

Содержание

[Введение 5](#_Toc59181419)

[1 Анализ предметной области 7](#_Toc59181420)

[1.1 Анализ литературных источников 7](#_Toc59181421)

[1.2 Анализ существующих аналогов 7](#_Toc59181422)

[1.3 Постановка задачи 13](#_Toc59181423)

[2 Разработка программного средства 14](#_Toc59181424)

[2.1 Структура программного средства 14](#_Toc59181425)

[2.2 Интерфейс программного средства 14](#_Toc59181426)

[2.3 Оптимизация графики 19](#_Toc59181427)

[3 Руководство по использованию программного средства 22](#_Toc59181428)

[Заключение 25](#_Toc59181429)

[Список использованной литературы 26](#_Toc59181430)

[Приложение А 27](#_Toc59181431)

Введение

История человечества представляет собой множества периодов, каждый из которых характеризуется способами производства, технологическим прогрессом, подходами и отношением к образованию.

Относительно недавно стало возможным смело заявить, что наше общество перешло от индустриально к постиндустриальному, данный переход характеризуется тем, что индустриальное производство уступило лидирующую позицию бурно развивающимся информационным технологиям. Появление на рынке новой отрасли не является единственным критерием постиндустриального общества, усложнение производственных процессов привело к тому, что современный человек начинает тратить всё больше времени на образование.

К сожаление современные модели образования из-за бюрократических нюансов не успевают за бурно развивающимся обществом, поэтому в наше время начали пользоваться популярностью альтернативные способы образования: курсы, самообразование, образовательные порталы и ресурсы. Естественно, что современного человека интересуют и современные темы, такие как программирование, аналитика, маркетинг, бизнес. Однако у всех этих тем есть общая часть, а именно они все подразумевают под собой активное взаимодействие с людьми как из своих регионов, так и из других стран. И в данном аспекте «бутылочным горлышком» становится языковой барьер, связанный с тем, что в государственных учреждениях образования изучению иностранных языков не уделяют особо внимания, а учащиеся в раннем возрасте не понимают их значимость.

Все эти факторы привели к появлению большого количества сайтов, учебников и мобильных приложения для изучения иностранных языков, с большей части английского, так как является официальным языком для международного общения и его знание либо обязательно, либо является огромным плюсом в современных отраслях.

Наиболее популярными в области изучения английского языка являются сайты и мобильны приложения, которые не отнимают много времени и позволяют быстро повторить необходимые темы или слова в свободное время.

В настоящее время появилось множество средств для разработки сайтов любого типа и специализации и данное направление активно развивается и по сей день, с точки зрения разработчиков и менеджеров проекта особый интерес представляют технологии, позволяющие ускорить разработку и снизить потребление ресурсов.

Относительно недавно появилось изящный язык программирования «Kotlin» от компании JetBrains, он уже успел получить признание среди мобильных разработчиков, однако его создатели не остановились на достигнутом и продолжают активно развивать данный язык, адаптируя его под разные платформы, технологии, которые представляют собой интересные решение, но не все их решения пользуются огромной популярностью и узнаваемостью у разработчиков.

Наиболее интересным решением для создание веб-ориентированных приложений является фреймворк Ktor, главными особенностями которого являются его асинхронность, возможность использовать функционал популярных веб-серверов, таких как Apache, Netty и других, и язык программирования Kotlin. Это всё, что касается разработки серверной части сайта, однако это не все, еще одним важной и интересной особенностью языка Kotlin является его возможность интерпретироваться в язык программирования JavaScript, сохраняя свои преимущества, такие как статическую типизацию и понятный синтаксис.

# Анализ предметной области

## Анализ литературных источников

Фáйтинг (от [англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) Fighting — бой, драка, поединок) — жанр [компьютерных игр](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0), имитирующих [рукопашный бой](https://ru.wikipedia.org/wiki/%D0%A0%D1%83%D0%BA%D0%BE%D0%BF%D0%B0%D1%88%D0%BD%D1%8B%D0%B9_%D0%B1%D0%BE%D0%B9) малого числа персонажей в пределах ограниченного пространства, называемого [ареной](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B5%D0%BD%D0%B0). [4]

Файтинги близки к играм жанра «[Избей их всех](https://ru.wikipedia.org/wiki/%D0%98%D0%B7%D0%B1%D0%B5%D0%B9_%D0%B8%D1%85_%D0%B2%D1%81%D0%B5%D1%85)», однако между ними существуют различия. Так, в большинстве файтингов игроку не требуется перемещаться по длинному [уровню](https://ru.wikipedia.org/wiki/%D0%A3%D1%80%D0%BE%D0%B2%D0%B5%D0%BD%D1%8C_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B5_%D0%B8%D0%B3%D1%80%D1%8B)) и нельзя выйти за границы арены, а бой состоит из нечётного числа отдельных раундов и не является непрерывным. Менее значительными и необязательно присутствующими признаками жанра являются использование многочисленных шкал для изображения жизненно важных показателей персонажей и прорисовка бойцов на арене в профиль.

