**МІНІСТЕРСТВО ОСВІТИ І НАУКИ**

**Національний університет водного господарства та природокористування**

**Навчально-науковий інститут автоматики, кібернетики і обчислювальної техніки**

**Кафедра прикладної математики**

**КУРСОВА РОБОТА**

**на тему:**

**«Програмна реалізація гри «Арканоїд»**

**Виконав: студент групи**

**ПМ-21 ННІАКОТ**

**Бойко М.В.**

**Рівне – 2014**

Зміст

1. [Вступ 3](#_Toc295764114)
2. Аналіз предметної області……………………………………………………4

2.1 Формування проблеми…………………………………………………...4

2.2 Аналіз предметної області……………………………………………….4

2.3. Постановка задачі………………………………………………………..5

2.4.Вибір мови програмування………………………………………………5

3. Алгоритмічна побудова………………………………………………………..6

3.1. Інтерфейс………………………………………………………………...6

3.2 Основні етапи розробки….……………………………………………...7

3.3 UML діаграми………………………………………….…...…………..8

4. Код програми………………………………………………………………….11

5. Текстовий приклад……………………………………………………………17

6. Інструкція користувача……………………………………………………….21

7.Висновок………………………………………………………………………..22

8.Література……………………………………………………………………...23

# Вступ

Розвиток цивілізації у нашому столітті неодмінно йде у парі з розвитком різноманітних технологій, зокрема комп'ютерних. Сьогоднішні темпи комп'ютеризації перевищують темпи розвитку всіх інших галузей. Без комп'ютерів і комп'ютерних мереж не обходиться сьогодні ні одна середня фірма, не кажучи про великі компанії. Сучасна людина починає взаємодіяти з комп'ютером постійно - на роботі, вдома, у машині й навіть у літаку. Комп'ютери стрімко впроваджуються в людське життя, займаючи своє місце в нашій свідомості. Комп'ютерний ринок постійно наповнюється новими, досконалішими програмами, збільшується швидкість процесорів, об'єм носіїв збереження пам'яті. У цій технологічній боротьбі не останнє місце займає явище, яке виникло разом з комп'ютерами, а саме, комп'ютерні ігри.

Комп'ютерні ігри відразу ж знайшли купу шанувальників. Вони з дитячих років супроводжують підростаюче покоління, викликаючи швидкий розвиток інтелекту, логічне мислення та уяву людини. Комп'ютерний гравець звикає переміщатися з одного віртуального світу в інший, швидко сприймати незнайомі ситуації й адаптуватися до них. У наш час розвинена інтелектуальна гнучкість забезпечить пристосування до нових, несподіваним реаліям. Комп'ютерні ігри виконують, таким чином, функцію соціалізації молоді в постіндустріальному суспільстві.

Розвиток і вдосконалення ігор тісно пов'язане з розвитком комп'ютерного забезпечення та технологій. Зараз багато хто складові частини комп'ютерів розробляються чи не спеціально для ігор. Наприклад, дорогі відеокарти, вартість яких доходить до половини вартості задовільного комп'ютера для роботи в офісі. Всі ігри розробляються з урахуванням останніх новинок комп'ютерної техніки, реагуючи на всі досягнення і все ближче підходячи до реальності зображення і звуку. На сьогоднішній день існують вражають своєю правдоподібністю гри з гарним графічним і звуковим оформленням, майже повністю імітує життя. Є і постійно виникає величезна кількість фірм, які надають все нові і нові ігри вкрай різноманітного характеру.

Серед ігор всіх часів виділяються так звані культові гри. Зазвичай ці ігри набувають мільйони шанувальників, творці таких ігор нерідко стають мільйонерами. Як правило, ігри такого роду дуже повільно старіють - наприклад, сьогодні багато людей із задоволенням грають в ігри десятирічної давності, такі як «Арканоїд».

