

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2020

(Випуск 2)

[електронне видання]



<http://nure.ua/department/kafedra-komp-yuterno-integrovanih-tehnologiy-avtomatizatsiyi-ta-mehatroniki-kitam>



<http://itez.zntu.edu.ua/>



<http://kafea.kdu.edu.ua>

Харків 2020

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки
кафедра комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки
(КІТАМ)



ЗБІРНИК

студентських наукових статей

«Автоматизація та приладобудування»

«Automation and Development of Electronic Devices»

ADED-2020

(Випуск 2)

[електронне видання]

Харків 2020

АВТОМАТИЗАЦІЯ ТА ПРИЛАДОБУДУВАННЯ («Automation and Development of Electronic Devices» ADED-2020) [Електронний ресурс] : збірник студентських наукових статей / Харківський національний університет радіоелектроніки ; [редкол.: І.Ш. Невлюдов та ін.]. – Харків : ХНУРЕ, 2020. – Вип. 2. – 298 с.

COLLECTION OF STUDENTS' SCIENTIFIC PAPER «AUTOMATION AND DEVELOPMENT OF ELECTRONIC DEVICES» ADED-2020 Part 2 (Key infrastructure 2020) - Kharkiv/ The Editorial.: Nevlyudov I.Sh. (head), that all. Kharkiv: Kind of Kharkiv National University of Radio Electronics [electronic edition], 2020.- 298 p with.

Рекомендовано рішенням
Науково-технічної ради
Харківського національного
університету радіоелектроніки
протокол №6 від 29.11.2018

Рекомендовано рішенням Вченої ради
факультету Автоматики і комп'ютеризованих
технологій
Харківського національного
університету радіоелектроніки
протокол № 2 від 23.11.2020

Збірник містить наукові статті студентів кафедри комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки (КІТАМ) Харківського національного університету радіоелектроніки, кафедри Інформаційних технологій електронних засобів (ІТЕД) Запорізького національного технічного університету та кафедри Електронних апаратів (ЕА) Кременчуцького національного університету ім. М. Остроградського які навчаються за спеціальностями: 151 Автоматизація та комп'ютерно-інтегровані технології, 172 Телекомунікації та радіотехніка, 171 Електроніка та 163 Біомедична інженерія, першого (бакалаврського), другого (магістерського) рівнів вищої освіти. Статті надані в авторській редакції.

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ТА ТЕСТУВАННЯ ПАРАМЕТРІВ ВИРОБНИЧОЇ ЛІНІЇ

Бондаренко Ю. В., Гіль А. А., Валківська Є. Ю.

Харківський національний університет радіоелектроніки

Україна, 61166, Харків, пр. Науки, 14

E-mail: yurii.bondarenko@nure.ua, anastasiia.lezhenina@nure.ua, danylo.iurkov@nure.ua

Анотація: В даній роботі приведено аналіз програмного забезпечення для моделювання та тестування параметрів виробничої лінії, розглянуто основний алгоритм роботи даного продукту, описано перелік функцій які використовувались для розробки даного продукту та код Unit тесту.

Ключові слова: Програмне забезпечення, Unity, моделювання, тестування, виробничі лінії.

ANALYSIS OF SOFTWARE FOR MODELING AND TESTING PRODUCT LINE PARAMETERS

Y. Bondarenko, A. Gil, Y. Valkivska

Kharkiv National University of Radio Electronics

Ukraine, 61166, Kharkiv, Nauky av., 14

E-mail: yurii.bondarenko@nure.ua, anastasiia.lezhenina@nure.ua, danylo.iurkov@nure.ua

Annotations: This paper presents an analysis of software for modeling and testing the parameters of the production line, considers the basic algorithm of this product, describes the list of functions used to develop this product and the Unit test code.

Key words: Software, Unity, modeling, testing, production lines.

Програмний продукт який аналізується розроблено на платформі Unity за допомогою вбудованого конструктора UI елементів.

Платформа Unity має наступні переваги та недоліки:

Переваги:

- можливість програмувати для 2D та 3D проектів;
- потужна графічна підтримка;
- можливість працювати з звуком;
- простий інтерфейс та легкий в освоєнні;
- можливість переносити на будь-яку платформу програму.