Важной особенностью файтингов является их нацеленность на соревнование, а не на сотрудничество игроков, что делает игры этого жанра подходящими для [киберспортивных](https://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%B1%D0%B5%D1%80%D1%81%D0%BF%D0%BE%D1%80%D1%82) чемпионатов. Обычно файтинги предоставляют игроку возможность вести бой в режиме «один на один» против компьютерного противника или другого игрока, реже — позволяют сражаться одновременно трём или четырём противникам на одной арене.

Главными атрибутами хорошего файтинга являются:

* визуальное составляющее игры;
* баланс и механика.

При разработке игры большое количество внимание уделяется на происходящее на экране. Это одна из «фишек» данного жанра. Разработчики тщательно продумывают внешний вид, с целью привлечения внимания широкой публики. Порой создают отдельные характеры персонажей и их особые взаимодействия и диалоги между собой, чтобы игрок мог максимально погрузиться в игру и получать от этого удовольствие

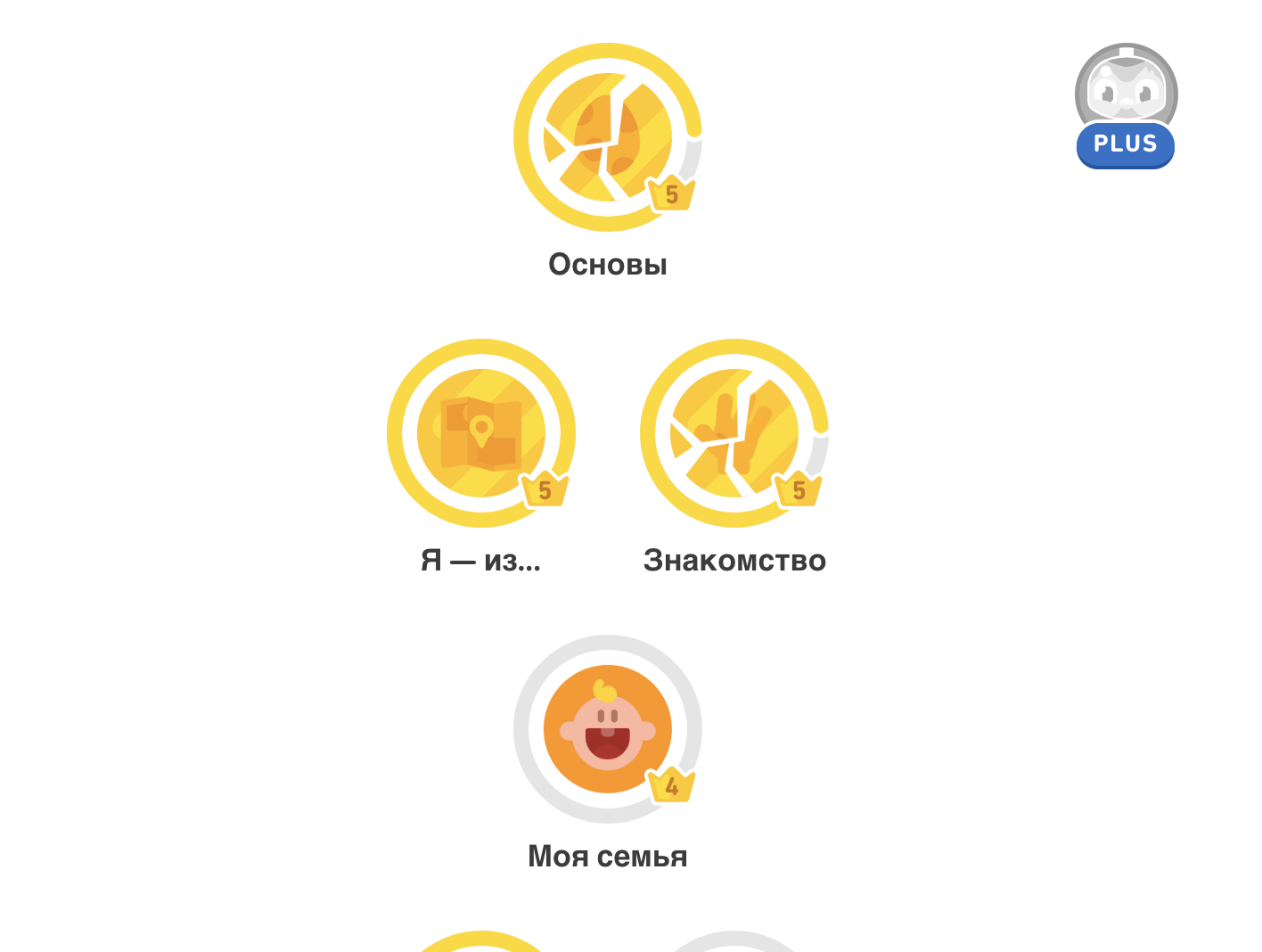
При разработке образов делают акцент на внешнем виде (волосы, одежда и ее цвет), размерах самого персонажа (крупный или небольшой), стойка и фирменные удары.