**Аналіз предметної області, постановка задачі**

**2.1**. **Формулювання проблеми**

Комп’ютерні ігри для людини – це не лише засіб розважитись. Вони допомагають розвинути людську реакцію, інтелект, знання в певних галузях і областях. Гра арканоїд – одна з таких ігор. Рівень, до якого дійде гравець, залежить лише від швидкості його реакції на поточні обставини гри, які динамічно змінюватимуться в сторону збільшення з кожним новим раундом, що також сприятиме збільшенню реакції. Провівши частину вільного часу за грою, можна отримати задоволення від набраних очок і пройдених рівнів. Якщо людина ніколи не працювала з комп’ютером, то гра допоможе набути мінімальних навиків з клавіатурою без необхідності розбиратись з складними правилами великих ігор та програм.

Правила гри такі: на прямокутному ігровому полі створюється бита, блоки та м’яч. М’яч починає рухатися в довільному напрямку на швидкості, яка відповідає рівню гри. Коли м’яч вийде за межі ігрового поля та кількість життів буде рівна 0, гру буде припинено. Коли м’яч «врізається» в ціль, то блок знищується і гра продовжується далі.

**2.2.Аналіз предметної області**

Предметна галузь, що я обрав, являє собою популярну гру «Арканоід», історія якої починається на прикінці 20-ого сторіччя. Гра пройшла довгий шлях розвитку від програми розробленої під MS DOS, до сучасної версії для сервісів Windows Life та Xbox 360 Life та різноманітних варіантів цієї гри створених для Internet браузерів за допомогою технологій Adobe Flash, Microsoft Silverlight та інших.

Існує багато програмних реалізацій цієї гри, а також багато модифікацій цієї гри з моменту випуску першої версії цієї гри. Найпопулярнішою, певно, була реалізація цієї гри для портативної ігрової системи Brick Game.

**2.3.Постановка задачі**

Метою курсового проекту є реалізація гри «Арканоід» на основі рушія Unity3D, що буде виконувати такий список функцій:

а) Функціонування та відображення меню. Переключення між пунктами

б) Створення грального рівня. Відображення блоків на полі.

в) Пересування бити по полю. Функція дозволяє пересувати вліво або вправо биту у грі.

г) Функції кульки. Функція обробки зіткнень з іншими об’єктами, функція пересування по гральному полю з заданним прискоренням, зміна напрямку кулі при зіткнені із межами вікна та іншими об’єктами.

д) Відображення усіх об’єктів на полі. У верхній частині вікна виводиться статистика про стан гри, в нижній частині відображається бита гравця та кулька, яка переміщюється по усьому просторі.

е) Реалізація функцій для ведення статистики. Функцію підрахунку кількості балів гравця, яка обчислює кількість балів, яку отримує він за розбиті блоки. Функція обліку життів, вона відповідає за об’єм життів та якщо вони закінчились, гра теж закінчується.

є) Звукові ефекти. Програвання звуків у грі, коли кулька зіткнеться із одним з блоків.

* 1. **Вибір мови програмування**

Для реалізації гри буде використана мова програмування С#. Дана мова є однією із найбільш простих та зручних мов для реалізації ігрових алгоритмів у середовищі Unity3D. Синтаксис C# близький до С++ і Java. Мова має строгу статичну типізацію, підтримує поліморфізм, перевантаження операторів, вказівники на функції-члени класів, атрибути, події, властивості, винятки, коментарі у форматі XML. Перейнявши багато що від своїх попередників — мов С++, Delphi, Modula і Smalltalk — С#, спираючись на практику їхнього використання, виключає деякі моделі, що зарекомендували себе як проблематичні при розробці програмних систем, наприклад множинне спадкування класів (на відміну від C++).

