewidywalna oraz gdy istnieje jedno "wąskie gardło" dla którego tworzony jest harmonogram. [7]

4.6. Czym są maszyny

Zgodnie z dyrektywą maszynową 2006/42/WE definicją wyjściową maszyny jest: "*Zespół, wyposażony lub przeznaczony do wyposażenia w mechanizm napędowy inny niż bezpośrednio wykorzystujący siłę mięśni ludzkich lub zwierzęcych, składający się ze sprzężonych części lub elementów, z których przynajmniej jedna wykonuje ruch, połączonych w całość mającą konkretne zastosowanie."* Maszyny są więc urządzeniami z mechanizmami w kadłubie, które wykonują pracę mechaniczną lub przetwarzają energię. *[10]*

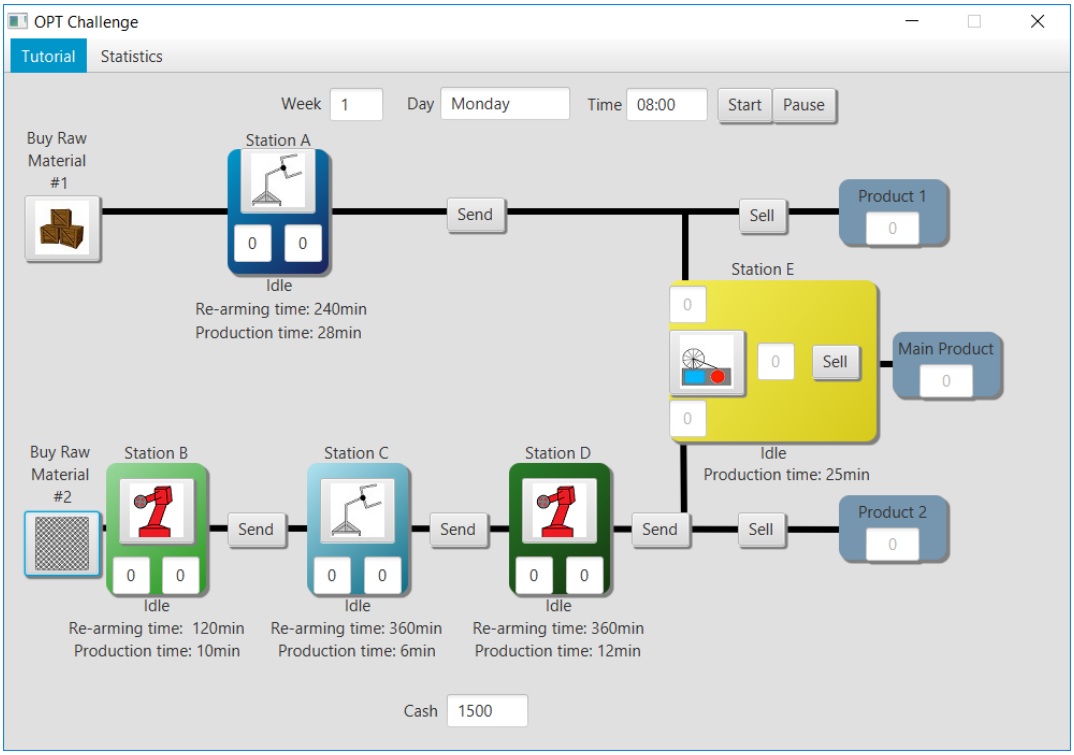
4.7. Przezbrajanie

Proces przezbrajania maszyn polega na demontażu, a następnie montażu nowych narzędzi, którymi operowała będzie maszyna. Do przezbrajania może wliczać się również: weryfikacja materiałów, centrowanie i ustawianie wymiarów oraz wykonanie próbnych egzemplarzy półfabrykatów. Czas przezbrajanie maszyny opóźnia oczywiście produkcję, dlatego celem jest osiąganie jak najmniejszego czasu przezbrojeń. Pozwoli

to na zmniejszenie wielkości partii produkcyjnych oraz ilości zapasów. Warto tutaj wspomnieć o metodyce SMED (Single Minute Exchange of Die), która pozwala na znaczne skrócenie czasów przezbrojeń. Jest to metodyka mająca na celu udoskonalić proces przezbrajania w fabryce i jest ona podzielona na 4 etapy:

1. analiza procesu przezbrajania - Polega on na przeprowadzeniu dokładnej obserwacji i przeanalizowaniu rzeczywistego procesu przezbrajania.
2. przezbrojenia wewnętrzne i zewnętrzne - Podzielenie czynności na te, które koniecznie trzeba wykonać na zatrzymanej maszynie, oraz te, które można wykonać na zewnątrz, bez zatrzymywania maszyny.
3. transformacja przezbrojeń - Przeprojektowanie procesu przezbrajania, wyprowadzając na zewnątrz kolejne fragmenty tego procesu.
4. Usprawnienia przezbrojeń - Ostatni etap metodyki SMED polega na podjęciu działań, mających na celu jak największe skrócenie czasu trwania operacji wewnętrznych. [11]

