**Технически университет- София**

**Факултет по компютърни системи и технологии**

**Курсова работа по „Програмиране за мобилни устройства“**

Тема:

Създаване на Android приложение

Изготвил: Проверяващ:

Даниел Тошков Велков Ас. Н. Николов

Фак.№ 121216201 гр.46

* Идея на приложението:

Приложението представлява игра от тип безкраен бегач (endless runner) в фантастична обстановка като главния играч се опитва да избегне врагове и препятствия, увеличавайки точките си със времето. Тя използва изображения и листове от кадри на различните обекти в играта, които се обновяват всяка секунда, за да имитират движение. Приложението също включва главно меню и опция за запазване на резултат на играч, като всички резултати се виждат след всеки край на играта. Самата програма е направена чрез фреймуърка Monogame, който е базиран на спряната от Microsoft, XNA среда за разработка на програми със контролирано време за изпълнение като видео игри и други системи. Monogame е основан с целта да се прехвърлят прости игри разработени на Windows чрез XNA на преносими устройства като Windows Phone, Android и IOS. Първоначално е поддържал разработка само за продукти на Apple, но след като проекта става с отворен код се добавят много други мобилни устройства, както и телевизори и конзоли. API-то развива голяма популярност сред независими разработчици на игри и приложения.

* Реализация:

Проектът съдържа две папки- едната е самата игра и всички ресурси свързани с нея наречена Shared, а другата е Android. Мonogame позволява програмиране на междуплатформен софтуер, който е много лесен за имплементиране. Android папката съдържа единствено изгледа, във който ще се намира самата игра в Android приложението и я зарежда със съответните настройки. Същата папка може да се добави за Windows, Linux, IPhone като функции специфични за дадените платформи като touch screen или GamePad, може да се обработят чрез специални If-statements, които изпълняват тялото си според даденото устройство. Те се записват в Shared папката, за да позволят тази cross platform функционалност. В папката Android, се намира класът Activity1, който зарежда приложението със необходимите настройки, като име на приложението, икона, начален екран, ориентация и др. Activity1 наследява Microsoft.Xna.Framework.AndroidGameActivity, което показва, че в фреймуърка може да използваме всички библиотеки на XNA, защото Monogame реално го надгражда. В този клас се задава изгледа на приложението да е на класът от тип Game1, който е началния и основния клас от където играта започва безкраен цикъл на рисуване и обновяване. Също е настроено да не се изгасва екрана след време чрез задаване на флагът: WindowManagerFlags.KeepScreenOn.

Добавени са и класовете свързани с базата данни, като моделът Score, който служи за създаване на таблицата с резултати, която ще попълваме и класът Constants във който е запазен пътят до базата данни, който ще се настройва при стартиране на приложението. От класът с базата данни използваме само два метода: saveScore(), който запазва резултат на играч при край на играта и GetAllScores(), който връща най-големите 5 резултата досега.

Във папка Shared е програмирана самата игра. Класът Game1 e главния за играта и служи като началната й точка, също като main(). Той съдържа основните методи Draw() и Update(), които може да се настроят колко пъти да се извикват в секунда. По подразбиране те се викат 60 пъти в секундата и така е настроено в приложението. Променливата graphics от типа GraphicsDeviceManager служи за използване на графичното устройство налично на дадения уред и чрез нея могат да се задават различни настройки. Тя е използвана за настройване на изгледа да е на цял екран. Конструкторът на Game1() инициализира променливата както и мястото където ще се намират всички изображения и графики свързани с играта чрез Content.RootDirectory = "Content";

За да се добавят във тази папка ресурсите трябва да се преобразуват в удобен за Monogame тип файл, който става чрез Content Pipeline инструмента добавен при инсталиране на Nuget пакета. Така чрез него се добавят изображения, ефекти, шрифтове и след build() те се преобразуват в .xnb файлове. В конструкторът на Game1 също се добавя и обработчик на събития при промяна на екрана, за да не излезе от поставените ограничения на 1920х1080 резолюция на приложението. Ако е по-малко или по-голямо, всичко което се изрисува на екрана се мащабрира според зададената резолюция, така че да запълни екрана изцяло.

