

УДК 004.4`2

Коцюба А.Ю., Лавренчук С.В.

Луцький національний технічний університет

## ПРО ПРОГРАМУ-КОНСТРУКТОР WEB-ІГОР РІЗНОГО РІВНЯ СКЛАДНОСТІ

**Коцюба А.Ю., Лавренчук С.В. Про програму-конструктор web-ігор різного рівня складності.** Засобами C++ Builder та за допомогою поєднання технологій подійно- та об'єктно-орієнтованого програмування розроблено програму, яка створює web-ігри різного рівня складності. Розроблений програмний продукт дає можливість користувачу на основі побудованої схеми доріг та вибору налаштувань створювати відповідну динамічну web-сторінку.

**Ключові слова:** двовимірний клітинний автомат, конструювання web-ігор, моделювання правостороннього руху транспорту, технологія автоматного програмування з явним виділенням станів, алгоритм Дейкстри.

**Коцюба А.Ю., Лавренчук С.В. О программе-конструкторе web-игр разного уровня сложности.** Средствами C++ Builder и с помощью сочетания технологий событийно- и объектно-ориентированного программирования разработана программа, которая создает web-игры различного уровня сложности. Разработанный программный продукт позволяет пользователю на основе построенной схемы дорог и выбора настроек создавать соответствующую динамическую web-страницу.

**Ключевые слова:** двумерный клеточный автомат, конструирование web-игр, моделирование правостороннего движения транспорта, технология автоматного программирования с явным выделением состояний, алгоритм Дейкстры.

**Kotsyuba A.Yu, Lavrenchuk S.V. About the designer-web-games of various levels of complexity.** Means of C++ Builder and using a combination of event-technology and object-oriented programming, a program that creates web-games of various levels of complexity. The software product allows the user on the basis of the scheme constructed roads, and select the settings to create the corresponding dynamic web-page.

**Keywords:** two-dimensional cellular automaton, construction web-games, modeling right-movement of transport, technology of automats programming of automats with evident allocation of state, Dijkstra's algorithm.

**Постановка наукової проблеми та її значення.** Не потрібно розповідати як важливо навчитися писати програми, які самі були б здатні створювати програмний код. Дослідження в цьому напрямку триває з часів, коли ще тільки почали створюватися технології програмування. А вирішення цієї проблеми зводиться до формалізації поняття алгоритму. Однією з методик (чи технологій програмування), яка дозволяє це зробити, є, так званий, стиль автоматного програмування з явним виділенням станів. Суть цієї технології полягає у тому, що кожному алгоритму у відповідність можна поставити, наприклад, двовимірний клітинний автомат [1,2], який можна побудувати на основі створеної користувачем схеми доріг. За допомогою такого підходу можна вирішувати цілу низку проблем, пов'язаних, наприклад, з моделюванням правостороннього дорожньо-транспортного руху у динамічній web-сторінці, що дозволяє будувати web-ігри, які б могли б зацікавити найвибагливіших геймерів. Для цього необхідно ще й реалізувати цілий ряд додаткових можливостей (зокрема, створити функцію за допомогою алгоритму Дейкстри, яка б без застосування складних структур даних, крім довжини найкоротшого шляху, давала ще б й послідовність комірок цього шляху, та інші функції, принцип роботи яких буде описано нижче), розробити дизайн результату тощо. Хоча комп'ютерні ігри і не відносять до серйозного програмного забезпечення, але саме завдяки ним, відбувається стрімкий розвиток комп'ютерних технологій програмування. І найбільш складною задачею у цьому напрямку є проблема створення програм-конструкторів.

Програму-конструктор будемо створювати в середовищі CodeGear RAD Studio C++ Builder 2009. Одержаний результат – web-гру можна буде запускати за допомогою браузерів нових версій Opera та Google Chrome.

**Аналіз останніх досліджень та публікацій.** У працях [3–6] вивчено основні принципи технології автоматного програмування з явним виділенням станів (ще цю технологію називають подійно-орієнтованим програмуванням (ПОП). Детальніше особливості роботи з середовищем C++ Builder описано в роботах [7–9]. А щодо технології написання скриптів мовою JavaScript, то тут достатнім буде використання джерела [10].

**Мета роботи** – поєднання можливостей об'єктно- та подійно-орієнтованого стилів програмування та одержання програми-конструктора, яка б створювала динамічні web-сторінки (web-ігри різного рівня складності).

**Методи розробки.** Було використано технологію ПОП, яка базується на понятті двовимірного клітинного автомата [1,2].

**Виклад основного матеріалу дослідження.** Робота всього програмного продукту побудована на роботі взаємопов'язаних модулів, кожен з яких відповідає за роботу конкретної частини

програмного продукту. Розроблений програмний продукт складається з декількох модулів, кожен з яких виконує певні функції і завдання (рис. 1).