5. Opis Procesu



Rysunek 5.1 Główne okno aplikacji

Schemat fabryki z rysunku 5.1 przedstawia drogę, którą “surowe materiały” muszą przebyć, by stać się gotowymi produktami. Droga rozpoczyna się w momencie zakupu “surowych materiałów” z magazynów, co symbolizuje kliknięcie przycisku “Buy Raw Material #”. Materiały są w następnej kolejności wysyłane do pierwszych w kolejce stacji, później przetwarzane przez stacje A-E, by na końcu zostać wysłane do magazynów z produktem głównym, półproduktem nr.1 i półproduktem nr.2. Produkt główny jest tworzony z półproduktów nr.1 oraz półproduktów nr.2, w stosunku 1:1, co oznacza, że potrzeba po jednym z każdego półproduktu by stworzyć jeden produkt główny. Popyt na główny produkt jest duży, dlatego sprzedać można dowolną ilość wyprodukowanych dóbr. Półprodukty również można sprzedawać, jednak ilość którą można sprzedać jest ograniczona do ilości już sprzedanych produktów głównych. Pola A, B, C, D oraz E reprezentują stacje na których odbywają się procesy. Pola te posiadają przycisk z ilustracją maszyny, który po kliknięciu uruchamia stację, oraz dwa pola z liczbami. Pole po lewej stronie oznacza liczbę produktów przed przetworzeniem lub produktów przetworzonych na poprzedniej stacji. Pole po prawej stronie oznacza liczbę produktów przetworzonych przez stację. Kolejność operacji w fabryce zawsze jest identyczna. Przykładowo materiały w stacji C zawsze uprzednio były przetwarzane przez stację B. Przetworzone materiały wysyła się do następnej stacji za pomocą przycisku “Send”. Pod Stacją umieszczona jest informacja o stanie maszyny (bezczynna, przezbrajana, pracująca), czasie przebrajania maszyny oraz o czasie przetwarzania jednego produktu. Ilustracje na polach stacji oznaczają różne maszyny. Jak można zauważyć, Stacje A i C oraz stacje B i D posiadają tą samą ilustrację. Oznacza to, że każda z tych operacji jest wykonywana przez tą samą maszynę, jednak oczywiście nie w tym samym czasie. Po włączeniu jednej z tych stacji zaczyna się ona przezbrajać, a druga posiadająca tę samą ilustrację zatrzymuje swoją pracę.

5.1. Cele rozgrywki

Rozgrywka składa się z dwóch tygodni, w których każdy dzień roboczy rozpoczynany jest o godzinie 8.00 a kończony o godzinie 16.00. Rozgrywkę rozpoczynamy posiadając początkową wartość gotówki w wysokości 1500. Podstawowym celem gry jest osiągnięcie zysku ze sprzedaży produktów, czyli osiągnięcie statystki “Net Cash Flow” na poziomie wartości większej niż 0, po dwóch tygodniach od rozpoczęcia rozgrywki. Po każdym tygodniu z konta ubywać będzie kwota 2500, jest to kwota związana z utrzymaniem biznesu. Warunkiem rozpoczęcia drugiego tygodnia jest posiadanie gotówki na końcu tygodnia o wartości większej niż 0. Gra rozpoczyna się wraz z kliknięciem przycisku Start. Rozgrywka kończy się wygraną, gdy nasza statystka “Net Cash Flow” osiągnie wynik powyżej zera, co równa się właściwemu zyskowi ze sprzedaży produktów.

5.2. Analiza procesu produkcyjnego

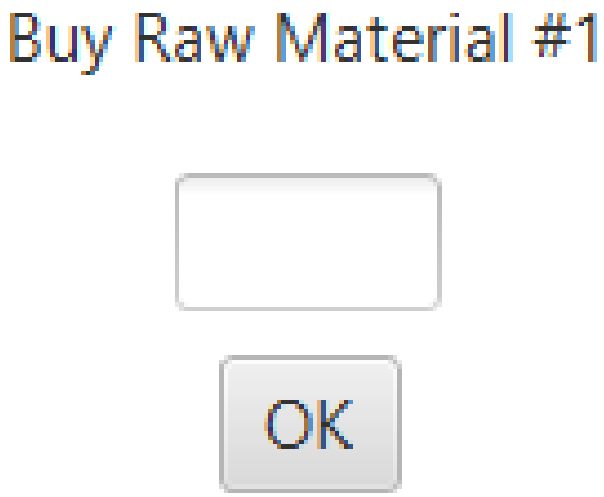
Czasy przezbrojeń są największe na stacjach C oraz D oraz największy kłopot stanowi maszyna niebieska i czasy jej przezbrojeń, ponieważ musi ona pracować na obu marszrutach, co ogranicza produkcję równoległą. Jeżeli chodzi o czas produkcji stacji A jest dużo większy niż każdej innej stacji poza E. To również potęguje problem maszyny niebieskiej, ponieważ zarówno musi pracować na dwóch marszrutach jak i czas produkcji jest bardzo długi. Wąskie gardło jest elementem w procesie produkcji, które ogranicza potencjał procesu produkcyjnego. Tutaj jest to oczywiście maszyna niebieska jak pokazują powyższe argumenty. Problemem w tym procesie jest właśnie zarządzanie niebieską maszyną, a sposobem poradzenia sobie z tym problemem jest planowanie produkcji.

5.3. Wygląd aplikacji

Wygląd podstawowego okna aplikacji został zainspirowany grą OPT Challenge, która została przetestowana przez studentów w ramach laboratorium z przedmiotu Zautomatyzowane Systemy Wytwarzania. Wprowadzono jednak kilka poprawek w porównaniu do pierwotnej wersji. Poza oczywistą zmianą szaty graficznej na bardziej nowoczesną, wprowadzone zostały również informacje odnośnie czasu przezbrajania maszyny oraz czasu produkcji poszczególnego materiału.

