ВСТУП

Штучний інтелект – це галузь комп’ютерних наук, що займається автоматизацією розумного поводження агентів, які одержують у результаті актів сприйняття інформацію про навколишнє середовище і виконують дії, що реалізують функцію від результатів сприйняття і попередніх дій [2].

Дана гра відноситься до логічних ігор. При великому полі тяжко вирішати цю задачу за допомогою жорсткого алгоритму. Тому перспективним є використання систем штучного інтелекту для вирішення цієї проблеми.

Гра «Довгі нарди» є однією з найпопулярніших настільних ігор. В даний час у зв'язку з розвитком інформаційних технологій гра має безліч реалізацій на комп'ютерах. Це більш зручно, так як процес підготовки до гри (розстановка шашок, складання дошки, переміщення шашок, пошук позиції на яку потрібно перемістити шашки і т.п.) займає досить багато часу. Так само в процесі гри виникає безліч розбіжностей з приводу правил гри і шахрайства. Комп'ютерна програма присікає шахрайство і містить в собі інтегровану довідку за правилами гри. До того ж не потрібно шукати другого гравця для гри - грати можна проти комп'ютера.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

4

08-26.КП.010.11.076.ПЗ

Програма буде розрахована на користувачів усіх вікових категорій. Так як не всі користувачі (особливо діти) досконало володіють комп'ютером і технікою гри, програма буде мати максимально спрощений інтерактивний інтерфейс і спрощену інтелектуальну складову комп'ютера.

Об’єктом даного курсового проекту є вивчення технологій застосу­вання штучного інтелекту для розв’зання інтелектуальних ігрових задач, що не мають чітко визначених виграшних стратегію.

Предметом проектування є комп’ютерне моделювання ігрової ситуації "Довгі нарди".

Метою проектування є створення комп’ютерної програми, яка б успішно грала у „Довгі нарди” з людиною.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

5

08-26.КП.010.11.076.ПЗПЗ

Нарди - настільна гра для двох гравців на спеціальній дошці, розділеної на дві половини. Мета гри - кидаючи кості й пересуваючи шашки відповідно до випавших очків, пройти шашками повне коло по дошці, зайти ними в свій будинок (іноді фігурує вуличне назву хата) і викинути їх за дошку раніше, ніж це зробить супротивник. Є дві основні різновиди - довгі і короткі нарди.

При грі в довгі нарди пересування шашок проводиться проти годинникової стрілки. З вихідного положення чорні шашки рухаються з правої верхньої чверті в ліву верхню, потім в нижню ліву і, нарешті, в свій будинок - нижню праву чверть. Будинок білих знаходиться у верхній лівій чверті, куди шашки повинні рухатися за наступним маршрутом: нижня ліва чверть - нижня права чверть - верхня права чверть - будинок.

У процесі гри кожен учасник здійснює кидки тільки двома зарами. Після кидка гравець повинен пересунути одну з своїх шашок на число клітин, рівне що випав на одному з кубиків, а потім будь-яку шашку на число клітин, показане іншим кубиком. Наприклад, на зарах випало чотири-два. У цьому випадку гравець пересуває одну шашку на чотири клітини, іншу - на дві. Можна пересунути тільки одну шашку на шість клітин.

У ситуації, коли шашки не йдуть, тобто гравець не може їх пересунути на ту кількість лунок, яке випало на зарах (шашки потрапляють на зайняті лунки), всі очки пропадають, і гравець пропускає хід.

У ситуації, коли гравець може виконати будь-який з двох ходів, але тільки один, він повинен вибрати більший. Наприклад, випала комбінація шість-чотири. Гравець може зробити хід на шість лунок або на чотири. Правильний хід - на шість лунок. При цьому менші окуляри згоряють.

Якщо на зарах випадає дубль, тобто однакову кількість очок, то гравець повинен зробити чотири хода, пересунувши шашки на стільки клітин, скільки показують зари. Такий результат кидка називається куш.

**2 РОЗРОБКА ОЦІНОЧНОЇ ФУНКЦІЇ**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

6

08-26.КП.010.11.076.ПЗ

.

Cуттю оціночної функції є отримання числової оцінки того, наскільки вигідно зробити у тій або іншій ігровий ситуації конкретний хід. Очевидно, у нашому випадку, головною метою ходу є хід шашкою проти годинникової стрілки на стільки кроків, на скільки випало на зарах. Хід здійснити можливо двома шашками окремо, або однією, хід кроків якої буде сумою кроків двох шашок. Метою гри є переміщення своїх шашок в "дім" (Рис. 2.1) і блокування ходів противнику. Виграш отримує той хто перший перемістив всі свої шашки в "дім".

В грі забороняються такі дії :

* Cтавити шашку на лунку, занятою шашкою соперника.
* Закривати всі шашки противника перед першою його шашкою.
* Передвигати дві шашки на число клітинок, показаних на одному із зарів.

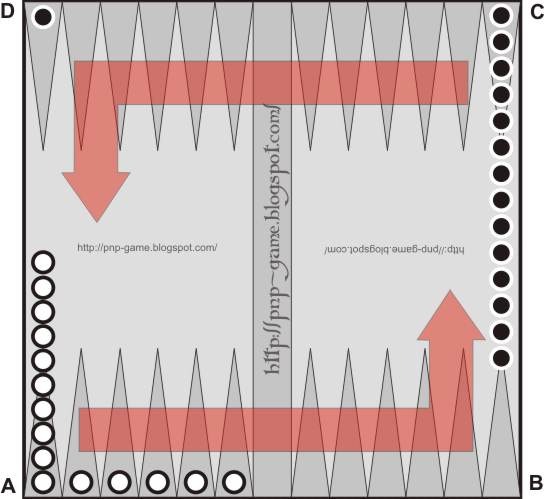


Рисунок 2.1 - Напрямок руху шашок

Для прискорення аналізу можливостей здійснення кроку, процедуру аналізу можна розбити на два етапи:

1. Аналіз можливості заповнення вільної лунки. Для кожної з лунок такий аналіз здійснюється дуже швидко і не залежить від розмірів ігрового поля.
2. Проста оціночна функція. Вона дозволяє швидко, але дуже приблизно оцінити поточну зайнятість ігрового поля. При цьому за досить короткий проміжок часу можна побудувати дерево станів.

Оскільки розмір ігрового поля невеликий, то при нескладній функції оцінювання є можливість проглянути все дерево станів (до термінальних вершин). Таким чином в якості оціночної функції використаємо суму ваг кожної з позицій шашки, що знаходяться на полі. Така оціночна функція дозволяє швидко оцінити положення шашок на ігровому полі шляхом перегляду всього списку, що відповідає за збереження стану.

Процес знаходження оптимального ходу об’єднується в одну процедуру, яка заснована на альфа-бета алгоритмі і виконується в такій послідовності:

1. Генерація списку позицій, які отримуються в результаті всіх можливих ходів шашок.
2. Оцінка термінальних вершин (або всіх вершин списку у випадку досягнення максимальної глибини) за допомогою функції оцінювання.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

7

08-26.КП.010.11.076.ПЗ

1. Оцінка нетермінальних вершин шляхом рекурсивного виклику процедури для кожної з вершин.
2. Обрання найкращої вершини і звуження інтервалу Альфа-Бета.