## Анализ существующих аналогов

Приложений для изучения иностранных языков в наше время стало настолько много, что их можно разделить на категории: изучение грамматики, изучение по темам, улучшения восприятия иностранных слов на слух, заучивание иностранных слов.

Самым популярным ресурсом для изучения иностранных языков является приложение Duolingo, оно предоставляет широкий функционал от изучения разговорных ситуаций и правил до прослушивания историй. Так же данное приложение доступно на всех популярных платформах: веб-приложения, android, iOS.

Данное приложение предоставляет фиксированный список тем, включающих различные этапы и разбитые на категории по уровню сложности. Ниже на рисунке представлена часть интерфейса веб-приложения Duolingo, а именно список тем, который доступы для изучения.

Рисунок 1.1 – Темы Duolingo

Данное приложение пользуется большой популярностью за счет приятного дизайна, узнаваемого символа и своей подачи, как приложения для быстрого изучения языка, не требующего много времен.

Такого рода приложениям удается легко вовлечь пользователей простотой подачи и грамотной рекламной компании. Однако основной задачей является удержать пользователя, с этой задачей данное приложение так же справляется изящно, благодаря системе наград, и самое главное благодаря статистике, предоставляемой приложением, которая показывает прогресс в изучении и удерживает пользователя на платформе.

Интересным аналогом так же является сайт Memrise, который нацелен исключительно на заучивание новых слов. Пользователю предоставляется программа, состоящая из нескольких этапов, которые содержат в себе определённый набор слов для заучивания, для перехода на следующий этап необходимо набрать определённое количество очков. Пользователю так же как и в предыдущем приложении предоставляется возможность следить за статистикой своего прогресса и соревноваться с другими пользователями в темпе набора очков. Пример одного из экранов представлен ниже на рисунке 1.2.