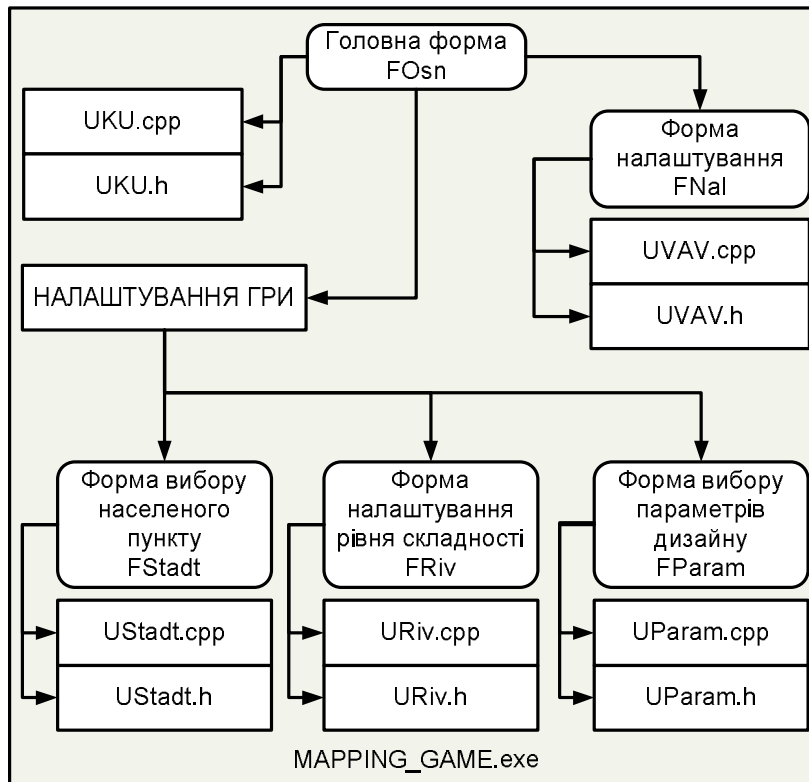


Рис. 1. Схема архітектури програми-конструктора

Програма-конструктор web-ігор різного рівня складності призначена для виконання наступних основних функцій:

- зміни розмірів вертикального та горизонтального зазорів;
- конструювання користувачем довільної схеми доріг;
- виправлення помилок (компіляції) при створенні файлу-результату;
- вибору населеного із уже запрограмованого переліку назви населеного пункту та номеру гри, пов'язаного з ним (останній вибір дає можливість для деякого населеного пункту створювати 999 варіантів ігор);
- налаштування різних параметрів гри, що дозволяє керувати рівнем складності;
- вибору параметрів, за допомогою яких можна змінювати дизайн отриманої web-сторінки тощо.

Для виконання вищевказаних функцій служать наступні основні алгоритми:

- алгоритм конструювання схеми доріг;
- алгоритм автоматичного збереження проміжного результату конструювання схеми в txt-файл в папці History, яка автоматично знищується після виходу із програми;
- алгоритм збереження користувачем результату конструювання в txt-файл;
- алгоритм зчитування з txt-файлу та формування відповідної схеми доріг;
- алгоритм вирівнювання;
- алгоритм пошуку помилок;
- алгоритм побудови web-файлу (даний алгоритм є досить громіздким і містить у собі велику кількість підалгоритмів, детально про які розповімо далі) тощо.

Перераховувати та описувати роботу всіх процедур та функцій не будемо, оскільки цей опис буде досить об'ємним. Достатньо сказати, що розроблений програмний продукт складається з 5-ти модулів (це очевидно з рис. 1) та з 92-х побудованих функцій та процедур. Покажемо скорочену схему алгоритму роботи програми (рис. 2).