**3 РОЗРОБКА СХЕМИ АЛГОРИТМУ**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

8

08-26.КП.010.11.076.ПЗ

Одним з найважливіших блоків даної програми є контроль правил гри. Для перевірки можливості ходу по заданих параметрах була створена функція.

Функція виконує наступні перевірки:

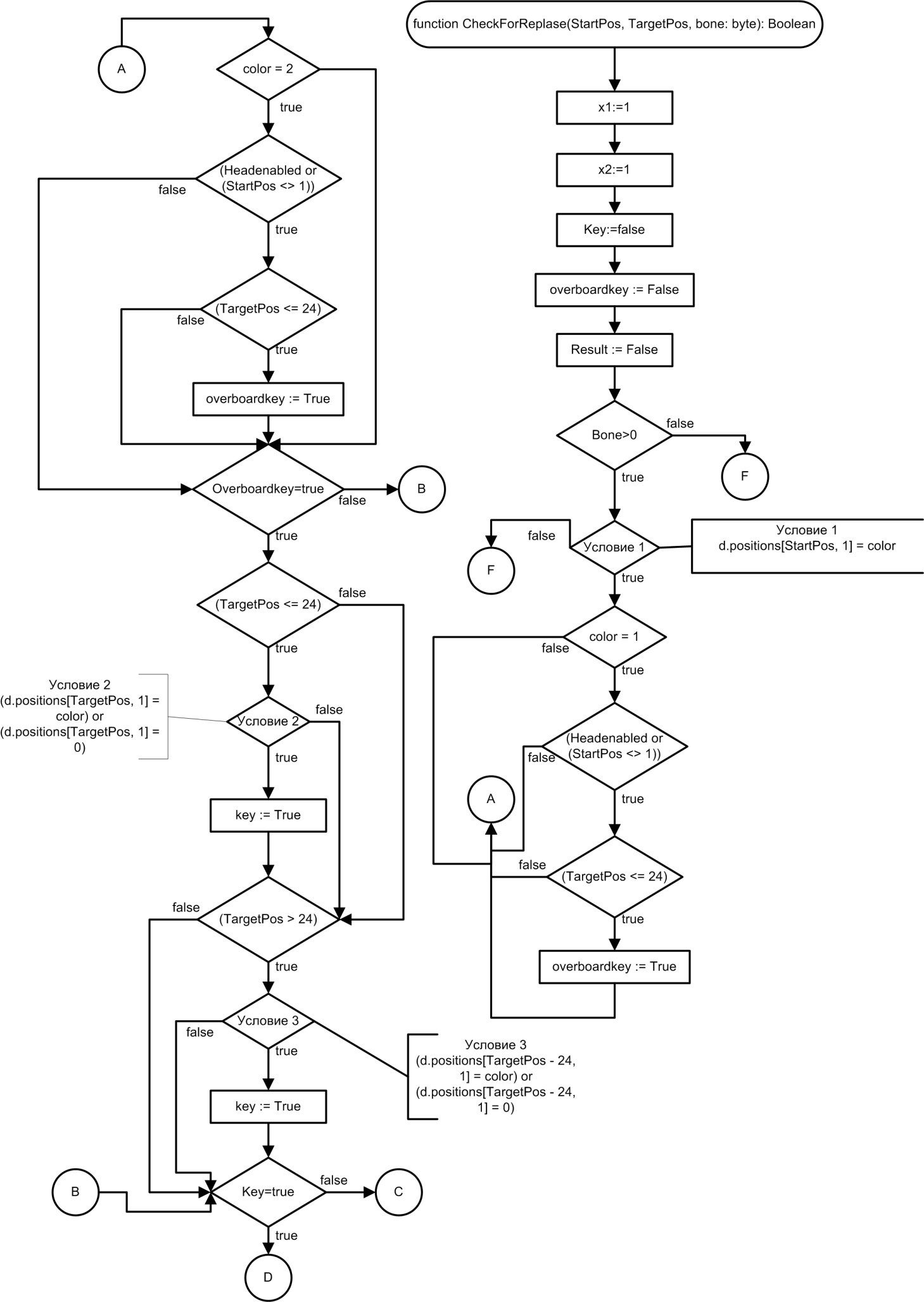
* Чи відповідає стартова і кінцева позиція заданим очкам.
* Чи є на стартовій позиції шашка свого кольору.
* Чи немає на кінцевій позиції шашки супротивника.
* Чи не утворюється ряд з 6 шашок свого кольору, якщо попереду ряду немає шашки супротивника.
* Перевірка логічної правильності переданих параметрів.

Пояснення до алгоритму:

* Алгоритм: Метод-функція CheckToReplase класу Player
* Передані параметри: StartPos, TargetPos, bone: byte;
* StartPos-Початкова позиція (ціле число)
* TargetPos-Кінцева позиція (ціле число)
* Bone-очки зарів (ціле число)
* Повертаємий результат: логічний (БРЕХНЯ, ІСТИНА)

Призначення методу: перевірка можливості ходу шашки за переданими параметрами: стартова позиція, кінцева позиція, очки зарів. Якщо хід можливий, функція повертає ІСТИНА.

Схема алгоритму перевірки ходу зображена на рисунку 3.1.

Рисунок 3.1 - Схема алгоритму перевірки хода

Змн.

Арк.

№ докум.