Jak można zobaczyć na rysunku 5.1 gra oparta jest na zakupie materiałów, które później zostaną przekształcone na pięciu stacjach, przez 3 maszyny. Aplikacja wyposażona jest w przyciski “Send” oraz “Sell”, wysyłające gotowe materiały do następnej maszyny oraz sprzedające półprodukty i gotowe produkty.

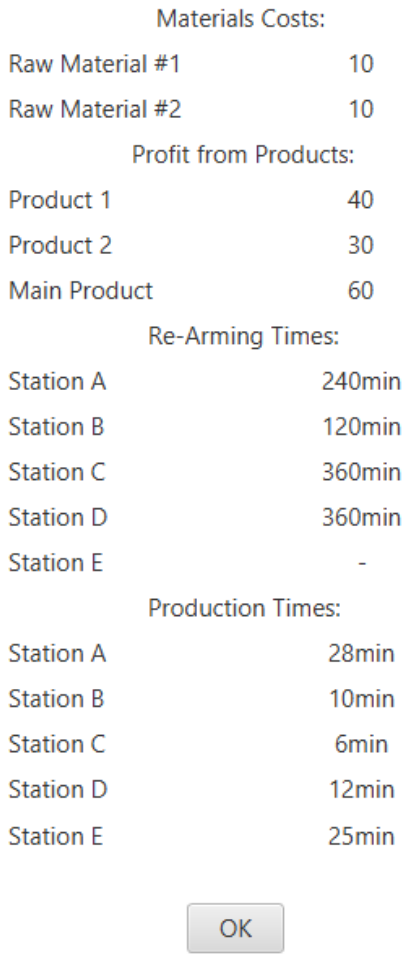
Po kliknięciu przycisków “Buy Raw Material” pojawia się następujące okno:



Rysunek 5.2 Okno “Buy Raw Material”

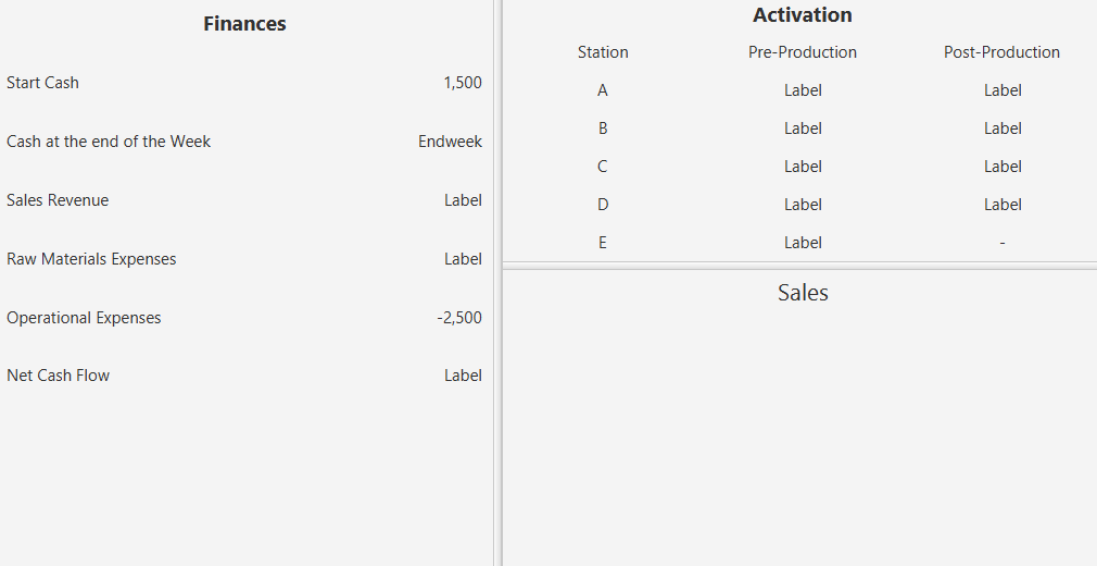
Okno jest zabezpieczone przed wpisywaniem wartości innych niż liczbowe.

W razie wątpliwości odnośnie konkretnych statystyk poszczególnych stacji lub ceny sprzedaży lub zakupu materiałów i produktów w głównym menu znajduje się przycisk “Statistics”. Wygląda on w ten sposób:



Rysunek 5.3 Okno statystyk

Po każdym tygodniu wyświetlało się będzie okno statystyk tygodniowych:



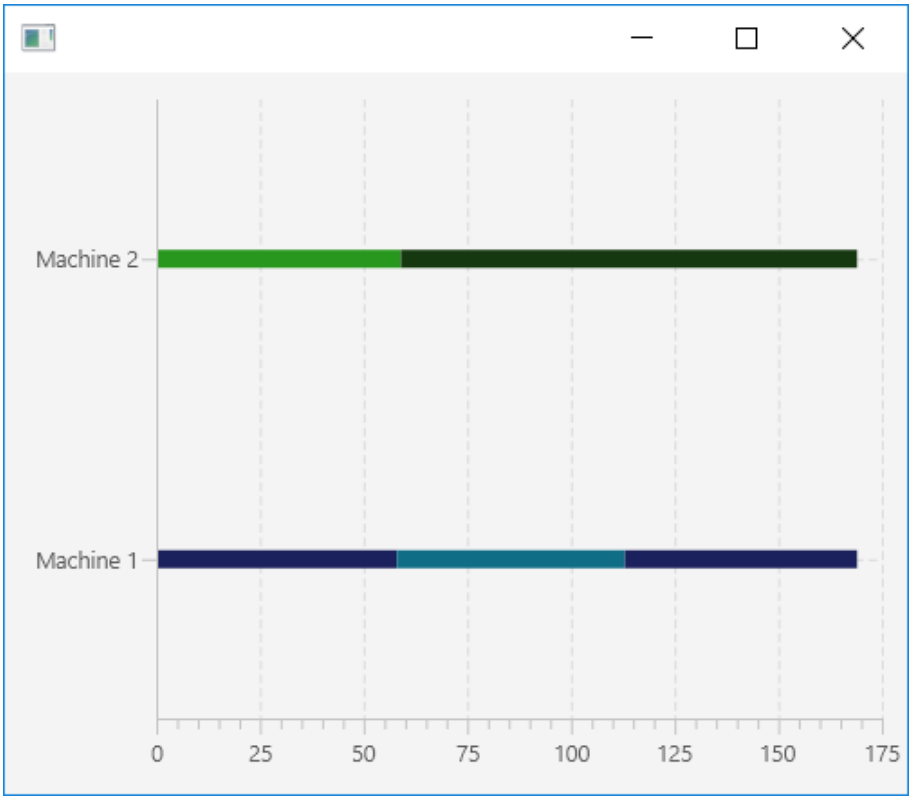
Rysunek 5.3 Okno Statystyk tygodniowych

Poszczególne liczby z części “Finances” liczone są w następujący sposób:

- “Start Cash” - Początkowa ilość pieniędzy na początku gry
- “Cash at the end of the Week” - Ilość pieniędzy po zakończonym tygodniu
- “Sales Revenue” - Zysk ze sprzedaży produktów
- “Raw Materials Expenses” - Ilość gotówki przeznaczonej na zakup materiałów
- “Operational Expenses”- Tygodniowy koszt operacyjny(np. zapłata dla pracowników, koszty elektryczności itp.)
- “Net Cash Flow” - Jest to różnica pomiędzy zyskami ze sprzedaży, a wszystkimi wydatkami

Część “Activation” zawiera dane odnośnie ilości materiałów aktualnie znajdujących się na poszczególnych stacjach. Ostatnia część zawiera diagram kołowy pokazujący sprzedaż gotowych produktów oraz półproduktów.

Po drugim tygodniu rozgrywki program wyświetla również wykresy Gantta. Kolory na wykresie odpowiadają kolorom poszczególnych stacji z okna głównego.



Rysunek 5.4 Wykres Gantta

5.4. Aplikacja ze strony programistycznej

Działanie zarówno czasu przezbrajania, czasu przetwarzania materiałów, jak i ogólnego czasu gry oparte jest na tzw. Timerach. W aplikacji wybrano bibliotekę ReactFX, ze względu na krótszą implementację oraz ważną wadę Timerów z podstawowych bibliotek JavaFX. Tą wadą jest często występująca sytuacja, w której nawet po anulowaniu Timera, akcja zostanie wykonana jeszcze raz, po raz ostatni. W grze powodowało to między innymi: przetworzenie dodatkowego materiału już po kliknięciu innej stacji w celu przezbrojenia, spowodowanie błędu całej aplikacji, gdyż tablica zawierająca poszczególne dni tygodnia wychodziła poza zakres. Biblioteka ReactFX zapewniła brak tego błędu.

```
Timeline timeline = new Timeline(new KeyFrame(
        Duration.millis(2500),
        ae -> doSomething()));
timeline.setCycleCount(Animation.INDEFINITE);
timeline.play();
```

Jsing `FxTimer`, it looks like this:

```
FxTimer.runPeriodically(
        Duration.ofMillis(2500),
        () -> doSomething());
```

Rysunek 5.4 Porównanie długości implementacji Timera działającego cyklicznie w standardowej bibliotece oraz w ReactFX

Całość aplikacji została napisana we wzorcu projektowym MVC (Model-View-Controller), który jest bardzo popularnym wzorcem, gdy programuje się aplikacje graficzne. Polega on na podziale aplikacji na 3 główne części:

- Model - W tej części zaprogramowana jest ścisła funkcjonalność programu
- View - część programu odpowiedzialna za szatę graficzną
- Controller – tutaj przyjmujemy dane wejściowe od użytkownika i wywołujemy odpowiednie funkcje z części model

Wygląd szaty graficznej został zaprojektowany we wspomnianym w części teorytycznej Scene Builderze oraz wzbogacony o efekty wizualne z języka CSS. Część modelowa zawiera czyste programowanie w języku Java i właśnie tam znajduje się kod odpowiedzialny za np. przetwarzanie danych po kliknięciu przycisku maszyny lub rysowanie wykresów Gantta. Controller reaguje po prostu na kliknięcia poszczególnych elementów okna i wywołuje funkcje części modelowej. Ta część jest również odpowiedzialna za zmiany wywoływane w konkretnym oknie np. zmiany w ilości materiałów gotowych do przetworzenia jak i tych po przetworzeniu. W JavieFX dzięki wprowadzeniu nowego typu zmiennych “SimpleIntegerProperty” możliwa jest dynamiczna zmiana wartości bez użycia Timerów. Java jest językiem obiektowym, więc