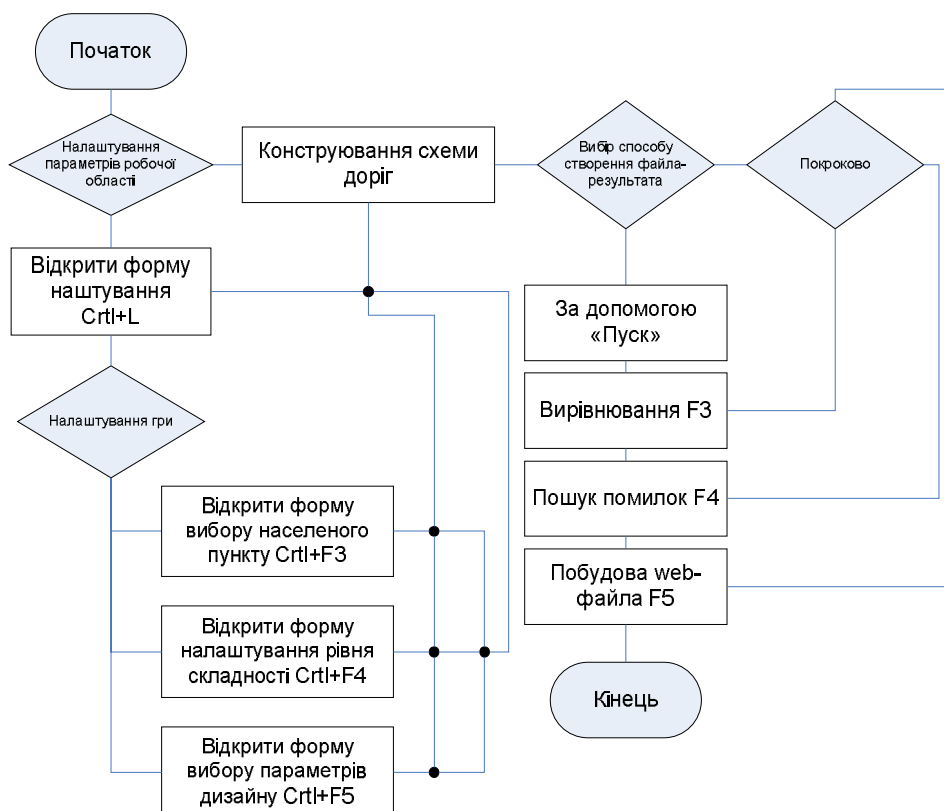


Рис. 2. Схема алгоритму роботи програми

Детальніше функціональні можливості та правила користування розробленою програмою-конструктором покажемо на прикладі побудови конкретної web-гри. Для цього спочатку побудуємо схему доріг, що зображена на рис. 3.

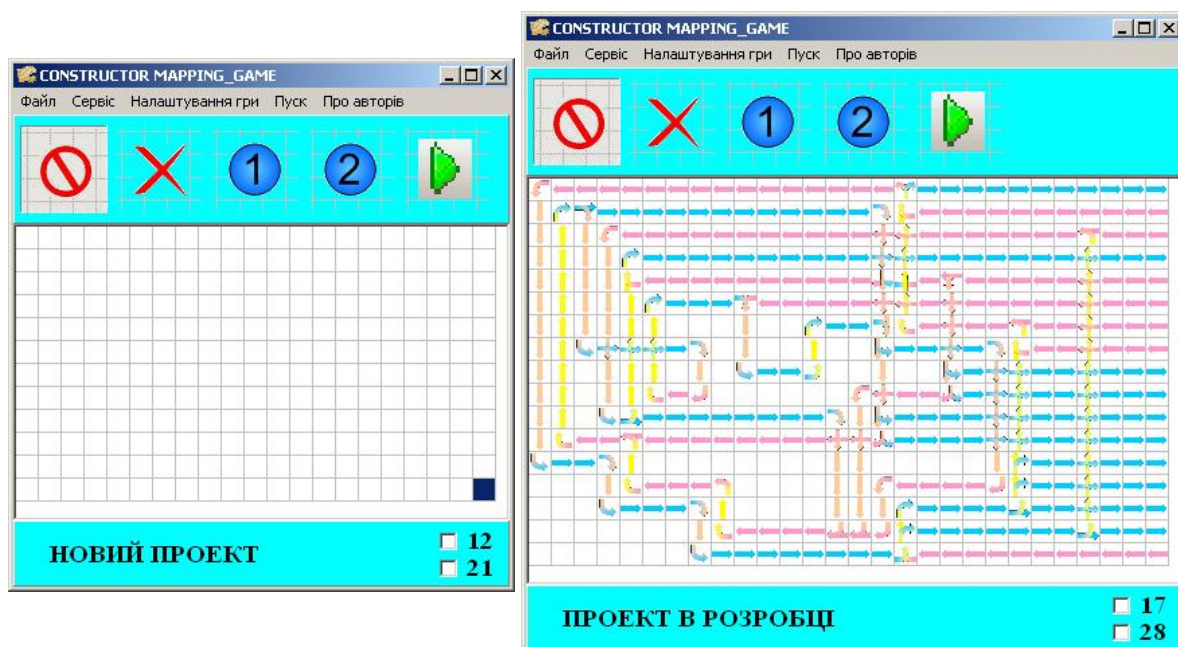


Рис. 3. Приклад, сконструйованої користувачем, схеми доріг

Зауважимо, що з даного рисунку видно, як у стрічці стану перший підпис "НОВИЙ ПРОЕКТ" змінився другим підписом "ПРОЕКТ В РОЗРОБЦІ". Надалі будемо слідкувати за цими підписами та нумерувати їх (це важливо).

Коли схема доріг сконструйована, то її можна зберегти у txt-файл. Крім цього, в процесі конструювання створюється папка History з проміжними txt-файлами. Це дає можливість рятувати

втрачену роботу, оскільки, крім збереження, ми можемо робити за допомогою відкриття не обхідного файлу ще й завантаження схеми. Зауважимо лише, що папка History знищується автоматично після виходу з програми і, що ми завжди можемо розпочати роботу "з нуля". Всі ці можливості реалізуються за допомогою головного меню "Файл" (рис. 4).