Недоліки:

- заплутана система масштабування програми;
- великі проекти можуть мати занадто багато сцен що може призвести до непорозуміння власного проекту.

Програмне забезпечення працює за наступним алгоритмом роботи (рис. 1).

Даний алгоритм описує взаємодію компонентів а саме початком являється верхній блок з надписом початок, після чого йде блок вхідних даних «вибір функції» після вибору функції користувач має вибір почати симуляцію чи почати розрахунки.

При виборі почати симуляцію користувач вводить необхідні дані та отримує вихідні данні, які містять значення загальної похибки виробництва деталей, похибку кожного агрегату данної лінії, кількість браку виготовленого по кожному агрегату та точність агрегатів.

При виборі Початок розрахунків користувачу буде доступно 5 кнопок 4 із яких будуть відповідати за певні розрахунки після вибору однієї з 4 кнопок користувач повинен може ввести дані та отримати результат розрахунків, якщо користувач не хоче проводити розрахунки то він може повернутися в попереднє меню після чого в головне якщо йому не потрібні розрахунки і вийти з програми.

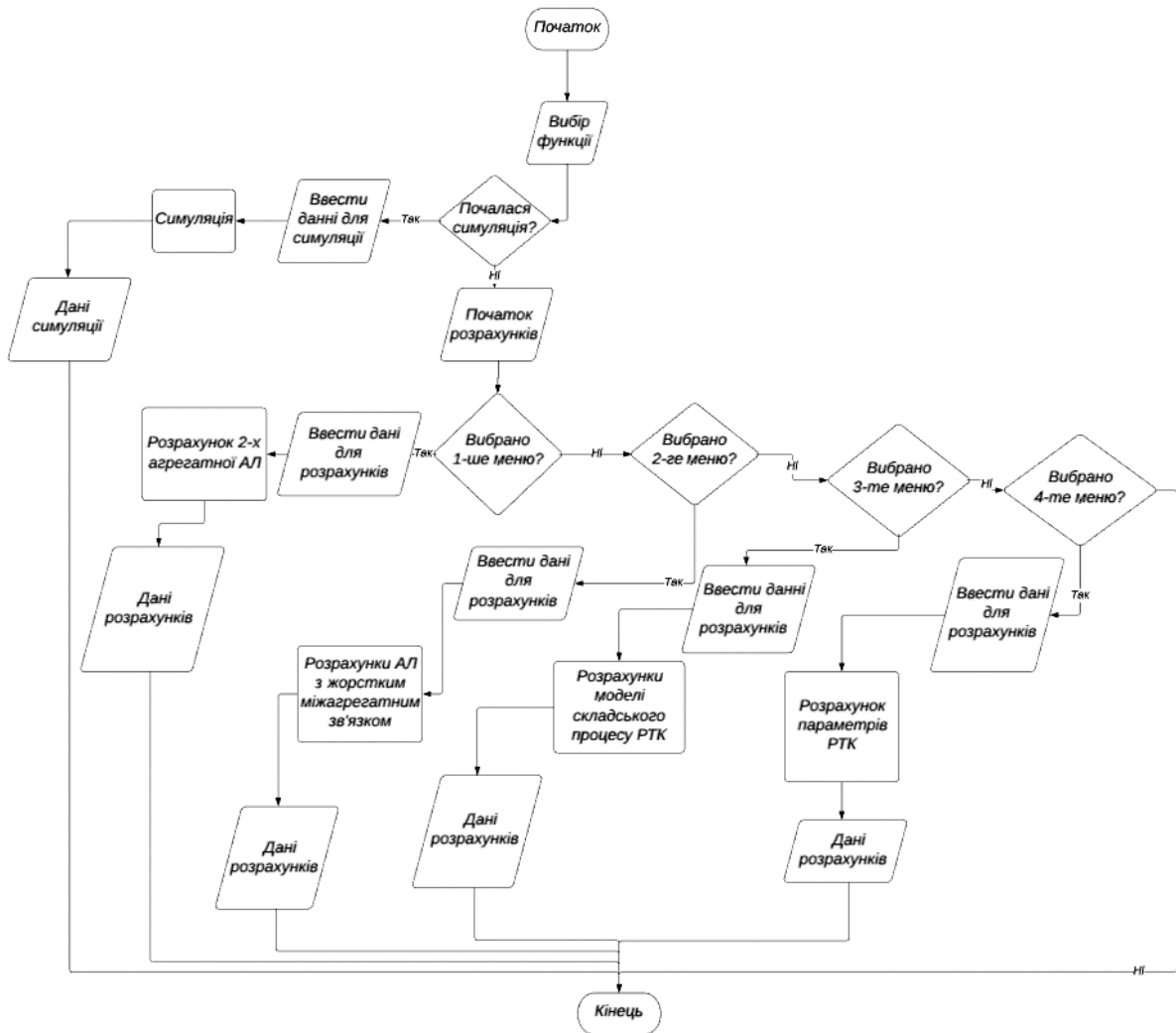


Рисунок 1 – Алгоритм роботи програмного засобу

Функції розрахунку містять 4 основні напрями:

- параметри двох агрегатних ліній;
- параметри автоматизованих ліній з жорстким міжагрегатним зв'язком;
- параметри модуля складського процесу РТК (робото-технічний комплекс);
- параметри РТК.

Параметри двох агрегатної лінії можуть визначатися критерієм відмови автоматизованої лінії (АЛ). За параметрами продуктивності може бути:

- припинення функціонування на час, що перевищує припустимий;
- зниження продуктивності чи ритму випуску продукції нижче заданого рівня;
- невиконання завдання за обсягом випуску продукції.