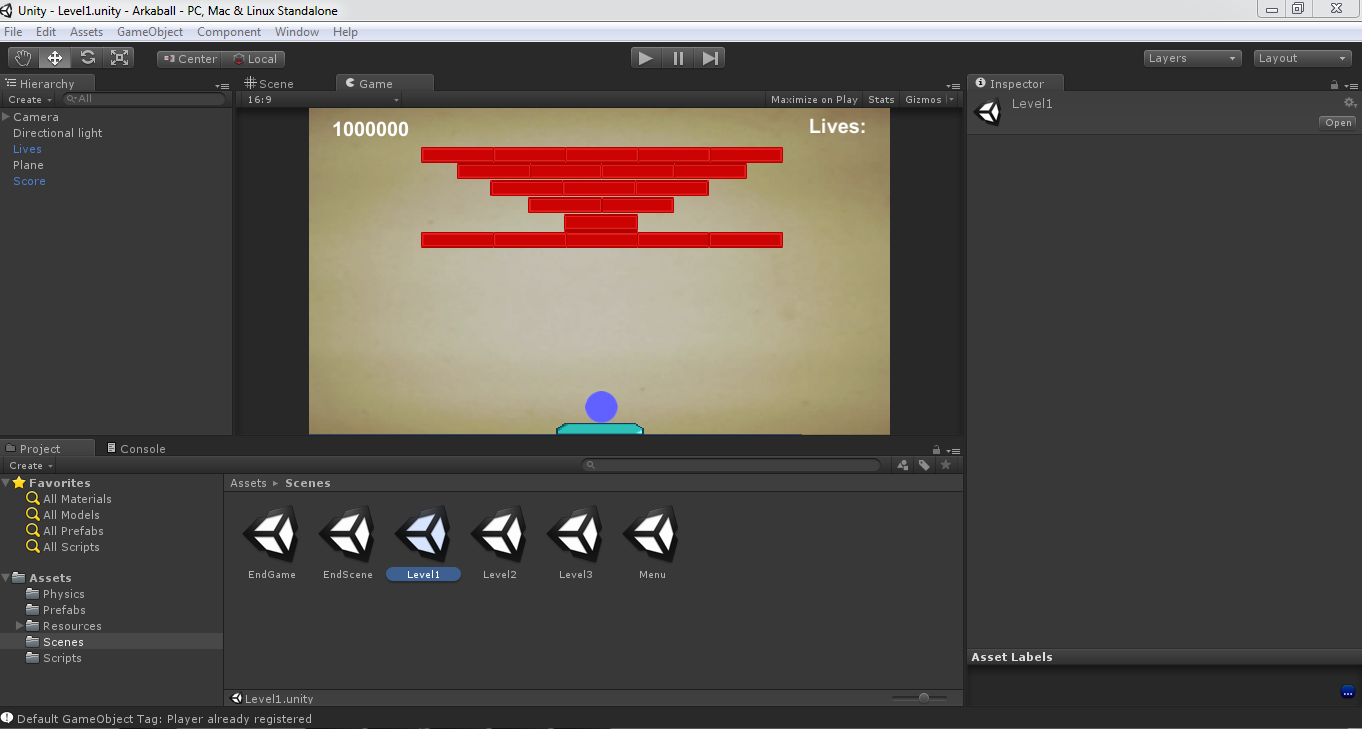
Для реалізації поставлених цілей можливостей рушія Unity3D версії 4.3.3 більше ніж достатньо, тому для ускладнення задачі було вирішено використовувати новітню розробку цього рушія, а саме фізику двохвимірних об’єктів - RigidBody 2D.

**Алгоритмічна побудова**

* 1. **Інтерфейс**

Задана деяка ширина та висота ігрового поля, яка повинна зберігатись та не деформуватись внаслідок зміни розширення екрану. В грі передбачено часткове використання миші(підчас запуску гри та коли потрібно переграти чи закрити її), а передбачена обробка даних з клавіатури.

Інтерфейс програми розроблений на недостатньому рівні, оскільки меню є не повністю насичене, та немає складних систем. При виборі користувачем нової гри, відкривається підменю із вибором доступного рівня. Після вибору рівня завантажуються графічні елементи вибраного рівня і безпосередньо розпочинається гра.



* 1. **Загальний алгоритм**

На етапі основного виробництва виконується величезний обсяг робіт. Спочатку пишеться вихідний код, малюється графіка, такі як спрайт. Розробляються звукові ефекти, а також пишеться музика для гри. Величезний обсяг роботи також припадає на створення та розробку рівнів.