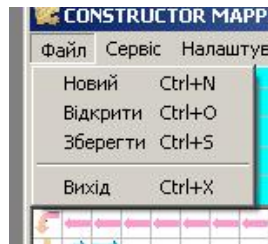


Рис. 4. Функціональні можливості головного меню "Файл"

Головне меню "Сервіс" та відповідне діалогове вікно "Налаштування" (рис. 5) дозволяє змінювати розміри вертикального та горизонтального зазорів. Це необхідно, щоб у різних версіях ОС Windows можна було забрати горизонтальну та вертикальну лінії прокрутки у робочому полі, тобто біля сітки, на якій відбувається конструювання схеми доріг.

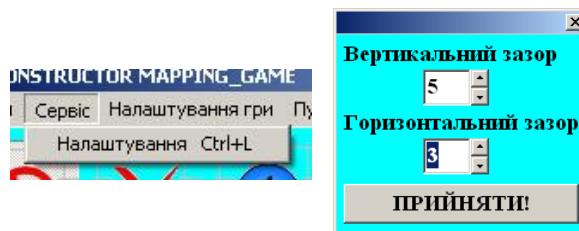


Рис. 5. Функціональні можливості головного меню "Сервіс" та вигляд діалогового вікна "Налаштування"

Як видно зі схеми алгоритму роботи програми (рис. 2), далі роботу можна виконувати двояко:

- по-перше, за допомогою функціональних можливостей головного меню "Налаштування гри" (рис. 6) змінювати ці налаштування і, тим самим, робити web-гру такою, якою захочемо;
- по-друге, можна відразу перейти до створення web-сторінки з налаштуваннями за замовчуванням.

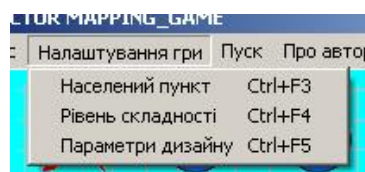


Рис. 6. Функціональні можливості головного меню "Налаштування гри"

Очевидно, що для опису всіх етапів роботи з програмою необхідно, навчитися працювати зі всіма наведеними на цьому рисунку діалоговими вікнами. Зробимо це у вигляді порівняння налаштувань за замовчуванням зі зміненими налаштуваннями та роз'яснення всіх цих змін. Почнемо з діалогового вікна "Населений пункт" (рис. 7).

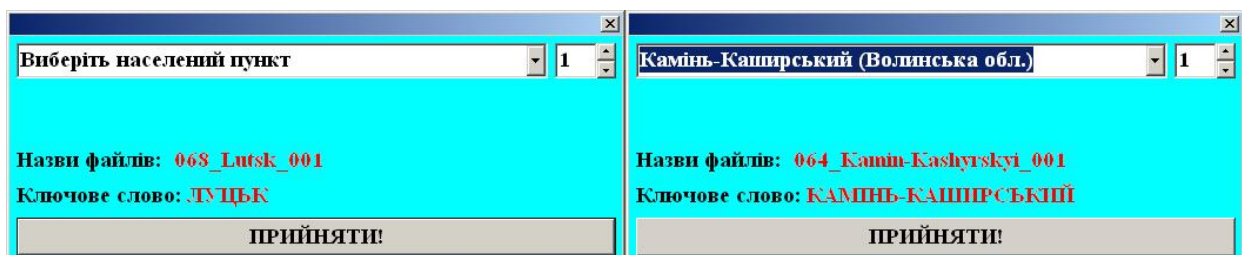


Рис. 7. Принцип роботи з діалоговим вікном "Населений пункт"

До зображеного на цьому рисунку вибору населеного пункту слід додати ще те, що можна змінювати і номер гри (після назви пункту). Як видно, кожен населений пункт ідентифікується спеціальним номером (перед назвою пункту). Для чого це? Щоб уникнути проблем, які можуть виникнути з населеними пунктами, що мають однакову назву.

Далі покажемо, як працювати з діалоговим вікном "Рівень складності" (рис. 8).