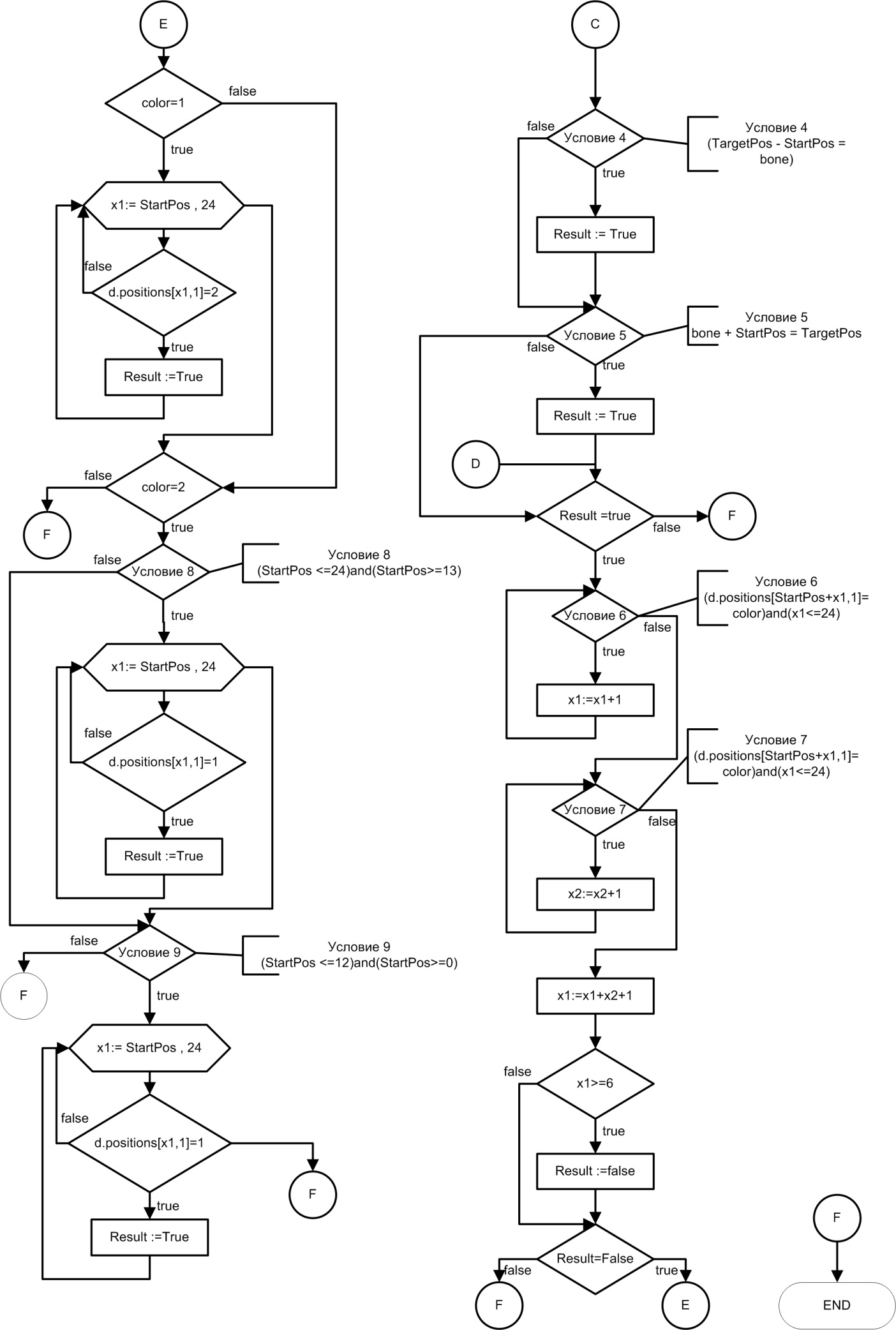
Підпис

Дата

Арк.

9

08-26.КП.010.11.076.ПЗ



Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

10

08-26.КП.010.11.076.ПЗ

Рисунок 3.1 - Схема алгоритма перевірки хода (Продовження)

У розробляємій програмі управління ігровим процесом буде здійснюватися за допомогою миші. Для організації системи управління ігровим процесом необхідно обробити виникше переривання і в залежності від ситуації прийняти рішення. Для обробки дій користувача була створена процедура. Процедура викликається з обробника переривань миші.

Процедура виробляє наступні операції:

* Якщо натискання було скоєно на шашку, показати покажчики ходу (якщо раніше вже була обрана шашка - також приховати покажчики від попередньої шашки).
* Якщо натискання було скоєно на покажчик ходу, зробити хід.
* Якщо викинута остання шашка, видати повідомлення про перемогу (і новому рекорді, якщо він був досягнутий).
* Обробка виняткової ігрової ситуації, яка може виникнути при першому ході.
* Перевірка логічної правильності переданих параметрів.
* Передача ходу комп'ютера, якщо доступних ходів не залишилося.

Пояснення до алгоритму:

* Алгоритм: Метод-процедура CalculateClick класу Human;
* Передані параметри: i, index: byte;
* i-Позиція зображення (1 .. 25) на яку було здійснено натискання (25 позиція-місце, куди викидаються шашки гравця-людини) (ціле число).
* index-Тип зображення, на яке було здійснено натискання (1-шашка, 2-покажчик ходу) (ціле число).

Призначення методу: Подальша обробка натискання на зображення після того як натискання було оброблено обробником миші.

Схема алгоритму системи управління зображена на рисунку 3.2.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

11

08-26.КП.010.11.076.ПЗ

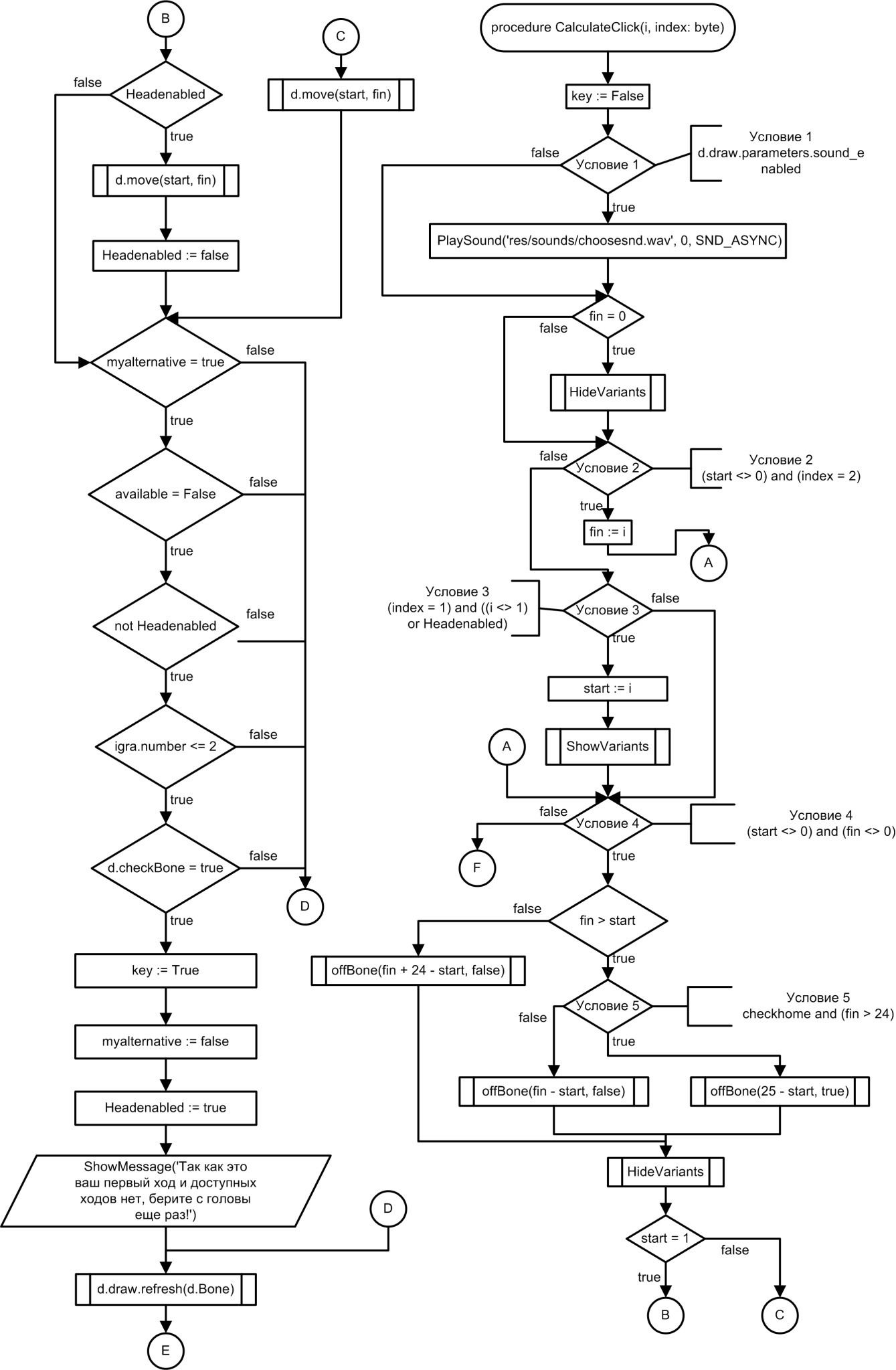


Рисунок 3.2 - Схема алгоритма системи управління

Змн.

