

УДК 004.054

Використання алгоритмів штучного інтелекту у створенні програмних додатків ігрового спрямування засобами мови програмування

Objective-C

Ругало Дмитро

Науковий керівник канд. техн. наук, Присяжнюк О.В.

Центральноукраїнський державний педагогічний університет імені

Володимира Винниченка, м. Кропивницький, Україна

В статті розглядаються особливості використання елементів штучного інтелекту у створенні програмних додатків ігрового спрямування засобами мови програмування Objective-C. Проаналізовано переваги та особливості платформи iOS та її потенціальні можливості. В якості мобільного ігрового додатку обрано ігровий додаток make a square. Проаналізовано особливості роботи Objective-C з графічними елементами та логікою гри з використанням інструментів штучного інтелекту і зроблені певні висновки про потенційні можливості даного інструменту.

Ключові слова: програмні додатки, логічні ігри, Objective-C, iOS.

Using Artificial Intelligence Algorithm to Build Objective-C Gaming

Software Applications

Rugalo Dmytro

Scientific supervisor: Candidate of Technical Sciences, Prsyazhnyuk O.V.

The Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,

Kropyvnytsky, Ukraine

The article discusses the peculiarities of the use of artificial intelligence elements in the creation of software applications of game direction by means of the Objective-C programming language. The advantages and features of the iOS platform and its potential are analyzed. The make a square game application is selected as a mobile gaming application. The peculiarities of Objective-C work with graphical elements and game logic using artificial intelligence tools are analyzed and some conclusions are made about the potential capabilities of this tool.

Keywords: software applications, logic games, Objective-C, iOS.

Постановка проблеми.

У наш час логічні ігри є популярним ігровим жанром. Грати в логічні міні-ігри люблять усі, незалежно від статі, віку, займаної посади та соціального статусу. Такі ігри дають нагоду людям відволіктись від повсякденності на роботі, домашніх турбот і стресу, якого вистачає в реальності. Думки людини

перебудовуються на вирішення певної задачі в логічній грі. Коли дану задачу успішно вирішено, людина відчуває певне інтелектуальне задоволення, яке є приводом для поліпшення настрою. При цьому є можливість поєднати приємне з корисним. Логічні ігри чинять розвиваючий ефект на людину. У процесі гри відбувається тренування логічного мислення, уваги, виробляється вміння помічати й запам'ятовувати різні малопомітні деталі, прораховувати дії на кілька кроків наперед. Логічні ігри допомагають розумовому розвитку дітей дошкільного та шкільного віку. Присутній також ще один плюс, характерний для сучасних комп'ютерних ігор – творці логічних ігор сьогодні приділяють достатньо уваги та їх естетичній стороні. Логічні ігри відрізняються модульністю, об'єктно-орієнтованою архітектурою, високою якістю комп'ютерної графіки, якісно промальованими задніми планами, спецефектами. Все це в комплексі надає задоволення гравцям. Тому зараз досить існує значна кількість шанувальників логічних ігор серед людей, які не є професійними користувачами комп'ютера. У логічних іграх нескладні і зрозумілі правила, і звичайно гра має ще і систему допомоги гравцям (інтерактивні підказки), яка допоможе віднайти вірне рішення.

Аналіз досліджень та публікацій.

Перспективною для проведення подальших досліджень і зручною з точки зору ілюстративності алгоритмів є iOS - операційна система, яка працює виключно на пристроях iPhone, iPod touch і iPad. Операційну iOS систему розробляє корпорація Apple, що за ринковою долею капіталу сьогодні є найбільшою в світі. iOS має широке розповсюдження, а отже, має популярність серед значної кількості користувачів. Операційна система керує обладнанням пристроїв і надає технології, необхідні для написання платформенно-орієнтованих додатків. Операційна система поставляється з системними додатками, такими як «Телефон», «Пошта» і браузер «Сафарі», які надають стандартні системні служби користувачу.

Набір засобів розробки додатків для iOS (iOS SDK) включає інструменти та інтерфейси, необхідні для розробки, установки, запуску і тестування

платформенно-орієнтованих додатків. Платформенно-орієнтовані додатки створюються за допомогою системних бібліотек iOS і мови програмування Objective - C і працюють безпосередньо в операційній системі iOS. На відміну від веб-додатків, такі додатки встановлюються на пристрій фізично і тому завжди доступні користувачеві, навіть якщо пристрій знаходиться в режимі «Польоту». Вони розташовуються разом з іншими системними додатками, і можуть бути синхронізовані з комп'ютером користувача за допомогою програми iTunes. Актуальність дослідження зумовлює високий ступінь популярності операційної системи, а також перспектива розвитку логічного жанру відеоігор.