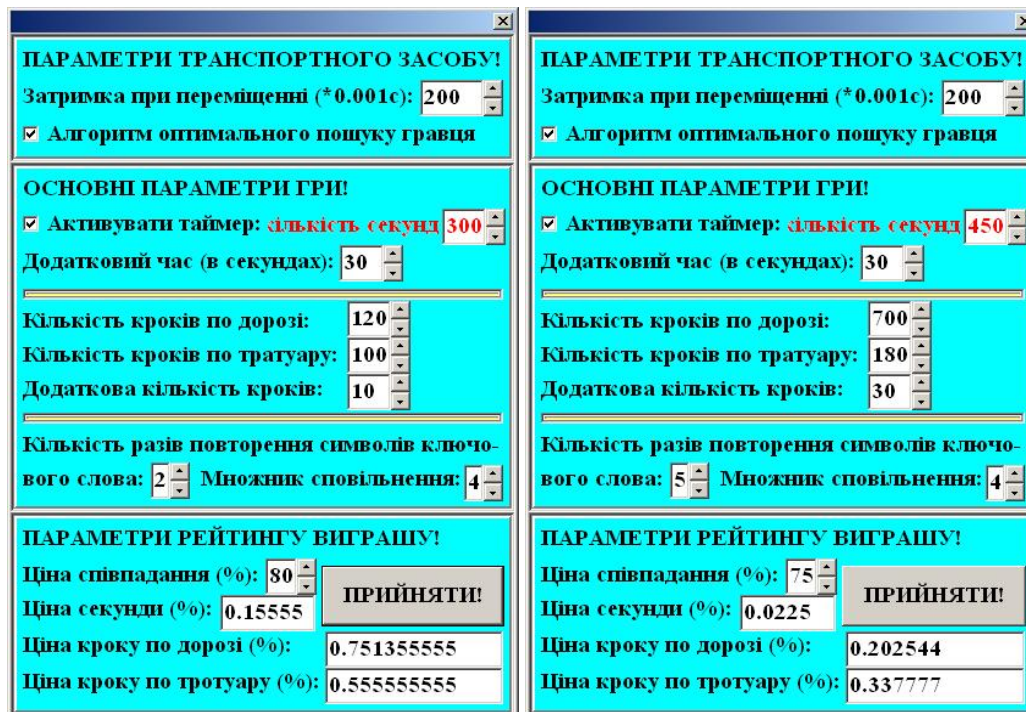


Рис. 8. Принцип роботи з діалоговим вікном "Рівень складності"

Отже, затримку при переміщенні транспортного засобу (ТЗ) не змінимо і залишимо її рівною 0.2 секунди. Щодо алгоритму пошуку найкоротшого маршруту від ТЗ до гравця, який взяв символ, то його також вимикати не будемо. У таймері встановимо 450 секунд. Якщо вимкнути активацію таймера, то автоматично зникає можливість встановлення кількості секунд. Додатковий час в секундах залишаємо також незмінним. Він впливає на додаткові можливості, пов'язані часом, а саме: добавляння додаткового часу, уповільнення ТЗ на додатковий час та зникнення ТЗ на додатковий час. Залежно від необхідності цю характеристику можна змінювати.

Зауважимо, що відміна активації, чи таймера, чи алгоритму оптимального пошуку гравця, допомагає значно скоротити html-файл-результат: (в кодї автоматично зникають зайві функції та змінні).

Ще можна впливати на кількість кроків по дорозі та по тротуару. Як видно, з рис. 8, першу кількість було змінено на 700, а другу – на 180.

А зараз коротко пояснимо основні правила гри, які далі допоможуть при роз'ясненні параметрів у вищевказаному діалоговому вікні:

- гра полягає у тому, що гравець при збиранні усіх символів ключового слова (які можуть мати кратність, тобто зустрічатися певну кількість разів, не рахуючи кількості повторень у ключовому слові) повинен використати кількість кроків, не більше, ніж вказано, якщо не планує використовувати відповідні додаткові можливості;
- аналогічно з активованим таймером;
- якщо якась характеристика вичерпується, то гра автоматично завершується програвом та нульовим рейтингом;
- аналогічно відбувається, коли гравця збиває ТЗ;
- хоча гравець і може застосовувати додаткові можливості, у разі виграшу, додаткові секунди, та кроки на рейтинг не впливають, а в разі залишку вплив є, і ним можемо керувати;
- клавішею "1" можна вимкнути паузу та підказку, якщо вона активована;
- крім цього, є ще одна допомога, яку ніхто не вимкне: як тільки гравець підібрав символ і бачить, що ТЗ наближається, то нехай залишається на місці і не активує підказку (чи паузу).

Після вищеописаних правил очевидно, що кількість разів повторення символів ключового слова це кратність символів, про яку вже було сказано, змінюємо її з 2 на 5. Множник сповільнення 4 означає, що при виборі додаткової можливості сповільнення ТЗ на додаткову кількість секунд, ТЗ на протязі 30 секунд буде рухатися із затримкою при переміщенні  $0.2 \cdot 4 = 0.8$  секунд.

Щодо налаштування параметрів рейтингу виграшу, то тут важливо вміти так підбирати значення, щоб гравець не обманював. Роз'яснимо детальніше, що це означає. Нехай, необхідно сконструювати гру таку, щоб користувачу було вигідно збирати символи по-порядку, тоді необхідно встановити високу ціну співпадіння і низькі ціни на та кроки та секунди, що залишилися. І навпаки, якщо хочемо, щоб користувачу не вигідно було збирати символи по порядку. А тепер, найскладніше, як зробити, щоб ці вигоди були однакові – лише підбором і багаторазовим тестуванням гри. Внесені зміни на ціни дозволяють урівноважити ці вигоди. Оскільки символи на дорозі весь час перемішуються, а на тротуарі, дуже часто зливаються з фоном, залишається побажати гравцю удачі і уважності.