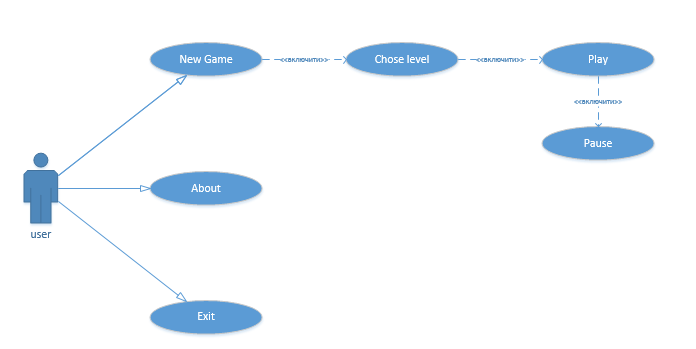
Весь цей час доповнюється та змінюється ігровий дизайн, щоб відобразити поточне бачення гри. Деякі особливості або рівні можуть бути видалені, деякі додані. Художня трактування може еволюціонувати, а процес гри - змінитися. Може з'явитися нова цільова платформа, а також нова цільова аудиторія. Всі ці зміни мають бути задокументовані і більшість з них має з'явитися в дизайн-документі.

З точки зору часу перший рівень гри розробляється довше за всіх інших. Оскільки при створенні рівнів і графіки використовуються інструменти для створення рівнів, потрібні можливості і зміни внутрішніх інструментів. З появою нових можливостей деякі рівні можуть застаріти, тому в перший рівень гри можуть вноситися різні виправлення. Крім того, в силу динамічної природи розробки ігор, дизайнерське бачення першого рівня з часом може змінюватися. Наступні рівні розроблюються значно швидше, так як список можливостей стає більш повним, а бачення гри - більш ясним.

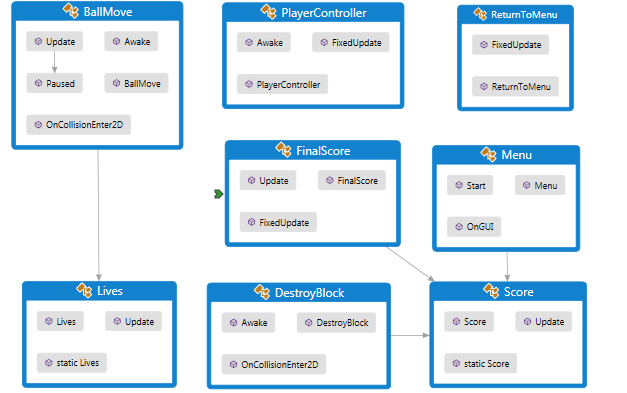
Тести підключаються до гри, коли з'являється щось іграбельне. Це може бути один рівень або підмножина ігор, що може використовуватися в будь-яких розумних межах. На ранньому етапі тестування гри віднімає відносно малу частку часу. У міру наближення розробки до кінця, гра може почати відбирати для тестів весь час - і навіть понаднормово - оскільки тести намагаються охопити і протестувати нові можливості, для яких існують тести регресії.

Сьогодні тестування є життєво важливим для ігор, оскільки, в силу складності більшості з них, одна єдина зміна може призвести до катастрофічних наслідків.