Арк.

№ докум.

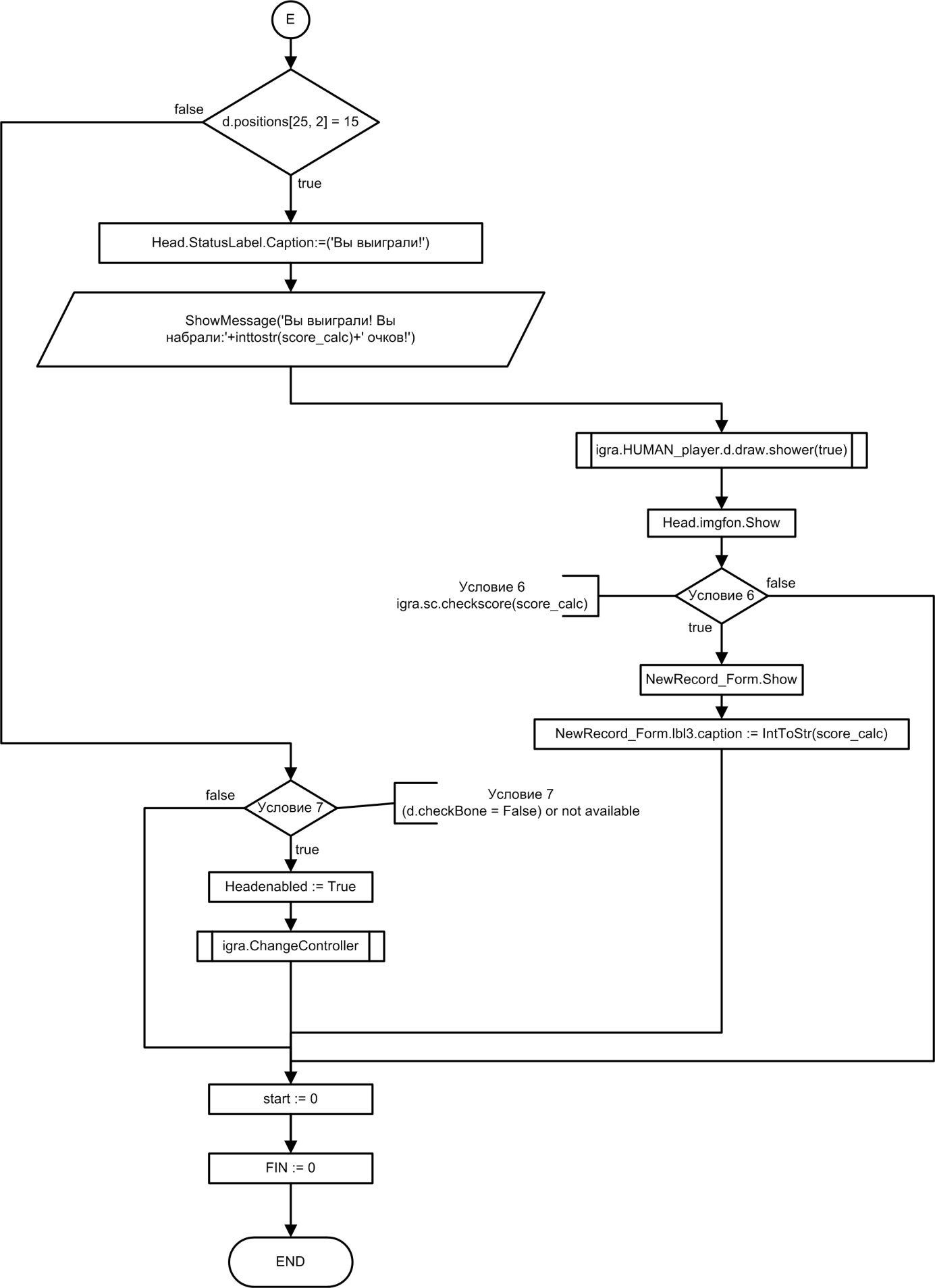
Підпис

Дата

Арк.

12

08-26.КП.010.11.076.ПЗ



Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

13

08-26.КП.010.11.076.ПЗ

Рисунок 3.2 - Схема алгоритма системи управління (продовження)

Згідно концептуальної діаграмі класів, в програмі є КУІП (клас управління ігровим процесом). Головне місце в КУІП займає операція, що відповідає за розподіл права ходу між гравцями. У програмі дана операція реалізована як метод-процедура.

Процедура виробляє наступні операції:

* Організовує кидок кісток.
* Знімає блокування з системи управління, дозволяючи користувачеві ходити.
* Змушує комп'ютер вчинити хід.
* Якщо комп'ютер переміг, видає повідомлення про програш.

Пояснення до алгоритму:

* Алгоритм: Метод-процедура ChangeController класу Game;
* Передані параметри відсутні;

Призначення методу: Передача права ходу між гравцями.

Примітка: в процедурі використовується явна рекурсія.

Схема алгоритму розподілу права ходу на рисунку 3.3

Змн.

Арк.

№ докум.