Значний інтерес у сучасних дослідженнях представляють вивчення штучного інтелекту, а також його практичні застосування, зокрема у теорії ігор. З огляду на це доцільною є інтеграція особисто створеного алгоритму машинного навчання покрокових ходів у навчальну логічну гру.

Мета статті. Метою статті є з'ясувати і продемонструвати принципи практичного застосування мови програмування Objective-C для побудови програм ігрового жанру для відповідної мобільної платформи на прикладі розробки програмного продукту мультимедійної гри логічного жанру під назвою “Make a square”.

Виклад основного матеріалу (результатів) дослідження.

Задача створеного алгоритму штучного інтелекту полягає у виконанні покрокових ходів на основі аналізу машиною попередніх ходів гравця. Стратегія машинного навчання полягає у вивченні комп'ютером передбачуваного бажаного наступного ходу гравця і перехоплення ініціативи комп'ютером – з метою недопущення виконання гравцем виграшного ходу.

Перспективною для написання додатків ігрового спрямування є операційна система iOS, компанії Apple Inc.

Apple Inc. — компанія, що проектує та виробляє електроніку, програмне забезпечення та комерційні сервери.

Objective-C — рефлексивна, високорівнева об'єктно-орієнтована мова програмування загального призначення, розроблена у вигляді набору розширень стандартної C, використовується для написання додатків для операційних систем Apple.

Розроблена компанією Apple, використовується в основному у Mac OS X та GNUStep — середовищах, розроблених на основі стандарту OpenStep, та Cocoa — бібліотеки компонентів для розробки програм. Програму на Objective-C, що не використовує цих бібліотек, можна скомпілювати для будь-якої платформи, яку підтримує gcc компілятор з підтримкою Objective-C.

Objective-C є розширенням C, тому будь-яку програму на C можна скомпілювати компілятором Objective-C.

ООП в Objective-C включає інтерфейси, класи, категорії. Реалізовано одиничне, невіртуальне наслідування. Відсутній єдиний базовий клас для всіх об'єктів. Всі методи в класі — віртуальні. Категорія — парадигма, яка дозволяє описувати інтерфейс з методами, які не обов'язково необхідно імплементувати.

Однією з відмінних рис Objective-C є її динамічність: ряд рішень, які зазвичай приймаються на етапі компіляції, відкладаються безпосередньо до етапу виконання. Ще однією особливістю мови є те, що вона орієнтована на повідомлення, в той час як C++ орієнтована на функції, тобто в ній виклики методу інтерпретуються не як виклик функції, а як відправлення повідомлення (з ім'ям і аргументами) об'єкту, подібно до того, як це відбувається в Smalltalk. Даний підхід дає переваги: будь-якому об'єкту можна надіслати будь-яке повідомлення. Об'єкт може замість обробки повідомлення просто переслати його іншому об'єкту для обробки (делегування), зокрема, так можна реалізувати розподілені об'єкти (об'єкти, що знаходяться в різних адресних просторах на різних комп'ютерах). Прив'язка повідомлення до відповідної функції відбувається безпосередньо на етапі виконання.

Мова Objective-C підтримує нормальну роботу з метайнформацією - у об'єкта безпосередньо на етапі виконання можна запитати його клас, список методів (з типами переданих аргументів) і змінних, перевірити, чи є клас

нащадком заданого, чи підтримує він заданий протокол тощо. У мові є нормальна підтримка протоколів (поняття інтерфейсу об'єкта та протоколу чітко розділені). Для об'єктів підтримується успадкування (не множинне), для протоколів підтримується множинне спадкування. Об'єкт може бути успадкований від іншого об'єкта і відразу декількох протоколів. На даний момент мова Objective-C підтримується компіляторами Clang і GCC (під управлінням Windows - Windows використовується у складі MinGW або cygwin). Досить багато в мові перенесено на runtime-бібліотеку і значною мірою залежить від неї. Разом з компілятором gcc поставляється мінімальний варіант такої бібліотеки. Також можливо завантажити runtime-бібліотеку компанії Apple: Apple's Objective-C runtime.

Xcode – інтегроване середовище розробки для побудови додатків для OS X і iOS, розроблене корпорацією Apple. Останню версію Xcode можна завантажити з Mac App Store. Остання версія – 5, не підтримується застарілими версіями OS X.