Далі покажемо, як працювати з діалоговим вікном "Параметри дизайну" (рис. 9).

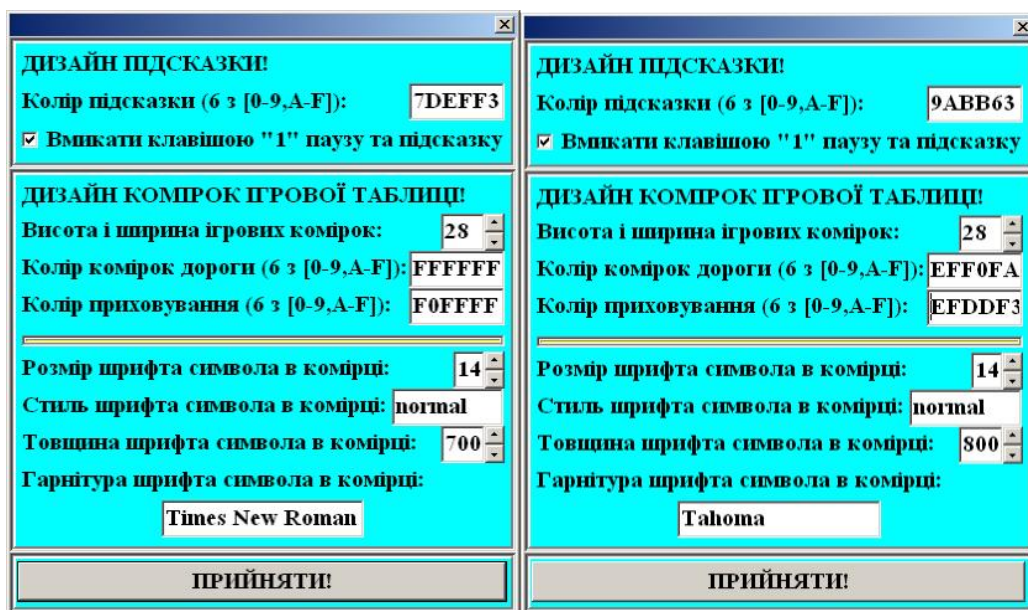



Рис. 9. Принцип роботи з діалоговим вікном "Параметри дизайну"

У лівій формі на цьому рисунку показані параметри за замовчуванням, у правій формі – внесені зміни. Зробимо деякі роз'яснення з приводу внесених змін.

При наведенні курсором на кнопки з додатковими можливостями та на комірки, у яких подаються кількості кроків та секунд, що залишилися, впливає вікно з підказкою. При відведенні курсору або при натисканні на кнопку ця підказка зникає. Є можливість змінювати колір її фону (тільки необхідно це робити так, щоб колір фону не зливався з чорним кольором підпису у цій підказці). Тут також можна вирішувати, дозволяти користувачу робити паузу та підказку за допомогою клавіші "1", чи ні. До вищесказаного необхідно ще додати, якщо у рейтингу є відсотки, що нараховані за "співпадіння", то виклик підказки призводить до зменшення їх на 1%. Це не завжди можна побачити, бо у разі, коли рейтинг перевищує 100%, то у лівій частині стрічки стану, що у грі, все одно залишається підпис "100%" (100% – максимум рейтингу). Щодо висоти та ширини комірок ігрової комірки, то її необхідно узгоджувати параметрами шрифту символів (у наведених змінах все узгоджено) тому, що якщо комірки необхідно розтягуватися для того, щоб уміщати в собі символ ключового слова, то, у разі, коли у відповідному рядку і (або) стовпці зникають символи, комірка буде стягуватися до початкових розмірів, і у зв'язку з постійним перемішуванням символів на дорозі (коли ТЗ наїжджає на символ, то він там зникає і з'являється випадковим чином у іншому вільному від символу місці на дорозі) ці скачки зовсім не сприяють якості дизайну гри.

Будемо вважати, що у "Налаштування гри" необхідні зміни внесені. Тепер залишається лише натиснути кнопку . Як видно зі схеми роботи алгоритму це натиснення рівносильне покроковій реалізації опцій головного меню "Пуск" (рис. 10).

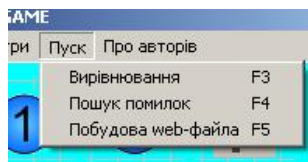




Рис. 10. Функціональні можливості головного меню "Пуск"

І хоча можна замість натискання на кнопку "Пуск" звернутися до опцій головного меню "Пуск" необхідно зважати на те, що у довільному порядку ці опції натискати не потрібно, оскільки вони повинні виконуватися по-порядку. З вищесказаного відомими є назви 2-х перших станів, або підписів у стрічці стану. Розглянемо наступні стани: 3-й стан має назву "ПРОЕКТ ВИРІВНЯНО", 4-й стан має назву "ПРОЕКТ КОМПІЮЄТЬСЯ", 5-й стан має назву "ПРОЕКТ ВІДКОМПІЛЬОВАНО", 6-й стан має назву "WEB-ФАЙЛ СТВОРЮЄТЬСЯ" і останній – 7-й стан має назву "WEB-ФАЙЛ СТВОРЕНО". І хоча вирівнювання можна і не робити, перед побудовою web-файлу, якщо проект не був відкомпільованим, то компіляція запускається автоматично.