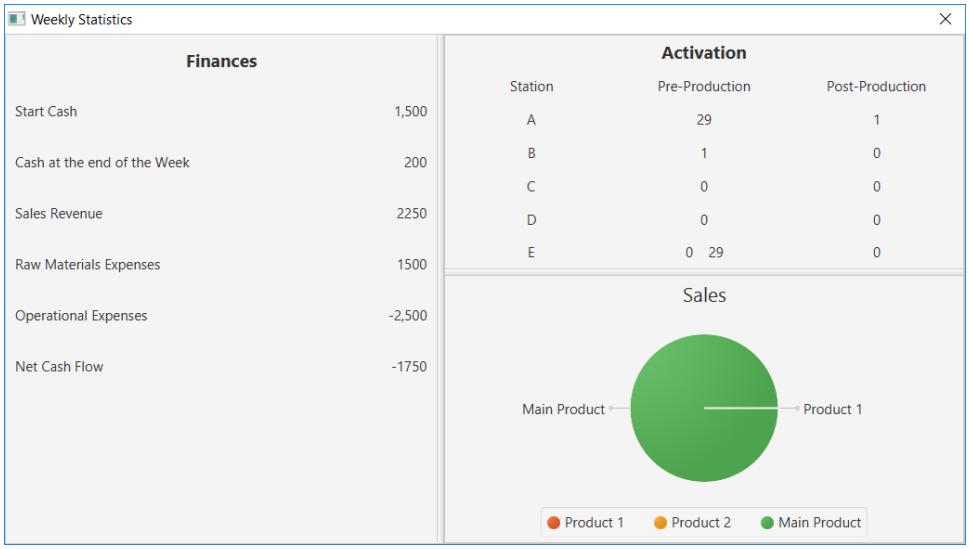
oczywiście to klasy są głównym szkieletem programu. Jak najlepsze użycie klas, gwarantuje zwięzłość kodu, w programie inżynierskim jednak, w części graficznej programu, znajduje się bardzo duża ilość elementów zmieniających wartość, jak np. TextField, w którym pokazywane są ilości materiałów na poszczególnych stacjach. Te elementy niestety trzeba traktować osobno i dla każdego takiego elementu napisać osobny kod. Scene Builder zapewnia ogromną oszczędność kodu, poprzez zastosowanie plików fxml i systemu Drag & Drop. W porównaniu do poprzedniego, najbardziej popularnego Frameworka Java Swing, można założyć, że dzięki ogromnej liczbie elementów w aplikacji, kod napisany fizycznie przez użytkownika został zmniejszony o nawet 300%!

6.Testy aplikacji

W pokazanym w aplikacji procesie produkcyjnym istnieje wiele strategii, które zapewnią przedsiębiorstwu zysk. „Wąskie gardło" jest jednak ograniczeniem, które wymaga planowania i w wypadku przedstawionego procesu produkcyjnego jest sporym problemem. Poniżej przedstawiono 3 strategie, dwie pokazujące błędne planowanie oraz jedną zapewniającą zysk przedsiębiorstwu.

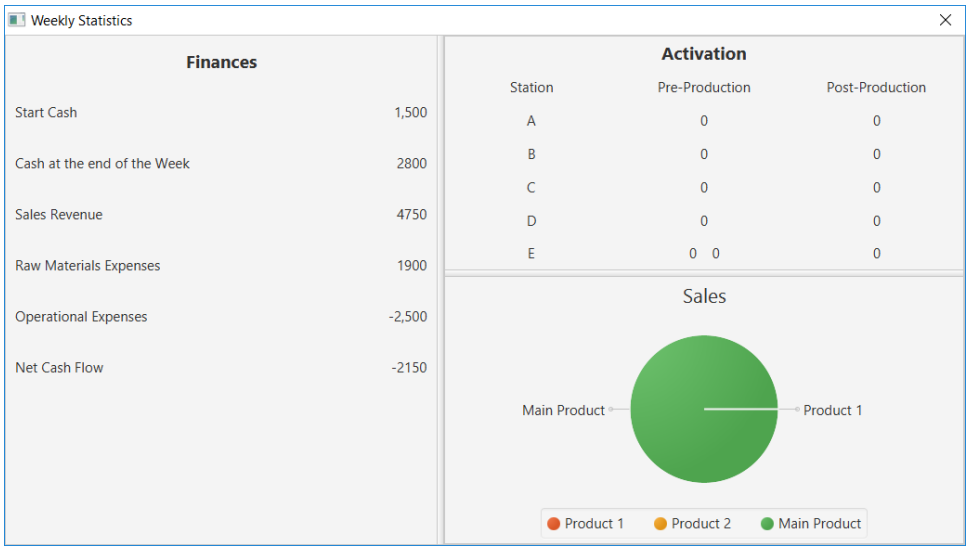
Strategia nr1:

Pierwsza strategia zakłada sprzedawanie tylko głównych produktów i jest swego rodzaju prezentacją, jakiego rzędu zyski można osiągnąć skupiając się tylko na sprzedaży głównego produktu. Przed rozpoczęciem rozgrywki zakupujemy po 75 surowców na każdej z marszrut i uruchamiamy pierwsze stacje. Zaczynamy rozgrywkę. Po wytworzeniu 25 gotowych półproduktów na stacji A, uruchamiamy stację C i D oraz wysyłamy półprodukty na stację E. Po przetworzeniu materiałów na stacji D od razu wysyłamy półprodukty na stację E, by nie hamować produkcji. Gdy stacja C przetworzy wszystkie materiały, ponownie uruchamiamy stację A.