Основним додатком пакету є вбудоване середовище розробки, яка називається Xcode. Крім цього, пакет Xcode включає в себе більшу частину документації розробника від Apple і Interface Builder - додаток, що використовується для створення графічних інтерфейсів (його буде розглянуто більш детально нижче).

Пакет Xcode включає в себе змінену версію вільного набору компіляторів GNU Compiler Collection і підтримує мови C, C++, Objective-C, Objective-C++, Java, AppleScript, Python і Ruby з різними моделями програмування, включаючи Cocoa, Carbon і Java. Сторонніми розробниками реалізовано підтримку GNU Pascal, Free Pascal, Ada, C#, Perl, Haskell і D. Пакет XCode використовує GDB в якості бек-енда для свого відлагоджувальника.

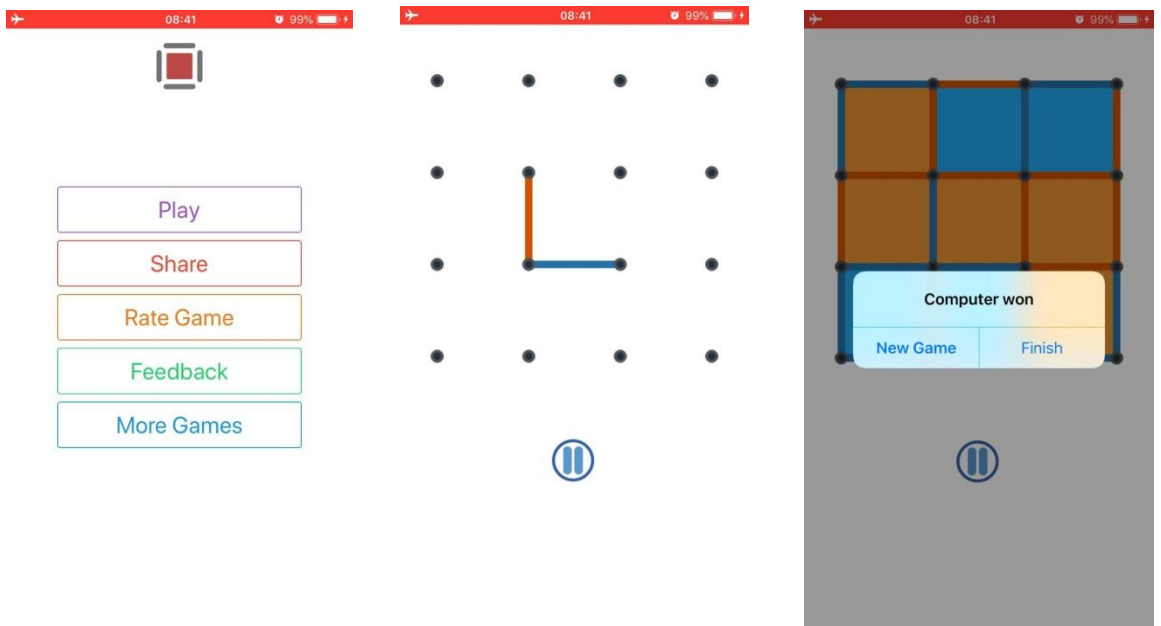
Interface Builder – додаток від Apple для операційної системи Mac OS X. Він є частиною Xcode, спеціальної системи інструментів для розробників Apple Developer Connection. Interface Builder дозволяє розробникам Cocoa і Carbon створювати графічні інтерфейси для додатків. Результат розробки зберігається

у файлі з розширенням .Nib (скорочення від NeXT Interface Builder), проте останнім часом частіше для файлів використовується розширення .Xib.

Interface Builder походить від однойменного програмного забезпечення ОС NeXTSTEP, яке мало таку ж назву. Версія Interface Builder також використовується в розробці програм OpenStep. 27 березня 2008 року, в рамках iPhone SDK Beta 2, було запущено спеціальну версію Interface Builder, яка надає можливості розробки графічних інтерфейсів для iPhone-додатків. У Xcode 4 Interface Builder не існує в рамках окремого додатка, функції інтегровані безпосередньо до Xcode.

Interface Builder (IB) вперше було представлено в 1986 році і було написано на Lisp. Interface Builder був задуманий і розроблений Д. Халлотом з використанням інструментів об'єктно-орієнтованого програмування в ExperLisp і глибоко інтегрований з інструментами Macintosh. С. Джобс, тодішній директор Apple, визнав представлені розробки значущими, і включив їх до NeXTSTEP, і до 1988 року IB став частиною NeXTSTEP 0.8. Це було перше комерційне ПЗ, яке дозволяло розробляти елементи інтерфейсу, такі як кнопки, меню і вікна; ці елементи могли бути інтегровані до інтерфейсу натисненням миші. Interface Builder надає палітри, або колекції, об'єктів для користувача інтерфейсу для Objective-C розробників.