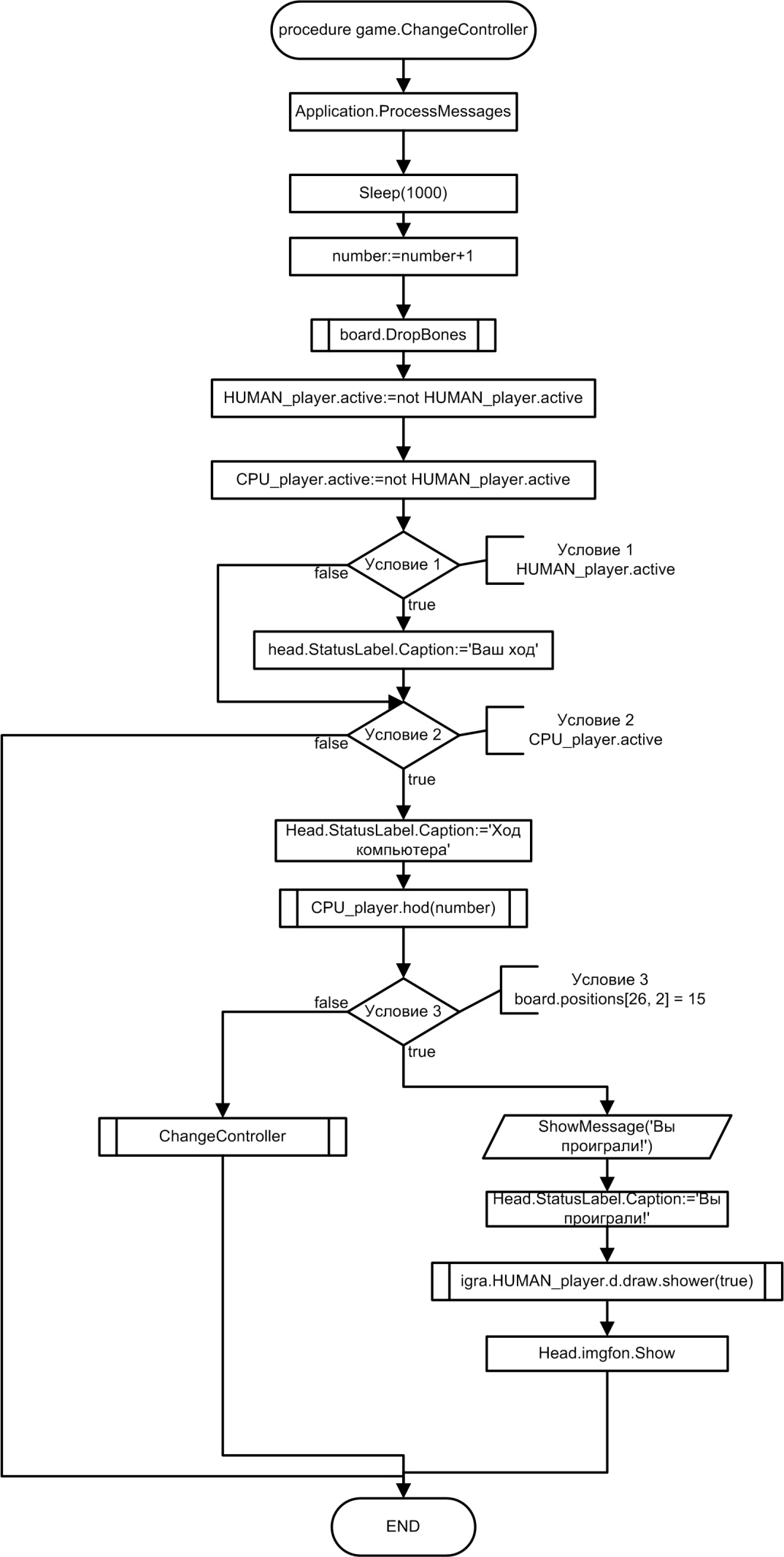
Підпис

Дата

Арк.

14

08-26.КП.010.11.076.ПЗ

Рисунок 3.3 - Схема алгоритма розподілу права ходу4 **РОЗРОБКА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ ІГРОВОЇ СИТУАЦІЇ**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

15

08-26.КП.010.11.076.ПЗ

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

16

08-26.КП.010.11.076.ПЗ

В результаті аналізу предметної області було прийнято рішення про необхідність використовувати об'єкти для реалізації функцій програми. Так як сама програма являє собою програмну реалізацію взаємодії реальних об'єктів, а конкретні функції пов'язані з конкретним об'єктом, то реалізовувати об'єкти та їх взаємодію слід засобами ООП. Основною перевагою об'єктного підходу є зменшення кількості зв'язків між різними частинами програми і зменшення кількості глобальних даних, що дозволить допустити менше помилок при безпосередньому кодуванні і полегшити тестування. При об'єктному підході програма спирається на стандартні компоненти, значить її структура та інтерфейс користувача уніфіковані, що полегшує її розуміння і спрощує використання. Крім того, програма може змінювати поведінку під час виконання: один об'єкт може бути замінений іншим, що дозволяє легко, без зміни коду, адаптувати алгоритм залежно від того, який об'єкт використовується. Для зручності користувача було прийнято рішення використовувати графічний інтерфейс, аналогічний програмам-аналогам, а для його створення раціональніше використовувати об'єктний підхід. Так як інтерфейс найбільш важлива частина програми, для розробки програми був обраний об'єктний підхід, що передбачає використання подійно-орієнтованого програмування.

Враховуючи особливості поставленої задачі, з безлічі середовищ програмування, таких як Eclipse, Microsoft Visual Studio, Visual Basic, Delphi, Qt, було обране середовище Delphi, в якому вже продумано використання об'єктів в області інтерфейсу, що істотно полегшить роботу програміста, тобто Delphi дозволяє візуально створювати інтерфейс програми, використовуючи велику бібліотеку стандартних класів (компонент). Основною перевагою даного середовища, є те, що основні функції по взаємодії з операційною системою вона бере на себе, а обробку повідомлень

операційної системи вона замінює обробкою подій, що суттєво спрощує розробку програмного продукту. Тим більше для реалізації такої програми, яка не потребує глибокої інтеграції з системою, не доцільно використовувати середовище програмування Visual C + + (як більш потужну).

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

17

08-26.КП.010.11.076.ПЗ

Аналіз предметної області дозволив виділити в її складі головний об'єкт-клас управління ігровим процесом (КУІП), що має найбільшу кількість зв'язків з іншими об'єктами. У безпосередніх зв'язках з КУІП знаходяться клас таблиці рекордів, клас налаштувань гри, а також класи-гравці (гравець-людина, гравець-комп'ютер). Клас налаштувань гри забезпечує настроювання відображення графіки. Клас таблиці рекордів реалізує функції таблиці рекордів.

Класи гравців мають зв'язку з класом ігрової дошки, які реалізують аналітичну модель ігрової дошки.

Для відтворення графіки був виділений клас виведення графіки.

Повідомлення об'єктів:

1. змінити настройки
2. передача ходу
3. передача прийнятого рішення про хід
4. виклик графічних функцій
5. перегляд таблиці рекордів або додавання / зміна даних

Рисунок 4.1 - Концептуальна діаграма класів

Ігрок-людина

Ігрок-комп'ютер

Клас вивода графіки

Клас ігрової доски

Клас управління ігровим процесом

Клас налаштувань гри

Клас таблиці рекордів

5

2

1

2

3

3

4

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

18

08-26.КП.010.11.076.ПЗ

Програма за своєю суттю є настільною шашкової грою. Дошка разом з кістками і шашками - структурний елемент. Тому основні об'єкти роботи програми - масиви даних, які визначаються сукупністю критеріїв, що характеризують кожну шашку і кістка; Для зберігання інформації про ігрову дошку був обраний двовимірний цілочисельний масив (див. Таблицю 4.1). Це мінімальний обсяг даних, яким можна однозначно описати ігрову дошку.

Тоді формат масива наступний

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

19

08-26.КП.010.11.076.ПЗ

Таблиця 4.1 - Структура масива дошки

|  |  |  |
| --- | --- | --- |
| Розмірність | 1 | 2 |
| Призначення | Статус ячейки (0..2) | Кількість шашок (0..15) |
| Розмір (байт) | 1 | 1 |

Для переміщення шашок по області дошки доцільно використовувати заздалегідь розрахований масив координат для прискорення роботи програми та спрощення подальшого налагодження. Для зберігання координат був обраний тривимірний цілочисельний масив (див. Таблицю 4.2).

Таблиця 4.2 - Структура масиву дошки

|  |  |  |  |
| --- | --- | --- | --- |
| Розмірність | 1 | 2 | 3 |
| Призначення | Позиція по горизонталі (1..26) | Позиція по вертикалі  (1..16) | Координати  (1-X; 2-Y) |
| Розмір (байт) | 2 | 2 | 2 |

Для зберігання записів таблиці рекордів пропонується використовувати текстовий файл, що складається з послідовності записів наведеного нижче (див. Таблицю 4.3) формату. Такий спосіб зберігання дозволяє швидко зчитувати довільний запис з файлу таблиці рекордів "rec.txt" і перемикання між ними.