Для розробки ПЗ було підключено наступні бібліотеки функцій який використовувалися:

- System.Collections;
- System.Collections.Generic;
- UnityEngine;
- UnityEngine.SceneManagement;
- UnityEngine.UI.

Також для розробки скрипти використовувалися наступні функції:

- sceneManager.LoadScene (назва сцени);
- application.Quit ();

- `button.onClick.AddListener` (дія);
- `panel.SetActive` (`false/true`);
- `Convert.ToDouble` (змінна).

Функція `SceneManager.LoadScene` (назва сцени) завантажує сцену (наступну сторінку), ця функція використовується для кнопок або інших об'єктів як без умовний перехід, якщо використовується в порі з кнопкою то виконується по натисканні кнопки.

Функція `Application.Quit()` використовується для закриття програми майже завжди використовується з кнопкою.

Функція `button.onClick.AddListener`(дія) в даній програмі використовується з кнопкою та полем вводу(`InputField`) в цю функцію як параметр передається певна послідовність дій які виконуються по натисканню кнопки.

Функція `panel.SetActive(false/true)` використовується для зникнення та появи певної панелі або певного поля на робочій поверхні, де значення `false` відповідає за вимкнений стан панелі, а значення `true` за увімкнений.

Функція `Convert.ToDouble`(змінна) має логічне навантаження тобто вона використовується в коді для перетворень змінний типу `text`(текст) на змінні типу `double`(дробові числа) для подальших обчислень.

Unit тест представляє собою фрагмент коду, який перевіряє роботоспроможність алгоритму або частини алгоритму окремо від інших частин коду. Unit тест проводять частіше всього в IDE, які мають поля для виводу тексту такі як консоль.

Для перевірки роботоспроможності програми беруть фрагмент алгоритму(коду), який необхідно протестувати та передають в цей модуль параметри необхідні для роботи його функціонування.