В ході дослідження розроблено програмний продукт, створений у середовищі Apple Xcode з використанням мови програмування Objective-C використання і тестування якого описано нижче. Після запуску у програмі наявні пункти меню, відтворені кнопками в центральній частині екрана:



1. Play (Гра) – під час застосування даного пункту управління передається до вікна з ігровим полем, у якому розміщені ігрові елементи.
2. Share – під час застосування даного пункту користувач може поділитися грою з друзями в деяких соціальних мережах.
3. Rate Game – під час застосування даного пункту в Safari автоматично відкривається сторінка додатку, для її оцінки.
4. Feedback – під час застосування даного пункту користувач отримує зворотній зв'язок з розробником додатку.
6. More Games – під час застосування даного пункту відбувається показ спливаючого вікна з інформацією про інші ігрові додатки розробника даної гри.

Після першого запуску програми користувач бачить пункти меню, відтворені кнопками в центральній частині екрана. Вгорі в екрані меню вміщено зображення ігрового продукту.

При переході на екран гри користувач розпочинає ігровий процес. Внизу ігрового поля міститься кнопка паузи гри або виходу до головного меню, саме ігрове поле розміщується по центру і являє собою повнорозмірну сітку (розміром 3x3 з вершин квадрата гранями 3x3). Принцип гри полягає в

замальовуванні комірок кольором гравця. Мета гри – зафарбувати більшу частину комірок на полі ніж суперник.

За один хід користувач малює одну грань (лінія яка поєднує дві сусідні вершини по вертикалі чи горизонталі) з числа тих, які не були замальовані раніше. Три грані однакового кольору утворюють замкнутий квадрат (автоматично домальовується четверта лінія) і квадрат вважається зафарбованим. Стратегія гри полягає в тому, щоб з'єднувати вершини так, щоб якумога швидше замальовати більшу частину квадратів, таким чином виграшна стратегія - замальовувати за один хід декілька квадратів збільшуючи шанси на виграш.

Ходи здійснюються по черзі, якщо обраний хід виявився переможним, гравцю чи машині надається право позачергового додаткового ходу. Гра вважається завершеною, якщо замальовані всі комірки ігрового поля.

Результат гри показується у впливаючому вікні посередині екрану. В цьому самому полі гравець може обрати: грати в нову гру (new game), чи завершити (finish) та повернутися до головного меню.

Опишемо створення інтерфейсних файлів гри. Після створення та початкової конфігурації проекту переходимо до створення інтерфейсних Storyboard-файлів. Ці файли необхідно створити для кожного екрану, крім того, для кожного пристрою потрібен власний файл. Після додавання інтерфейсного файлу пов'язуємо об'єкт файлу з класом, що відповідає за відображення інтерфейсу на екрані пристрою. Потім додаємо та налагоджуємо для кожного екрану елементи управління, які присутні на ньому: кнопки, контейнери графічних зображень, написи тощо.