Таблиця 4.3 - Структура запису таблиці рекордів

|  |  |  |
| --- | --- | --- |
| Номер поля | 1 | 2 |
| Призначення поля | І'мя игрока | Очки |
| Розмір (байт) | 256 | 4 |

Для зберігання ігрових стилів, необхідне зберігання імен файлів на використовувані зображення. Масив рядків в даному випадку використовувати незручно, так як численних звернень по індексам не вимагається (відсутні циклічні операції), для доступу до рядків краще використовувати записи. Приклад такого запису наведено нижче (див. Таблицю 4.4).

Таблиця 4.4 - Структура запису теми

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Номер поля | 1 | 2 | 3 | 4 | 5 | 6 |
| Признач-ення поля (ім'я файла) | Зображення шашки гравця 1 | Зображення шашки гравця 2 | Підсві-чування ходу | Фон доски | Трику-тник нижній | Трику-тник верхній |
| Розмір (байт) | 256 | 256 | 256 | 256 | 256 | 256 |

Розробку програми доцільно почати з визначення варіантів використання, які визначають користувачів системи та їх взаємодію з системою.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

20

08-26.КП.010.11.076.ПЗ

Дійова особа у програми одна - користувач, який вибирає потрібну йому функцію.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

21

08-26.КП.010.11.076.ПЗ

У розроблювальної програми будуть наступні варіанти використання:

* Ігровий процес
* Перегляд таблиці рекордів
* Зміна ігрових налаштувань
* Перегляд довідкової інформації

Таблиця 4.5 - Варіант використання "Ігровий процес"

|  |  |
| --- | --- |
| Дії виконавця | Відклик програми |
| 1. Користувач запускає програму і натискає на пункт головного меню «Гра» - «Нова гра».  3. Користувач натискає кнопку «ОК».  5. Користувач натискає лівою клавішею миші на шашку свого кольору.    7. Користувач натискає лівою клавішею миші на покажчик ходу. (Покажчик ходу - це зображення у вигляді зірочки або будиночка, використовуване для позначення доступного ходу).  9. У результаті гри користувач встановив новий рекорд.  11. Користувач вводить в поле «Введіть ваше ім'я» довільний рядок і натискає кнопку «Прийняти»  13. Користувач натискає кнопку «Гра» - «Вихід».  15.Користувач натискає кнопку «ОК» | 2. Система уточнює у користувача, чи дійсно він хоче почати нову гру за допомогою діалогового вікна.  4. Система ініціює ігровий процес  6. Система підсвічує доступні ходи цієї шашкою, якщо вони є.  8. Система переміщує обрану користувачем шашку на місце покажчика ходу.  10. Система виводить діалогове вікно з повідомленням про новий рекорд і виводить форму, в якій пропонується ввести ім'я для збереження результату.  12. Система заносить запис про новий рекорд в таблицю рекордів.  14. Система уточнює у користувача, чи дійсно він хоче вийти з гри за допомогою діалогового вікна.  16. Завершення роботи системи. |

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

22

08-26.КП.010.11.076.ПЗ

Альтернатива 1

3.Користувач відмовляється від початку гри.

6.Сістема видає користувачеві повідомлення, що можна зняти з голови ще одну шашку.

Альтернатива 2

8. Система переміщує обрану користувачем шашку на місце покажчика ходу, і передає хід комп'ютеру.

Альтернатива 3

       11. Користувач не ввів ім'я в поле і натиснув «Прийняти».

15. Користувач відмовляється виходити з гри.

Альтернатива 4:

6.Система стирає раніше намальовані покажчики ходу і малює їх у місцях, куди можливий хід.

Альтернатива 5

7.Система видає користувачеві повідомлення про перемогу.

11. Користувач відмовляється від внесення нового рекорду в таблицю рекордів.

Таблиця 4.6 - Варіант використання "Перегляд таблиці рекордів

|  |  |
| --- | --- |
| Дії користувача | Відклик програми |
| 1. Користувач запускає програму і натискає на пункт головного меню «Гра» - «Рекорди».  3.Користувач натискає на кнопку «Очистити таблицю»  5. Користувач натискає кнопку «ОК»  7. Користувач натискає кнопку «Назад» | 2. Система показує користувачеві форму з таблицею рекордів.  4.Система видає діалогове вікно з уточненням, чи дійсно користувач бажає очистити таблицю.  6.Система очищає таблицю рекордів  8. Система закриває форму таблиці рекордів. |

Альтернатива:

5.Користувач відмовляється від очищення таблиці рекордів.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

23

08-26.КП.010.11.076.ПЗ

6. Система повертається в початковий стан.

4.1. Розробка інтерфейсу користувача

Як випливає з ТЗ, що розробляється програма призначена для користувачів всіх вікових категорій, в тому числі що мають слабкий навик роботи з комп'ютером, і до інтерфейсу пред'являються особливі вимоги. Так само програма є грою, отже, інтерфейс повинен бути добре продуманий. Саме тому, розробку доцільно почати з розробки користувальницького інтерфейсу.

Інтерфейс користувача являє собою сукупність програмних і апаратних засобів, що забезпечують взаємодію користувача з комп'ютером. Існує безліч різновидів інтерфейсу: текстові, графічні, голосові, інтерфейси-меню. Так само інтерфейси ділять на: «примітивні», меню, з вільною навігацією та прямого маніпулювання.

 Інтерфейс програми повинен бути досить простим, щоб забезпечити максимально швидку роботу в ньому користувача. Для простоти освоєння програми користувачем, було прийнято рішення використовувати графічний інтерфейс. Огляд інформації на різних ігрових порталах по настільним іграм показав, що у використанні зручніше інтерфейс з мінімально необхідною кількістю елементів управління і не втомлює користувача. Тому було прийнято рішення використовувати інтерфейс-меню. Такий тип графічного інтерфейсу дає можливість користувачу вибирати необхідні дії зі спеціального списку.

 На основі аналізу основних функцій і об'єктної декомпозиції був побудований граф станів інтерфейсу, представлений на рисунку 4.2.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

24

08-26.КП.010.11.076.ПЗ

Головне меню

Початок нової ігри

Таблиця рекордів

Настройки відображенняия

Керівництво користувача

Про програму

1

2

3

4

6

7

Звук вкл./викл.

5

9

100

8

8

8

8

8

Рисунок 4.2 - Граф стану інтерфейса

1. натиснутий пункт меню «Гра-Нова гра»;
2. натиснутий пункт меню «Гра-Рекорди»;
3. натиснутий пункт меню «Гра-Вихід»;
4. натиснуто пункт меню «Налаштування-Налаштування вигляду дошки та шашок»;
5. натиснутий пункт меню «Налаштування-Звук вкл. / викл.»;
6. натиснутий пункт меню «Довідка-Керівництво користувача»;
7. натиснутий пункт меню «Довідка-Про програму»;
8. натиснута кнопка «Назад»

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

25

08-26.КП.010.11.076.ПЗ

1. натиснута кнопка «Очистити таблицю»
2. натиснута кнопка «Застосувати»

4.2. Розробка меню

Для забезпечення швидкої та зручної навігації користувача по програмі основні функції програми доцільно об'єднати в головному меню програми, представленому на рисунку 4.3.