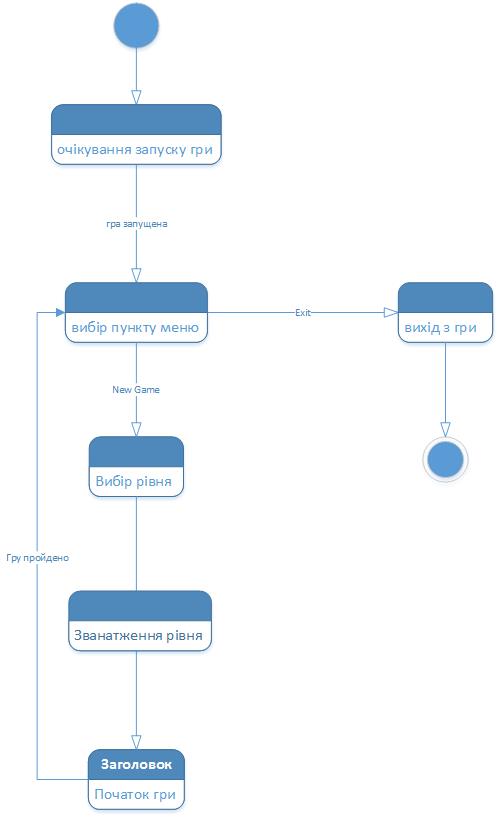
* 1. **UML діаграми**

Варіантів використання

Класів



Станів



**Код програми**

*BallMove.cs*

*using UnityEngine;*

*using System.Collections;*

*public class BallMove : MonoBehaviour {*

*public float speed;*

*public AudioClip pong, explosion;*

*private Rigidbody2D thisRigidbody;*

*private AudioSource thisAudio;*

*private bool paused = false;*

*void Awake () {*

*thisRigidbody = rigidbody2D; //Caching the rigidbody2D*

*thisAudio = audio;*

*thisRigidbody.velocity = Vector2.up\*speed; //Velocity of the ball*

*}*

*void Paused( bool pause)*

*{*

*if(Input.GetKey(KeyCode.Space))*

*{*

*if(!pause)*

*{*

*Time.timeScale = 0;*

*paused=true;*

*}*

*else*

*{*

*Time.timeScale = 1;*

*paused=false;*

*}*

*}*

*}*

*void Update()*

*{*

*Paused (paused);*

*}*

*void OnCollisionEnter2D(Collision2D collision)*

*{*

*switch(collision.gameObject.name) //If collisions...*

*{*

*case "Bot": //Lose and reload the level*

*{*

*Lives.currentLives--;*

*Application.LoadLevel(Application.loadedLevel);*

*if (Lives.currentLives == 0)*

*Application.LoadLevel("EndGame");*

*}*

*break;*

*case "Block": //Sound of explosion of the block*

*thisRigidbody.velocity = thisRigidbody.velocity\*1.05f;*

*thisAudio.PlayOneShot(explosion);*

*break;*

*case "Player": //Sound of bounce and increment the veolcity*

*thisAudio.PlayOneShot(pong);*

*break;*

*}*

*}*

*}*

*PlayerController.cs*

*using UnityEngine;*

*using System.Collections;*

*public class PlayerController : MonoBehaviour {*

*public float speed; //Movement speed*

*private Rigidbody2D thisRigidBody;*

*void Awake () {*

*thisRigidBody = rigidbody2D; //Caching the rigidbody2D*

*}*

*void FixedUpdate () {*

*thisRigidBody.velocity = new Vector2(Input.GetAxis("Horizontal")\*speed, 0); //Move the paddle*

*}*

*}*

*Menu.cs*

*using UnityEngine;*

*using System.Collections;*

*public class Menu : MonoBehaviour {*

*public int window;*

*public GUIStyle guiStyle;*

*void Start () {*

*window = 1;*

*Score.curScore = 0;*

*}*

*void OnGUI () {*

*GUI.BeginGroup (new Rect (Screen.width / 2 - 100, Screen.height / 2 - 100, 200, 200));*