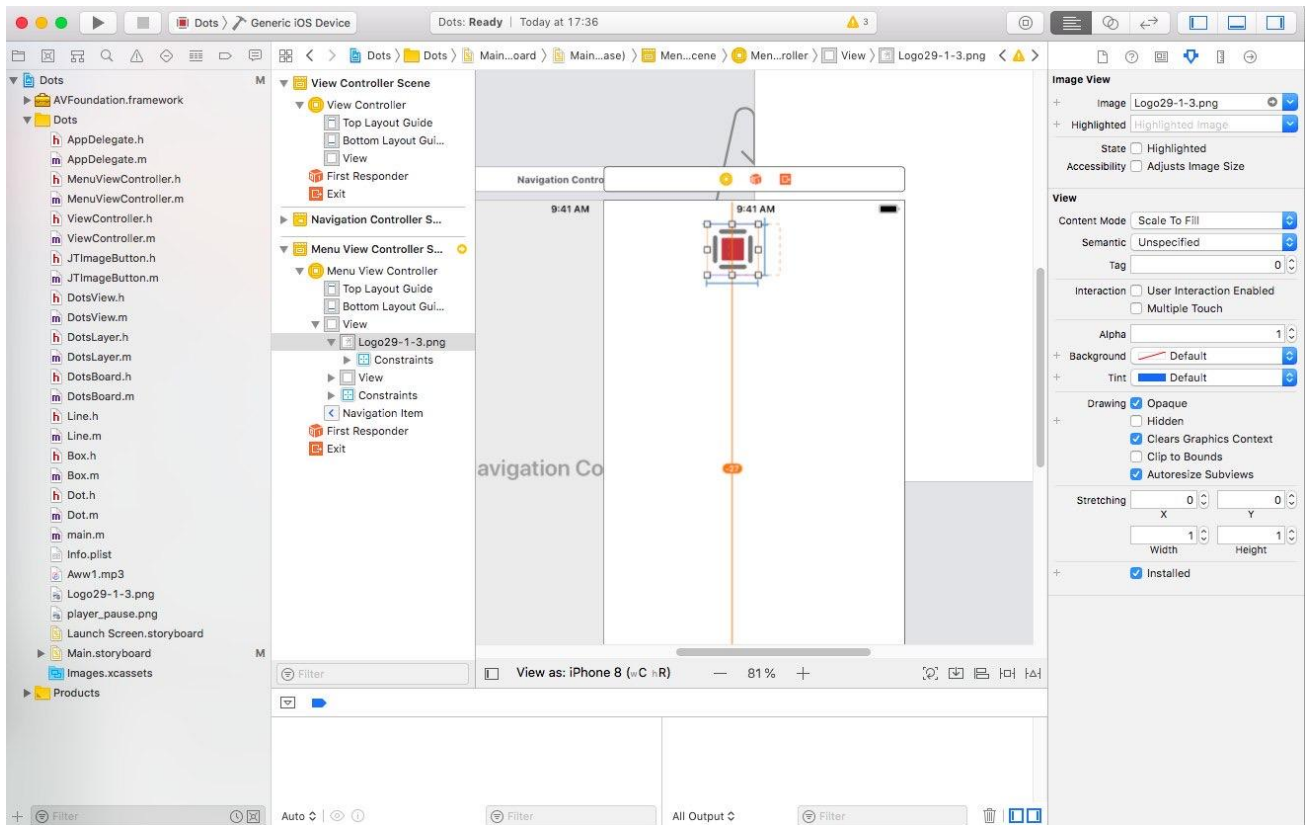


Рис. 1 Створення інтерфейсних файлів програми.

Додаток створюється для планшета і смартфона одночасно, адже storyboard файл дозволяє забезпечити універсальну підтримку екранів будь якого формату .

Для додатку ігрового спрямування доцільною є наявність звукових ефектів під час взаємодії з елементами ігрового поля та керування. Створено функцію відтворення звукових файлів з метою підвищення естетичної привабливості продукту:

```

- (void) configureAudio
{
    NSString *cheerPath = [[NSBundle mainBundle]
                           pathForResource:@"Aww1" ofType:@"mp3"];
    NSURL *cheerURL = [NSURL fileURLWithPath:cheerPath];
    AudioServicesCreateSystemSoundID((__bridge CFURLRef)cheerURL,
    &_cheerSound);
}

```

В якості параметру до створеної функції передається назву відтворюваного аудіофайлу, і в залежності від значення звукової змінної ініціалізуємо екземпляр програвача. Функцією використовується для відтворення звуків ефектів (підтримуються файли формату wav та mp3).

Висновки та перспективи подальших пошуків у напрямі дослідження.

У статті було продемонстровано реалізацію ряду концепцій програмування мовою Objective-C, використання інструментарію, який надають фреймворки Cocoa, AudioToolbox тощо.

Обрана тематика дослідження є перспективною, у майбутньому планується вдосконалювати створений програмний продукт шляхом удосконалення графічного та звукового оформлення, підтримки мережевого режиму гри, добавлення можливості вибору мови ігрового інтерфейсу.

Список використаних джерел.

1. Гэлловей М. Сила Objective-C 2.0. Эффективное программирование для iOS и OS X. – СПб.: Питер, 2014. – 304 с.
2. Далримпл М. Objective-C 2.0 и программирование для Mac. – Вильямс, 2010. – 315 с.
3. Донован Д. Системное программирование. – М.: Мир, 1975. – 540 с.
4. Елисеев Д. Разработка и продажа программ для iPhone и iPad. – БХВ-Петербург, 2010. – 336 с.
5. Здзиарски Д. iPhone. Разработка приложений с открытым кодом. – БХВ-Петербург, 2011. – 368 с.
6. Здзиарски Д. iPhone SDK. Разработка приложений. – БХВ, 2012. – 506 с.
7. Кнастер С. Objective-C. Программирование для Mac OSX и iOS. – Вильямс, 2012. – 304 с.