Дослідивши інтерфейс програм-аналогів, функції були розділені на 3 групи:

* Функції гри
* Функції налаштувань
* Довідкові функції

**Главное меню**

**Игра**

**Новая игра**

**Рекорды**

**Выход**

**Настройки**

**Настройки вида доски и шашек**

**Звук вкл./откл.**

**Справка**

**Руководство пользователя**

**Поддержка**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

26

08-47.КП 001.06.222.ПЗ

Рисунок - 4.3 Структура меню

4.3. Розробка форм вводу виводу

Для ведення обміну інформацією між користувачем і програмою в кожному з станів графа діалогу інтерфейсу, необхідно розробити форми введення і виведення цієї інформації. Для реалізації діалогу була обрана таблична форма виведення діалогу.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

27

08-26.КП.010.11.076.ПЗ

 Форми вводу виводу можна розділити на 3 частини: головна форма, форма налаштувань, довідкові форми. Основною формою для роботи з дошкою і шашками буде головна форма. Вона з'являється при запуску програми. Для фонового зображення використаний компонент Image (див. рисунки 4.4, 4.5)

Рисунок 4.4 - Головна форма

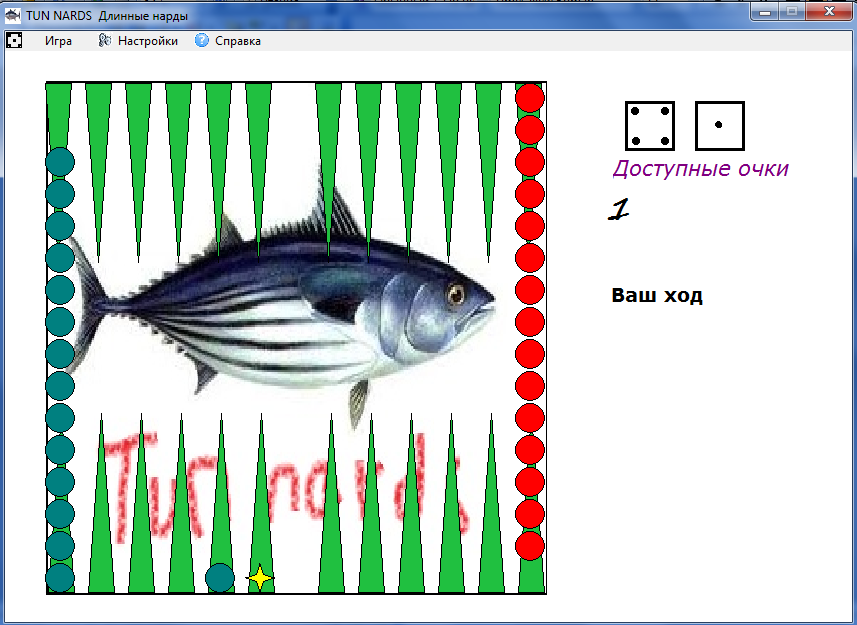


Рисунок 4.5 - Головна форма під час гри

Головна форма призначена для організації доступу до функцій програми через меню. На ній відбувається ігровий процес . Розмір форми був підібраний рівно під розміщення дошки, кісток і написів. Звернення до функцій програми здійснюється через меню, так само як у більшості програм-аналогів.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

28

08-26.КП.010.11.076.ПЗ

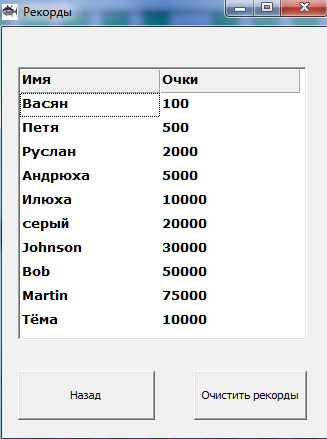


Рисунок 4.6 - Форма таблиці рекордів

Форма таблиці рекордів забезпечує доступ користувача до перегляду 10 кращих результатів гри, а так само дає йому можливість очистити таблицю. На рисунку 4.6 приведена заповнена таблиця рекордів. Для розміщення записів був використаний компонент StringGrid, так як з його допомогою дуже легко додавати або видаляти дані, а так само відсутня необхідність вирівнювати текст-він завжди буде вписаний в свою комірку. Знизу поруч були розташовані кнопки компоненту Button «Назад» і «Очистити рекорди».

Змн.

Арк.

№ докум.

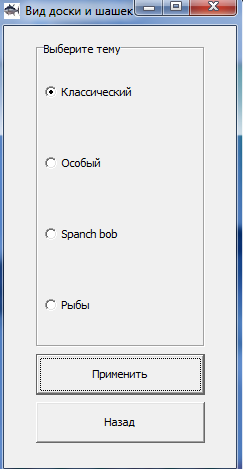
Підпис

Дата

Арк.

29

08-26.КП.010.11.076.ПЗ

******

Змн.

Арк.

№ докум.

Підпис

Дата

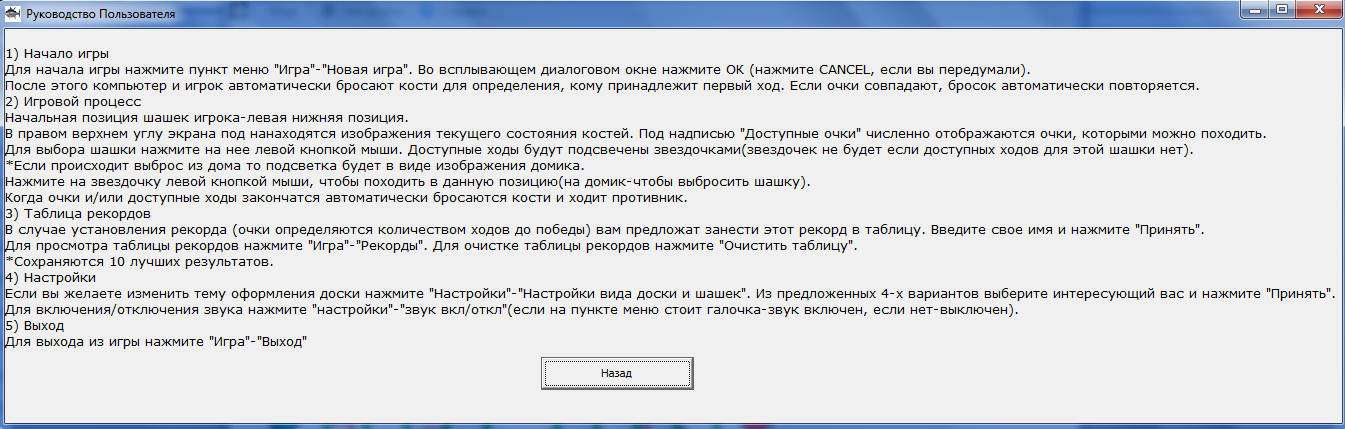
Арк.

30

08-26.КП.010.11.076.ПЗ

Рисунок 4.7 - Форма налаштувань вигляду

На рисунку4.7 приведена форма налаштувань виду ігрової дошки. На формі розташовані радіокнопки для вибору графічного стилю з чотирьох пропонованих. Для цього був використаний компонент RadioGroup - завдяки йому всі записи рівно один під одним, так само за допомогою нього до радіокнопки можна звернутися за індексом-так буде простіше обробити натискання кнопок при виборі. Знизу один під одним були розташовані кнопки компоненту Button «Застосувати» і «Назад», щоб надмірно не розширювати область форми.