Тепер детальніше зупинимося на вище описаних поняттях. Що таке компіляція у розумінні розробленого програмного продукту. Мета компілятора не допустити побудову web-сторінки, на якій через помилку конструктора ТЗ раптом перестане рухатися. По ідеї рух повинен бути безперервним до тих пір, доки користувач не зупинить ТЗ. І основна ідея пошуку помилок полягає у тому, щоб у суміжних клітинках, напрямки відповідних входу і виходу співпадали: наприклад, випадок  призведе до появи повідомлення про помилку (рис. 11), при цьому ця помилка, буде автоматично виправлена на .

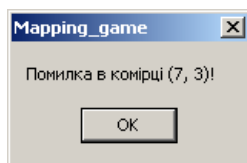





Рис. 11. Повідомлення про помилку

При цьому не завжди компілятор робить автоматичне виправлення помилок, і якщо робить то воно не завжди повністю правильне. У такому випадку необхідно натиснути кнопку  і увімкнути режим "Правка". У цьому режимі за замовчуванням відбувається заміна вмісту комірок, по яких рухаємося, на порожність, тобто відбувається стирання. Якщо застосовувати контекстне меню, то можна вибирати на що будемо замінювати вміст у комірок, по яких рухаємося. Залишається лише додати те, що за допомогою натиснення на кнопки  і  та руху стрілками по комірках робочої області формуються дороги з одностороннім та правим двостороннім рухом.

Нехай "ПРОЕКТ ВІДКОМПІЛЬОВАНО". Почнемо процес побудови web-файлу. Якщо схема доріг є замкненою, то перед створенням файлу, з'являється лише одне повідомлення (рис. 12).

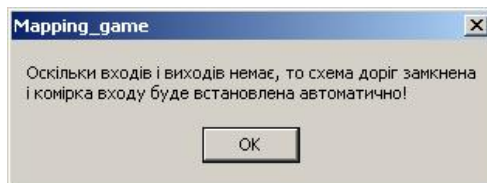


Рис. 12. Повідомлення про замкненість схеми доріг

Далі продовжимо роботу зі схемою доріг (рис. 3). Після включення опції "Побудова web-файлу" з'являється повідомлення використання входів як початкових (рис. 13). Використаємо всі входи.

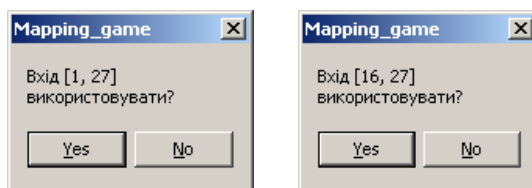


Рис. 13. Повідомлення про використання входів як початкових

Зауважимо, якщо ми хотітимемо використовувати усі входи, то про останній вхід програма навіть не запитуватиме, оскільки необхідно, щоб був хоча б 1 початковий вхід.

Далі у програмі необхідно встановити усі зв'язки між виходами та входами (рис. 14). Аналогічно, і у попередньому випадку, в програмі передбачено, що кожному виходу повинен відповідати хоча б 1 вхід.

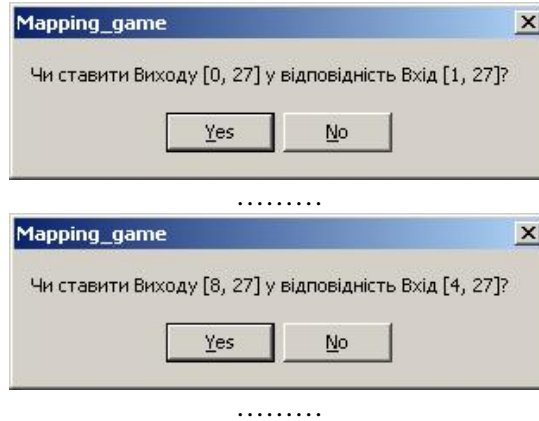



Рис. 14. Повідомлення про використання входів як початкових

Результатом одержаного розв'язку буде поява в папці, де знаходиться exe-файл даної програми, папки з назвою REZULTAT. При цьому, якщо, у папці уже була папка з назвою REZULTAT, то вона буде перейменована на REZULTAT\_(-1). Аналогічно відбувається, якщо у папці з exe-файлом уже є і папка з назвою REZULTAT, і папка з назвою REZULTAT\_(-1) (рис. 15).