Rysunek 6.1. Statystyki po pierwszym tygodniu gry

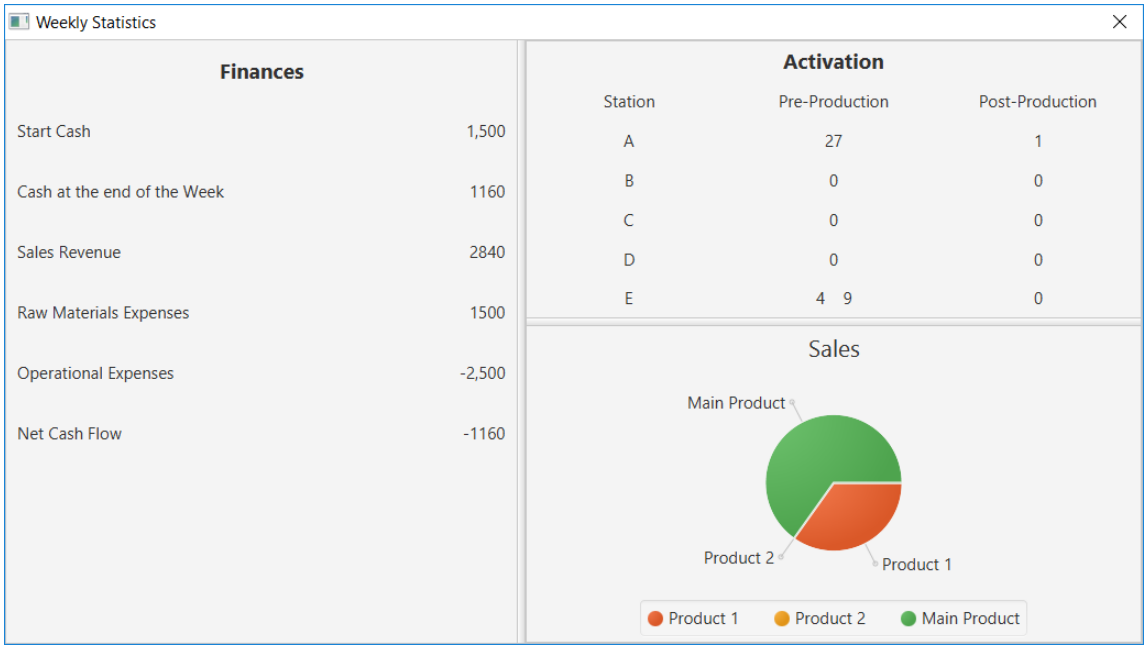
W drugim tygodniu dokupujemy surowce do obu stacji i postępujemy identycznie jak w pierwszym tygodniu. Jak można zauważyć, zysk z samych produktów głównych jest rzędu 5000, co nie jest wystarczające. Sprzedawanie półproduktów jest potrzebne by osiągnąć zysk.



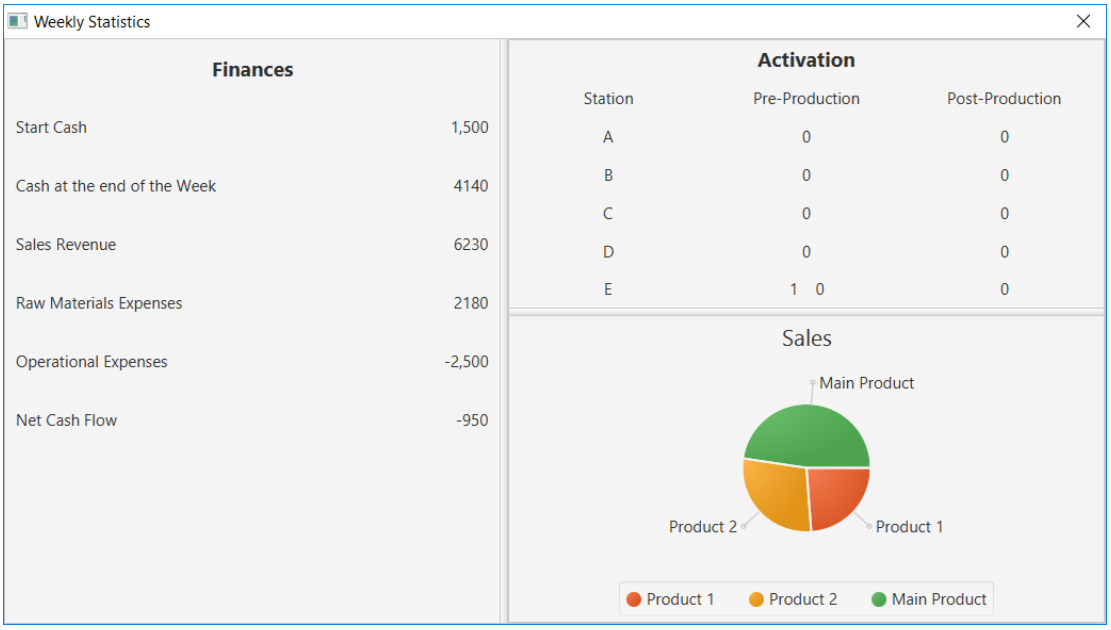
Rysunek 6.2. Statystyki po drugim tygodniu gry

Strategia nr2:

Ta strategia opera się na skupieniu się najpierw na głównym produkcie, by potem zacząć sprzedawać półprodukty z obu marszrut w podobnych ilościach. Rozpoczynamy identycznie jak w poprzedniej strategii. Kupujemy po 75 jednostek surowca z obu marszrut. Uruchamiamy stacje i rozpoczynamy rozgrywkę. Po utworzeniu 25 sztuk półproduktu w stacji A, uruchamiamy stacje C i D oraz wysyłamy półprodukt do stacji E. Po przetworzeniu przez stację D półprodukt wysyłamy na stację E i rozpoczynamy produkcję głównego produktu. Gdy przetwarzanie zakończy się na stacji C ponownie włączamy stację A, a przetworzone w niej materiały wysyłamy na stację E. W ostatni dzień tygodnia sprzedajemy pozostałe półprodukty z marszruty BCD. Gdy ilość sprzedanych głównych produktów osiągnie 50, skupiamy się na sprzedaży półproduktów z marszruty A oraz marszruty BCD, w ramach potrzeby zakupując potrzebne surowce.



Rysunek 6.3. Statystyki po pierwszym tygodniu gry



Rysunek 6.4. Statystyki po drugim tygodniu gry

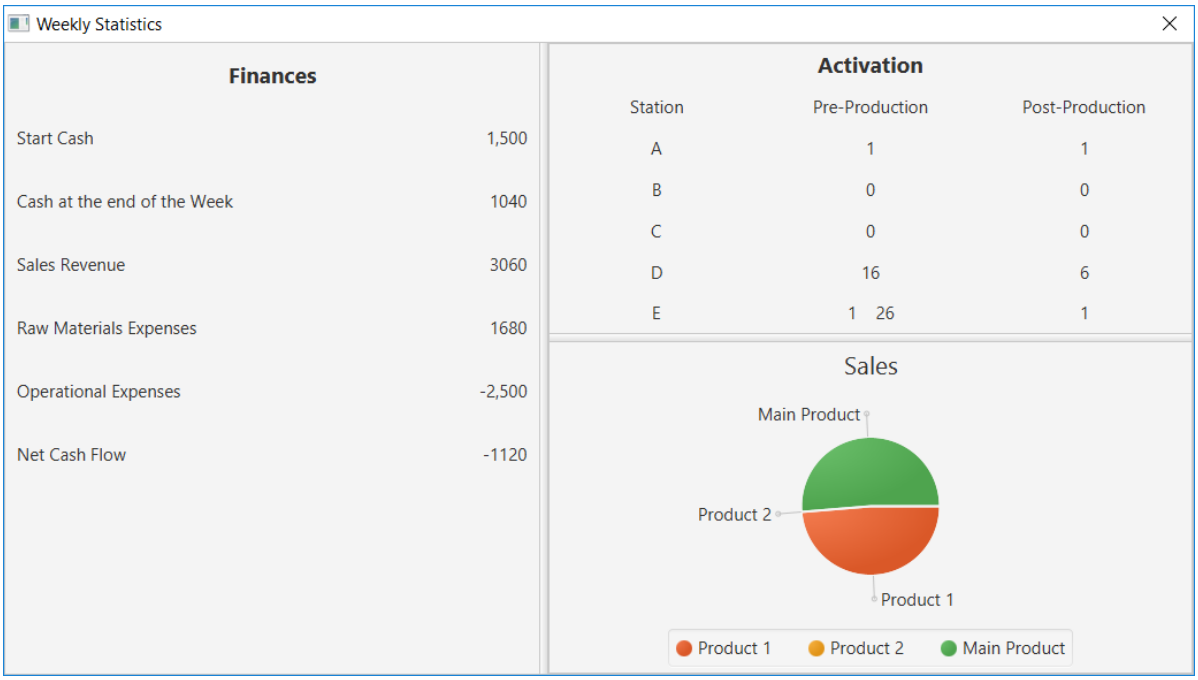
Strategia ta również nie gwarantuje nam zysku, jednak jest ona dużo bardziej efektywna niż pierwsza strategia. Problemem okazał się czas produkcji półproduktu na stacji A.

Strategia nr.3 (wygrywająca)

Druga strategia pokazała, że sprzedaż półproduktów generuje większe zyski. Ukazała ona również problem, który stanowi marszruta A, dlatego w tej strategii skupiono się na sprzedaży produktu głównego oraz półproduktu z marszruty BCD. Półprodukt z marszruty A był sprzedawany w znacznie mniejszych ilościach.