Рисунок 4.8 - Керівництво користувача

На рисунку 4.8 наведено форму керівництва користувача. Ця форма повинна вміщати великий обсяг тексту і була зроблена як можна ширше, щоб спростити читання тексту. Для розташування тексту був обраний компонент Label, так як з ним простіше працювати, і відсутнє поле введення, яке не потрібно.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

31

08-26.КП.010.11.076.ПЗ

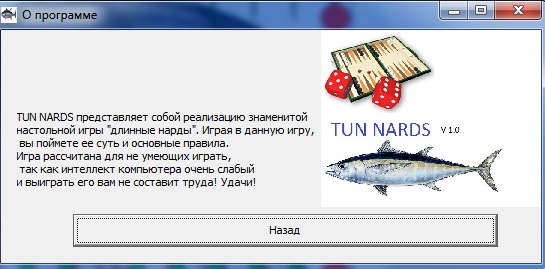


Рисунок 4.9 - Форма підтримки

На рисунку 4.9 приведена форма підтримки. Ця форма створена аналогічно формі для керівництва користувача, так як вона теж використовується лише для показу тексту. На ній так само розташований об'єкт Image для розміщення логотипу гри.

**5 АНАЛІЗ РЕЗУЛЬТАТІВ ІСПИТІВ ПРОГРАМИ**

Тестування - процес виконання програми, метою якого є виявлення помилок. Ніяке тестування не може довести відсутність помилок в програмі, однак проведення аналізу правильності роботи алгоритму є необхідним кроком у створенні якісного програмного продукту.

Згідно з методологією тестування програмного продукту розрізняють два підходи до формування тестових наборів: структурний і функціональний.

Структурний підхід-тестування, при якому відома структура програмного забезпечення («білий ящик»). Цим підходом бажано скористатися для тестування алгоритму видалення використаних очок.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

32

08-26.КП.010.11.076.ПЗ

Функціональний підхід-тестування, при якому не відома структура програмного забезпечення («чорний ящик»). Цей підхід доцільно застосувати для тестування алгоритму ходу.

Оціночне тестування програми на відповідність відповідним вимогам. Необхідно для аналізу роботи системи в цілому.

Тестування алгоритму знаходження перевірки ходу буде здійснений за допомогою розробки маршруту з комбінаторним покриттям умов. Згідно даним критерієм, тест повинен виконувати всі можливі комбінації результатів умов в кожному рішенні і всі оператори виконувалися хоча б один раз. Нижче наведені результати тестування в таблиці 5.1. Тестування проводилося з урахуванням початкового розміщення шашок.

Таблиця 5.1 - Результатати тестування алгоритма видалення використаних очків.

|  |  |  |  |
| --- | --- | --- | --- |
| Стан масиву очок | Видаляємі очки | Очікуваний результат | Отриманий результат |
| 0 0 0 0 | 1 | 0 0 0 0 | 0 0 0 0 |
| 4 4 4 4 | 16 | 0 0 0 0 | 0 0 0 0 |
| 3 3 3 3 | 11 | 0 0 0 0 | 3 3 3 3 (Помилка!) |
| 2 2 2 2 | 2 | 2 2 2 0 | 2 2 2 0 |
| 3 6 0 0 | 9 | 0 0 0 0 | 0 0 0 0 |
| 1 5 0 0 | 5 | 1 0 0 0 | 1 0 0 0 |
| 6 6 6 6 | 15 | 6 0 0 0 | 0 0 0 0 (Помилка!) |
| 6 6 6 6 | 12 | 6 6 0 0 | 6 0 0 0 (Помилка!) |
| 5 7 0 0 | 8 | 0 0 0 0 | 5 7 0 0 (Помилка!) |

В результаті проведеного тестування були виправлені знайдені помилки, а так само поліпшена структура даного алгоритму.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

33

08-26.КП.010.11.076.ПЗ

Тестування ігрового процесу було виконано за допомогою методу формування тестового набору (припущення про помилку) і складанням лог-таблиці гри на перші 12 ходів. Результати наведені в таблиці 5.2

Таблиця 5.2 - Результатати тестування ігрового процеса.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| № хода | Ходить | Випавші кості | Ходи ігрока | | Результат |
| Початкова позиція | Кінцева позиція |
| 1 | Людина | 6-1 | 1  2 | 2  8 | Вірно |
| 2 | Комп'ютер | 5-6 | 13 | 18 | Вірно |
| 3 | Людина | 6-2 | 8 | 16 | Вірно |
| 4 | Комп'ютер | 3-1 | 13 | 17 | Вірно |
| 5 | Людина | 3-5 | 1  16 | 4  21 | Вірно |
| 6  Змн.  Арк.  № докум.  Підпис  Дата  Арк.  34  08-26.КП.010.11.076.ПЗ | Комп'ютер | 6 6 6 6 | 13  17 | 19  23 | Помилка! Використані не всі очки! |
| 7 | Людина | 2 2 2 2 | 1  3  5  7 | 3  5  7  9 | Вірно |
| 8 | Комп'ютер | 6-5 | 13  19 | 19  24 | Вірно |
| 9 | Людина | 1-2 | 1  9 | 3  10 | Вірно |
| 10 | Комп'ютер | 3-3-3-3 | 13  16 | 16  19 | Помилка! Використані не всі очки! |
| 11 | Людина | 4-5 | 21  1 | 1  6 | Помилка! Поява покажчика ходу в місці куди хід заборонений. |
| 12 | Комп'ютер | 1-1-1-1 | 13  14  15  16 | 14  15  16  17 | Вірно |

В результаті проведеного тестування були виправлені знайдені помилки, а так поліпшені функції візуалізації ігрового процесу. В результаті тестування, проведеного на етапі розробки програми, були виправлені знайдені помилки і поліпшені алгоритми ігрового процесу, використовувані методами класів.

Оціночне тестування.

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

35

08-26.КП.010.11.076.ПЗ

Було проведено тестування всіх функцій програми 3-ма користувачами, в результаті якого були виявлені недоліки інтерфейсу, які були усунені.

Було зроблено висновок, що швидкість виконання програми практично не залежиться від різних ігрових ситуацій. Обсягах зайнятої оперативної пам'яті також залишається практично незмінним - близьким 4, 5 МБ.

Програма повністю задовольняє апаратні вимоги згідно ТЗ.

Тестування проводилося на системі: Intel Pentium P6200, 2.806 ГБ ОЗУ.

Працездатність програми перевірялась також на комп'ютерах з операційними системами:

1) Windows XP Professional (2 комп'ютера)

2) Windows Vista (1 комп'ютер)

3) Windows 7 (3 комп'ютери)

На всіх комп'ютерах програма вела себе стабільно.

ВИСНОВКИ

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

36

08-26.КП.010.11.076.ПЗ

В результаті курсової роботи була розроблена ігрова програма «TUN NARDS». Дану програму можна застосовувати в реальному житті, як ігрову програму для підлітків так і для дорослих. Розроблена програма досить швидко аналізує розташування шашок на ігровому полі і видає результат користувачу. Основною перевагою даної програми є те, що комп’ютер є досить сильним в суперництві з людиною. Даний алгоритм дозволяє прораховувати стратегію на декілька ходів наперед.

Оскільки не всі можливі операції для ігрового процесу були охоплені, то в наступних версіях програми можливе впровадження таких функцій:

1) Скасування помилково здійсненого ходу.

2) Збереження / завантаження гри.

3) Вибір рівня складності гри.

4) Анімація руху шашок.