Рис. 15. Створення папки REZULTAT

В цій папці автоматично будуть створені файли 0.jpg і 064\_Kamin-Kashyrskyi\_001.html. Якщо не закриваючи програму захочемо змінити гру з такою ж назвою, достатнього розпочати "НОВИЙ ПРОЕКТ" зробити необхідні зміни (при цьому не забути ще раз зробити аналогічні налаштування в діалоговому вікні "Населений пункт") і натиснути на кнопку . У разі, коли у новому проекті змінюються налаштування в діалоговому вікні "Населений пункт", то у папці REZULTAT, з'явиться ще один файл. І для остаточного завершення роботи, залишається в одержаній папці створити папку PHOTO, і розмістити в ній файл для фону 064\_Kamin-Kashyrskyi\_001.jpg (рис. 16).

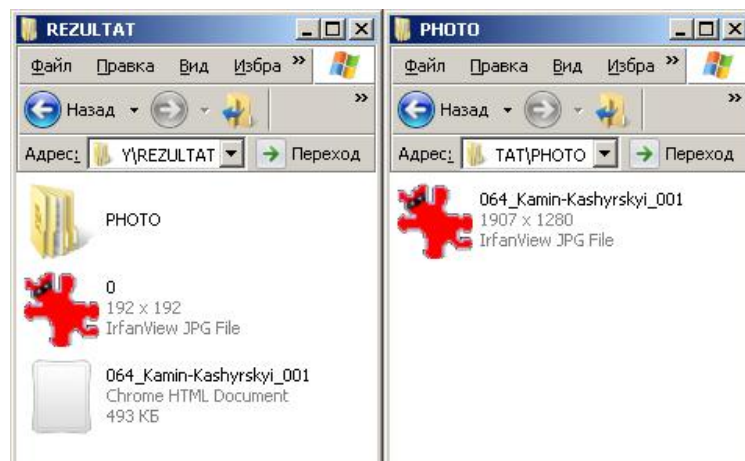


Рис. 16. Розташування отриманих та добавлених файлів, для ефективної роботи web-гри



Покажемо, як виглядає кінцевий результат – web-гра в процесі (рис. 17).



Рис. 17. Вигляд кінцевого результату – web-гри в процесі

Повний код одержаного html-файлу складається з 7880 рядків, а його об'єм сягає 493 КБ.

**Висновки.** Аналізуючи вище викладене можна констатувати, що поєднання технологій автоматного програмування з явним виділенням станів та об'єктно-орієнтованого програмування повністю себе виправдовує як технологія створення динамічних web-сторінок. А це в свою чергу дає наступні переваги:

- по-перше, використання розробленої програми для конструювання вищеописаного класу web-ігор дозволяє значно зменшити об'єм роботи (а відповідно і затрачений на цю роботу час);
- по-друге, можна створювати такі програми не лише для конструювання ігор, а і для інших динамічних web-ресурсів, за допомогою яких можна досліджувати різні фізичні, хімічні та ін. процеси (наприклад, використовуючи подібну логіку можна моделювати різну топологію комп'ютерних мереж та досліджувати їх поведінку);
- по-третє, розробка подібних програм-конструкторів суттєво може допомогти у розробці та наповненні контенту якогось спеціалізованого web-сайту.

1. Капітонова Ю.В. Основи дискретної математики / Ю.В Капітонова, С.Л Кривий, О.А Летичевський та ін. – Київ : “Наукова думка”, 2002. – 580 с.
2. Коцюба А.Ю. Застосування двовимірних клітинних автоматів в моделюванні правостороннього руху транспортного засобу по створеній користувачем схемі доріг // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – Луцьк, 2014. – Вип. 15. – С. 94-98.
3. Непейвода Н.Н. Стили и методы программирования / Н.Н. Непейвода. – М. : ИНТУИТ.РУ “Интернет-университет Информационных Технологий”, 2005. – 320 с.
4. Поликарпова Н.И. Автоматное программирование / Н.И. Поликарпова, А.А. Шалыто. – СПб. : “Питер”, 2009. – 420 с.
5. Nagel K. A cellular automaton model for freeway traffic / K. Nagel, M. Schreckenberg // J. Physique I France. – 1992, vol. 2 – P. 2221-2229.
6. <http://is.ifmo.ru>: сайт по автоматному програмуванню и мотивации к творчеству // Кафедра “Технологии программирования” Санкт-петербургского государственного университета информационных технологий, механики и оптики.
7. Архангельский А.Я. Программирование в C++ Builder / А. Я. Архангельский. – М.: ООО “Бином-Пресс”, 2010. – 896 с.
8. Архангельский А.Я. Язык C++ в C++ Builder. Справочное и методическое пособие / А. Я. Архангельский. – М: “Бином”, 2007. – 1012 с.
9. Пахомов Б. Самоучитель C/C++ и C++ Builder 2007 / Б. Пахомов. – Санкт-Петербург: “БХВ Петербург”, 2008. – 672 с.
10. Флэнаган Д. JavaScript. Подробное руководство / Д. Флэнаган. – СПб. : “Символ-Плюс”, 2008. – 992 с.