Освен споменатите досега методи съществуват и Initialise() и LoadContent() ,които се извикват при започване на приложението и главно се занимават с зареждане на графиките и за инициализирането на променливатa SpriteBatch, която по- късно ще ни служи за обединяване на всички изрисувани графики в едно цяло.

За играта са създадени моделите Player, Enemy, BackgroundSprite, които представляват съответно играча, враговете и графиките които стоят на фон. Player заедно с Enemy имплементират методите на абстрактният клас Sprite. В него се съдържа използваната texture, позиция, големина и скорост, като големината обикновено е размера на изображението, което сме добавили. Класът може да използва статични изображения, но е главно предназначен за кадри от анимация сложени в една картина.

useful skele.png

* Кадри от анимация на скелет

За да се повтаря дадено действие се използва параметърът gameTime, който е от тип TimeSpan, показващ изминалото време, но само за самото приложение и тактовете на процесора на устройството, защото при всяко, скоростта на процесора е различна. Тъй като кадрите са подредени един до друг, цялото изображение се разделя на отделни кадри след като разберем тяхната дължина и като зададем някакво време за превключване, ги вкарваме в безкраен цикъл на обновяване започвайки от първият кадър, ако стигнем края на изображението. Изрисуването се използва по един и същ начин, като задължително трябва да се използва споменатата от преди spriteBatch променлива. Тя съдържа Draw() метода, който реално изрисува изображения на екрана. Параметрите, които приема са texture и позиция, както и коя част от изображението точно се използва, защото се работят с множество от кадри и не се използва цялото изображение, а само кадри от него.

Всеки обект от тип Sprite, може да дефинира по свой начин колко голяма е неговата зона за колизия в която може да се каже, че се е докоснал с друг Sprite. Това става чрез абстрактният метод collisionRect(), който всеки трябва да имплементира по свой начин.

Обектът от тип играч си има свой дефинирани методи за Draw и Update, както и за LoadContent. Те правят така, че различни анимации да се изпълнят от класа Animation, според това какво прави играчът. Съществува анимация за навеждане и скачане, като те се изпълняват съответно при натискане или плъзгане по екрана. Така се доставя възможност за избягване на препятствията. Изрисуването на различните анимации се дефинира в предефинирания Draw метод. За пускането на дадена анимация и цялата обработка се използва допълнителен клас AnimationPlayer. Също е предефиниран правоъгълника, който определя размера, който играчът заема.

Враговете в играта си имат свой клас Enemy. В него се дефинира тяхното място или позиция на екрана като се появят, както и как се движат.Различните видове са представени със потребителски дефинирана променлива от изброими типове EnemyTypes. След като главния играч ги избегне те изчезват от екрана и не се изрисуват повече. Тяхното поведение и движение използва статичен клас Movement в който са дефинирани вълнообразни движения.

Класът SpriteManager служи като отделен модул, който работи заедно със Game1, за да се създаде по- слаба зависимост и да се позволи преизползване на кодът, ако е необходимо. Той наследява DrawableGameComponent, и в него е дефинирана цялата игра, но това не е задължително. Възможно е различните нива да са различни компоненти, но тъй като играта е малка всичко е добавено е един DrawableGameComponent. В него са дефинирани и заредени играчът, враговете, изображенията на заден фон, както и неща като време за показване на препятствие, менюта и др. Менютата се зареждат според статична променлива GameState , която показва в какво състояние се намира приложението в момента – дали е по време на игра, или при пускане или затваряне. Класът SpriteManager се пуска от Game1, затова неговите методи на изрисуване и обновяване са извиквани по същият начин.

За да се обработи добавянето на резултат при край на играта са добавени методите NewKeyboard(), ShowKeyboard() които работят асинхронно за да се изчака записване на даденото име на играч. След като се добави, играта продължава нормално и потребителя може да се сравни с най-големите 5 резултата. Тези резултати разбира се, работят с локалната база данни записана при инсталирането. В главното меню също може да се разглеждат.

* Сорс код