В нашому випадку тестування представляє собою метод `test_anim` в якому тестується роботоспроможність функції симулятора яка симулює певний відхід(брак) виготовлених плат. Тест відбувався за допомогою коду приведенного нижче.

```
public void test_anim()
{
    int amount = 1000000;
    int brak1 = 0;
    int brak2 = 0;
    int brak3 = 0;
    int brak4 = 0;
    double[] mass1 = new double[amount];
    double[] mass2 = new double[amount];
    double[] mass3 = new double[amount];
    double[] mass4 = new double[amount];
    for (int i = 0; i < amount; i++)
    {
        mass1[i] = RandomDoubleValue(-0.1505, 0.1507);
        if (mass1[i] > 0.15 || mass1[i] < -0.15) brak1++;
        mass2[i] = RandomDoubleValue(-0.1009, 0.1009);
        if (mass2[i] > 0.10 || mass2[i] < -0.10) brak2++;
        mass3[i] = RandomDoubleValue(-0.2509, 0.2509);
        if (mass3[i] > 0.25 || mass3[i] < -0.25) brak3++;
        mass4[i] = RandomDoubleValue(-0.4009, 0.4009);
        if (mass4[i] > 0.40 || mass4[i] < -0.40) brak4++;
    }
    double midle =(double) (brak1 + brak2 + brak3 + brak4) / amount;
    Console.WriteLine("brak1={0}", brak1);
    Console.WriteLine("brak2={0}", brak2);
}
```

```

Console.WriteLine("brak3={0}", brak3);
Console.WriteLine("brak4={0}", brak4);
Console.WriteLine("brak v %={0:0.#####}\n", midle);
double time = 0;
time = 0.00045 * amount;
Console.WriteLine("час на виготовлення продукту = {0}, годин", time);
}
public static double RandomDoubleValue(double dMin, double dMax)
{
    System.Random random = new System.Random();
    return random.NextDouble() * (dMax - dMin) + dMin;
}.

```

Результат тесту показав, що модуль симуляції є повністю працездатним та може генерувати значення в певному контексті значень.

ВИСНОВКИ: таким чином можна сказати, що даний продукт є повністю працездатним, функції використані при розробці продукту є найбільш підходящими для даного типу продукту, алгоритм є доволі простим, тому програма легка в користуванні.

ЛІТЕРАТУРА

1. Хокинг Д. Unity в действии. Мультиплатформенная разработка на C# / Д. Хокинг., 2018. – 352 с. – (Питер).
2. Невлюдов І. Ш. Трансфер технологій у сучасній науці, освіті та виробництві в умовах четвертої промислової революції «ІНДУСТРІЯ 4.0» / І. Ш.Невлюдов, О. О. Чала, Ю. М. Олександров // Сучасний рух науки: тези доп. VIII міжнародної науково-практичної інтернет-конференції, 3-4 жовтня 2019 р.– Дніпро, 2019. – Т.2 С.: 604-608
3. Виробничі процеси та обладнання об'єктів автоматизації. Збірник задач : навч. посіб. / І. Ш. Невлюдов, А. О. Андрусевич, Г. В. Пономарьова, А. О. Функендорф ; М-во освіти і науки України, Харків. нац. ун-т радіоелектроніки. – Кривий Ріг : КК НАУ, 2018. – 332 с.
4. Невлюдов І. Ш. Комп'ютерні технології автоматизованого виробництва: [навч. посібник] / І. Ш. Невлюдов, М. А. Бережна. – Харків : СМІТ, 2007. – 368 с. – ISBN 978-966-8530-99-9. – 41,00
5. I. Nevliudov "Using MEMS to adapt ultrasonic welding processes control in the implementation of modular robots assembly processes" / S. Maksymova, A. Funkendorf, O. Chala and K. Khrustalev // IEEE XIV-th International Conference Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2018 pp. 223-226.
6. Borojevic S. Modeling, simulation and optimization of process planning, Journal of Production Engineering / S. Borojevic, V. Jovisevic, S. Jokanovic – Serbia, 2009. – С. 87–90.
7. Siderska J. Application of tecnomAtix plAnt SimulAtion for modeling production And logiSticS proceSSeS / Julia Siderska // BUSINESS, MANAGEMENT AND EDUCATION / Julia Siderska., 2016. – (ISSN). – С. 64– 73
8. Danilczuk W. Analiza konfiguracji linii produkcyjnych na podstawie modeli symulacyjnych / W. Danilczuk, R. Cechowicz,, A. Gola – lubelsk, 2014. – С. 25–42.
9. Gola A. Innovations in management and production engineering / A. 8. Gola, A. Świc // Directions of Manufacturing systems' evolution from the lexibility level point of view / A. 8. Gola, A. Świc., 2012. – С. 226–238.
10. Devnozashvili M., Selivanova K. G. Medication reminder device development:дис. – ХНУРЕ, 2019.
11. Plant Simulation [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.plm.automation.siemens.com/store/ru-ru/trial/plant-simulation.html>.

Науковий керівник: Чала Олена Олександрівна, старший викладач кафедри КІТАМ Харківського національного університету радіоелектроніки