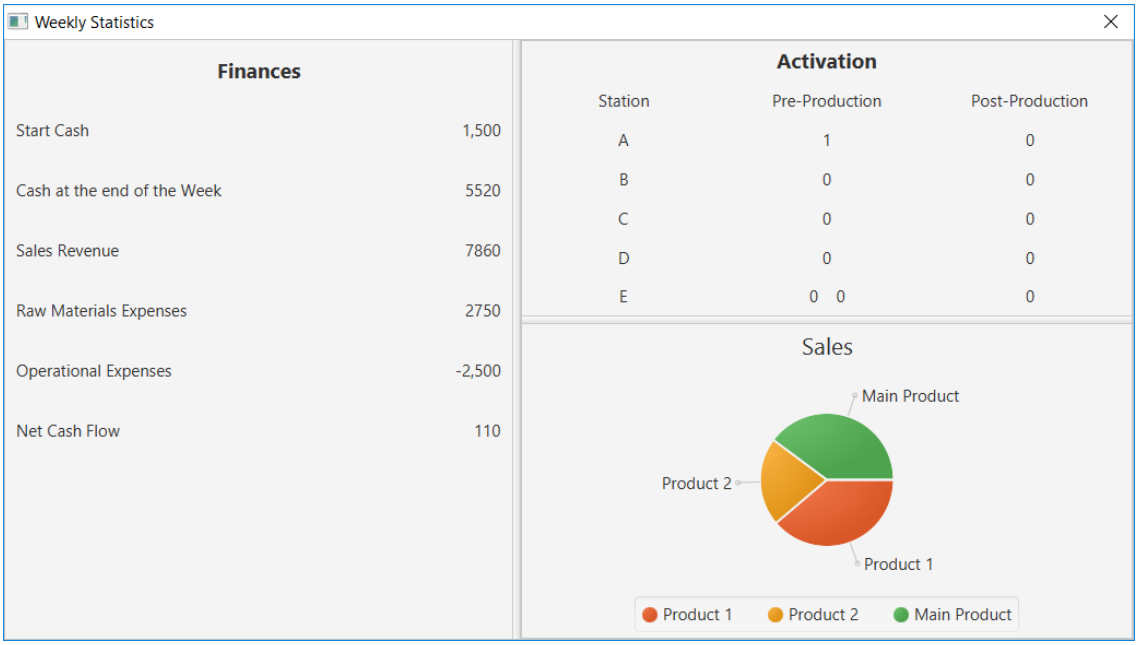
Nim rozpoczniemy grę kupujemy potrzebne surowce, wykorzystujemy cały budżet kupując 25 sztuk surowca 1 oraz 125 surowca 2, później uruchamiamy maszyny A, B i E. Zaczynamy proces. Po wytworzeniu 25 sztuk ze stacji A uruchamiamy maszynę na stacji C stale wysyłając produkty. Gdy maszyna na stacji A przetworzy wszystko, przezbrajamy maszynę na stacji D. Wszystko oczywiście wysyłamy na kolejne etapy. Gdy stacja D zacznie wydawać gotowe produkty przekazujemy je na stację E by nie

było przestojów. Jako, że maszyna niebieska ze stacji A produkuje mało produktów musimy również pamiętać o sprzedawaniu odpowiednich ilości produktu z marszruty BCD. Gdy stacja C przetworzy wszystkie produkty włączamy stację A i dostarczamy produkty od razu na stacje E.



Rysunek 6.5. Statystyki po pierwszym tygodniu gry

W drugim tygodniu tworzymy produkt na marszrucie A i wysyłamy go od razu do stacji E, w razie potrzeby dokupujemy surowiec. Gdy będziemy w stanie wyprodukować 65 sztuk głównego towaru sprzedajemy pozostałe produkty z marszruty BCD i A, aż do końca tygodnia.



Rysunek 6.6. Statystyki po drugim tygodniu gry

7. Podsumowanie

Celem pracy inżynierskiej było stworzenie aplikacji mającej na celu zapoznanie studentów z problemem planowania produkcji. Aplikacja ma formę gry, ze względu na łatwość w odbiorze. Projekt został napisany w języku JAVA, przy użyciu Frameworka JavaFX oraz narzędziu do tworzenia statycznych interfejsów: Scene Builder. Pozwoliło to na zaimplementowanie części logicznej w klasach języka Java, a części interfejsowej w języku XML w plikach fxml. Skomplikowanie programu było wysokie zatem finalnie kod liczył ponad 2000 linii kodu. Przedstawiony proces produkcyjny jest oczywiście przykładowy i w przyszłości może być z łatwością modyfikowany i rozbudowywany(np. o liczbę maszyn czy stacji), ze względu na uniwersalność kodu. Tworząc kod zastosowano wzorzec projektowy MVC, daje to łatwość w interpretacji kodu, dzięki czemu zwiększa się potencjalne grono osób rozbudowujących aplikację. Sytuacja przedstawiona w aplikacji przybliża problem „wąskich gardeł", który w mniejszym lub większym stopniu istnieje w każdym przedsiębiorstwie. Gra pokazuje (oczywiście w ograniczony sposób) proces produkcyjny i przybliża sposoby radzenia sobie z rzeczonymi wąskimi gardłami (w tym przypadku jest to poprawne planowanie).

8. Literatura

[1] https://javastart.pl/baza-wiedzy/darmowy-kurs-java/wprowadzenie/historia-javy

[2] https://stormit.pl/historia-java/

[3] https://www.java.com/pl/download/faq/javafx.xml

[4] https://pl.wikipedia.org/wiki/Eclipse

[5] Józef Gawlik, Jarosław Plichta, Antoni Świć "Procesy produkcyjne", 2013

[6] Ireneusz Durlik "Inżynieria zarządzania część 1", 2018

[7] Jerzy Lewandowski, Bożena Skołud, Dariusz Plinta "Organizacja systemów produkcyjnych", 2014

[8] http://kkapd.f11.com.pl/zsw/algorytm_johnsona.htm

[9] Eliyahu M. Goldratt "Cel I: Doskonałość w produkcji", 2008

[10] https://bezpieczenstwowsystemachsterowania.pl/2013/06/co-to-jest-maszyna/

[11] https://lean.org.pl/smed-czyli-skracanie-czasow-przezbrojen-maszyn-i-urzadzen/