*if(window == 1)*

*{*

*if(GUI.Button (new Rect (10,30,180,30), "Play"))*

*{*

*window = 2;*

*}*

*if(GUI.Button (new Rect (10,70,180,30), "About"))*

*{*

*window = 3;*

*}*

*if(GUI.Button (new Rect (10,110,180,30), "Exit"))*

*{*

*window = 4;*

*}*

*}*

*if(window == 2)*

*{*

*GUI.Label(new Rect(50, 10, 180, 30), "Choose level");*

*if(GUI.Button (new Rect (10,40,180,30), "Level 1"))*

*{*

*Debug.Log("Уровень 1 загружен");*

*Application.LoadLevel("Level1");*

*}*

*if(GUI.Button (new Rect (10,80,180,30), "Level 2"))*

*{*

*Debug.Log("Уровень 2 загружен");*

*Application.LoadLevel("Level2");*

*}*

*if(GUI.Button (new Rect (10,120,180,30), "Level 3"))*

*{*

*Debug.Log("Уровень 3 загружен");*

*Application.LoadLevel("Level3");*

*}*

*if(GUI.Button (new Rect (10,160,180,30), "Back"))*

*{*

*window = 1;*

*}*

*}*

*if(window == 3)*

*{*

*GUI.Label(new Rect(50, 10, 180, 30), "About");*

*GUI.Label(new Rect( 10, 40, 180, 40), "About game and help");*

*if(GUI.Button (new Rect (10,90,180,30), "Back"))*

*{*

*window = 1;*

*}*

*}*

*if(window == 4)*

*{*

*GUI.Label(new Rect(50, 10, 180, 30), "Do you want to leave us?");*

*if(GUI.Button (new Rect (10,40,180,30), "Yes"))*

*{*

*Application.Quit();*

*}*

*if(GUI.Button (new Rect (10,80,180,30), "No"))*

*{*

*window = 1;*

*}*

*}*

*GUI.EndGroup ();*

*}*

*}*

*DestroyBlock.cs*

*using UnityEngine;*

*using System.Collections;*

*public class DestroyBlock : MonoBehaviour {*

*private GameObject thisGameObject;*

*private Transform thisTransform;*

*private int curLevel;*

*void Awake() { //Caching the objects*

*thisGameObject = gameObject;*

*thisTransform = transform;*

*}*

*void OnCollisionEnter2D(Collision2D collision) {*

*curLevel = Application.loadedLevel;*

*if(thisTransform.parent.childCount == 1) //If the last block is broken change the level*

*{*

*Application.LoadLevel(curLevel+1);*

*} else { //Else destroy this block*

*Score.curScore += 10;*

*Destroy(thisGameObject);*

*}*

*}*

*}*

*FinalScore.cs*

*using UnityEngine;*

*using System.Collections;*

*public class FinalScore : MonoBehaviour {*

*/// Update is called once per frame*

*void Update () {*

*guiText.text = "Your final score is: " + Score.curScore.ToString();*

*}*

*void FixedUpdate () {*

*if(Input.GetKey(KeyCode.Escape)) //If escape key is pressed...*

*{*

*Application.LoadLevel("Menu"); //Load the menu*

*}*

*}*

*}*

*Lives.cs*

*using UnityEngine;*

*using System.Collections;*

*public class Lives : MonoBehaviour {*

*public static int currentLives = 3;*

*// Update is called once per frame*

*void Update () {*

*guiText.text = "Lives: " + currentLives.ToString();*

*}*

*}*

*ReturnToMenu.cs*

*using UnityEngine;*

*using System.Collections;*

*public class ReturnToMenu : MonoBehaviour {*

*void FixedUpdate () {*

*if(Input.GetKey(KeyCode.Escape)) //If escape key is pressed...*

*{*

*Application.LoadLevel("Menu"); //Load the menu*

*}*

*}*

*}*

*Score.cs*

*using UnityEngine;*

*using System.Collections;*

*public class Score : MonoBehaviour {*

*public static int curScore = 0;*

*// Update is called once per frame*

*void Update () {*

*guiText.text = curScore.ToString();*

*}*

*}*

**Тестовий приклад**

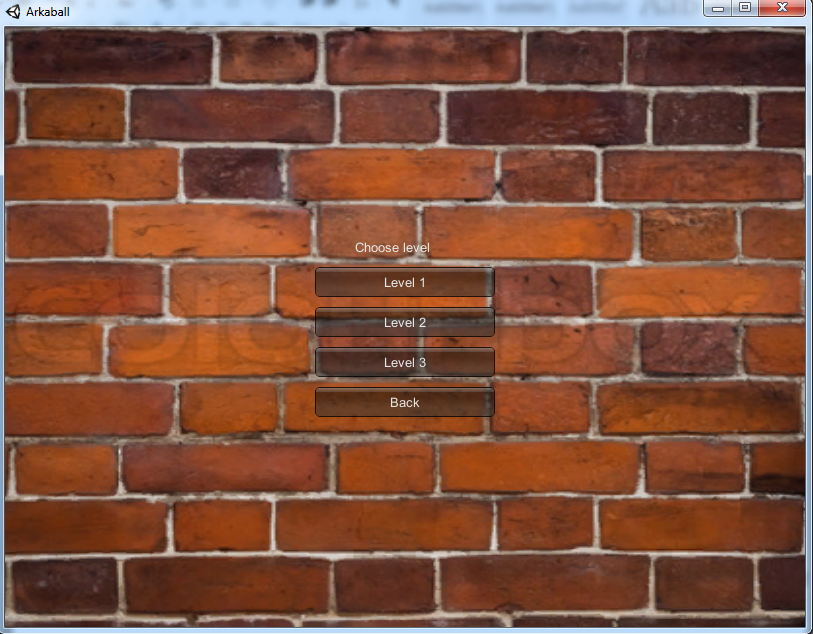
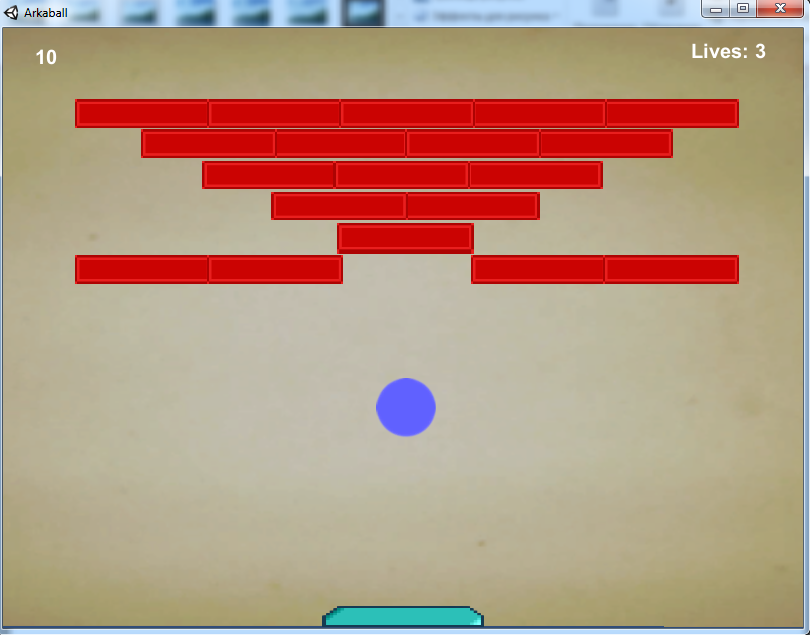
При запуску програми з’являється головне вікно із пунктами меню, де ми можемо, відповідно, розпочати гру, прочитати інструкцію або вийти.

Початковий вигляд при запуску програм



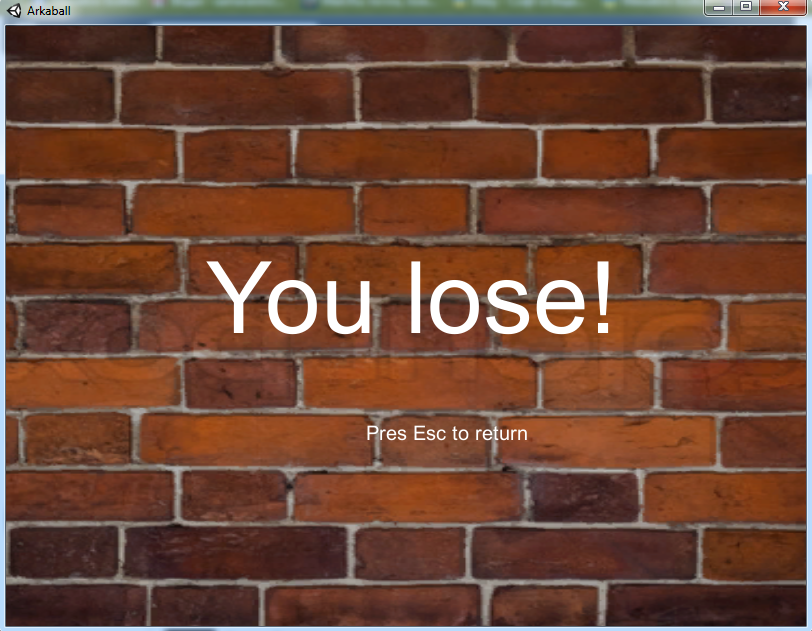
Далі для того щоб розпочати гру , необхідно нажати лівою кнопкою миші по Play та вибрати рівень, тоді на екрані з’являються відповідні компоненти – бита, блоки та м’яч.

Початковий вигляд після запуску гри



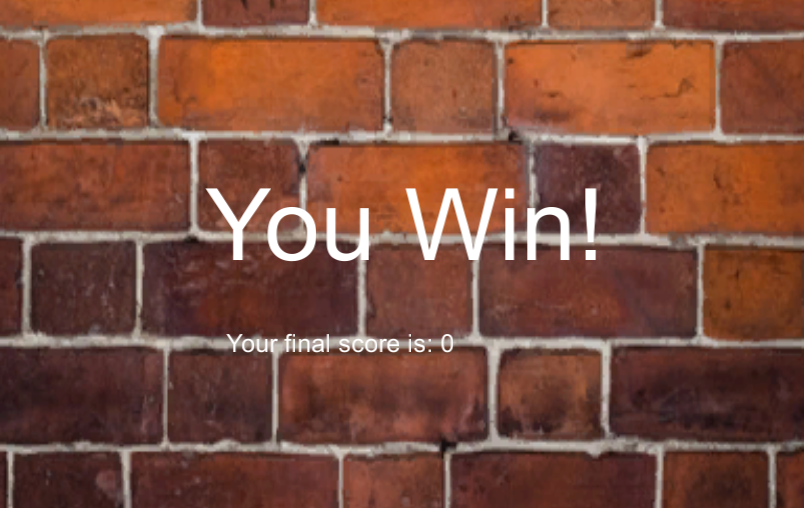
Гра буде продовжуватись до тих пір доки не будуть пройдені всі рівні або не закінчаться життя. Якщо кількість життів стане рівна 0, то гравцю зараховується програш і для продовження буде потрібно розпочати нову гру.

Життя закінчились (програш)



Якщо гравцеві вдалося знищити всі блоки на усіх рівнях і зберегти хоча б 1 життя, то він переходить до екрану привітання переможців, де дізнається свій фінальний результат.

Вітаємо, ви перемогли!



**Інструкція користувача**

Для проходження гри необхідно використовувати стрілки:

1. - рухає биту вправо.
2. - рухає биту вліво
3. Space – пауза у грі.
4. Esc – вихід у меню при програші.

Розпочати гру Play – Level 1.

Для закриття або виходу використовують закриття форми натиснувши клік «закрити» на формі , або в меню вибрати пункт Exit (вихід).

**Висновок**

В ході курсової роботи було проаналізовано та розроблено принципи побудови комп’ютерних ігор із використанням рушія Unity3D, були виконані завдання поставлені у постановці задачі з використанням чималої кількості функцій .

Цей проект можна доопрацьовувати, оскільки модель програми дозволяє легко та зручно її модернізовувати: додати нові команди; зробити сприйнятливий для користувача інтерфейс.

Отже, враховуючи аналіз предметної області і проект програми, можна зробити висновок, що розроблена гра „Арканоїд” розрахована на широке коло користувачів завдяки своїй простоті.

**Література**

Підручники:

1. Шилдт Г. «C# 4.0 Полное руководство»
2. Е. А. Никулин «Компьютерная геометрия и алгоритмы машинной графики»
3. Рихтер Дж. «Программирование на платформе Microsoft .NET Framework 3»

Джерела:

1. http://msdn.microsoft.com/
2. http://mvp.support.microsoft.com/
3. http://www.compulenta.ru/
4. <http://www.intuit.ru/>
5. www.unity3d.ru/