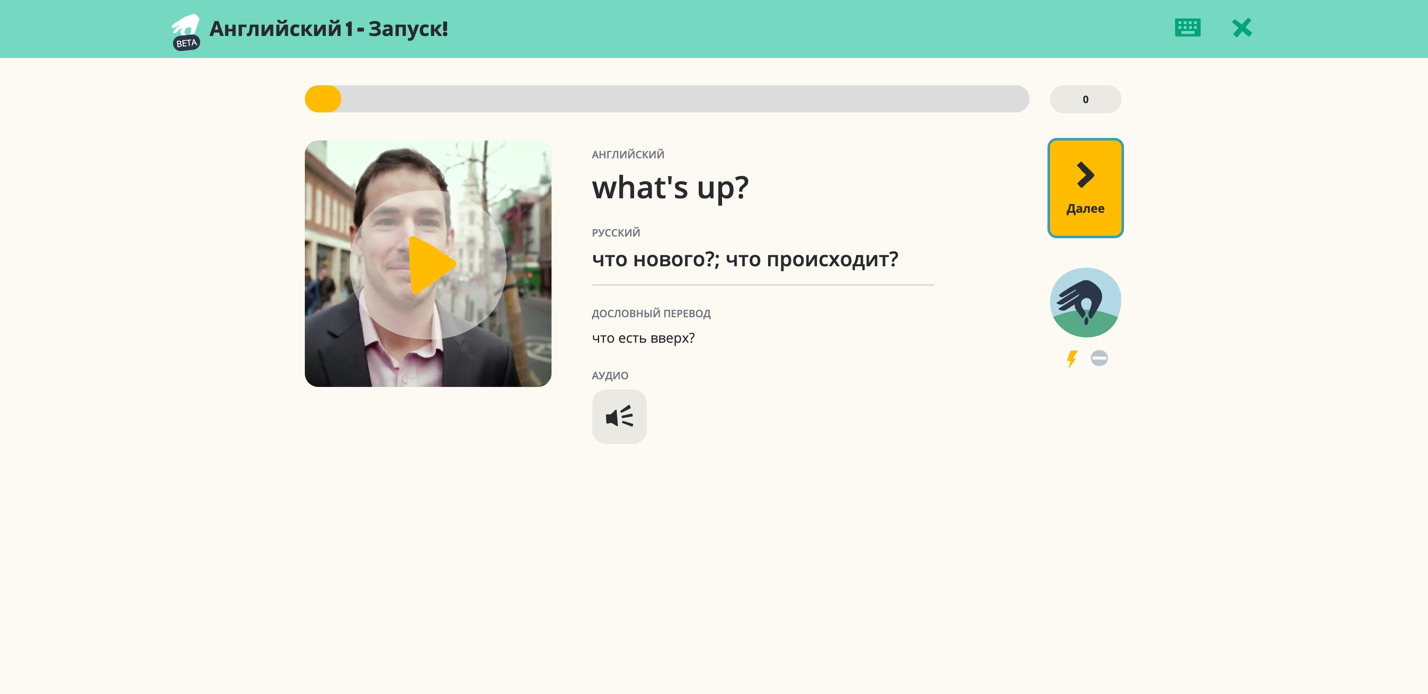


Рисунок 1.2 – [Street Fighter II](https://ru.wikipedia.org/wiki/Street_Fighter_II)

[Fatal Fury](https://ru.wikipedia.org/wiki/Fatal_Fury) от компании «[SNK](https://ru.wikipedia.org/wiki/SNK)» вышла всего несколькими месяцами позже Street Fighter II и имела несколько продолжений, а [Virtua Fighter](https://ru.wikipedia.org/wiki/Virtua_Fighter" \o "Virtua Fighter) (см. на рисунке 1.4) от «[Sega](https://ru.wikipedia.org/wiki/Sega" \o "Sega)» стала первым полностью [трёхмерным](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D1%91%D1%85%D0%BC%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0) файтингом. [4]



Рисунок 1.3 – Mortal Kombat

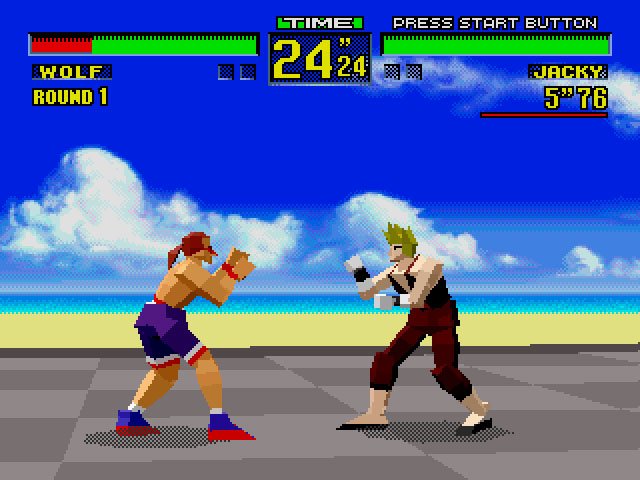


Рисунок 1.4 – Virtual Fighter

В 1994 году вышла первая игра серии [The King of Fighters](https://ru.wikipedia.org/wiki/The_King_of_Fighters" \o "The King of Fighters) от «SNK», особенностью которой была возможность играть сразу за трёх персонажей, предварительно задавая порядок их выхода на арену. The King of Fighters стала предтечей командных файтингов. В 1995 году компания «Capcom» выпустила игру [Street Fighter Alpha](https://ru.wikipedia.org/wiki/Street_Fighter_Alpha" \o "Street Fighter Alpha), дополнившую игровой процесс двумя важными способностями персонажей: блокированием ударов противника в воздухе и контратакой из блока. После этих нововведений процесс добавления в игровую механику новых деталей, значительно влияющих на [геймплей](https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%B9%D0%BC%D0%BF%D0%BB%D0%B5%D0%B9" \o "Геймплей), замедлился: основные механизмы уже были придуманы и нуждались только в отладке. [4]

С развитием файтингов популярность жанра «Избей их всех», чей золотой век пришёлся на 80-е годы XX века, начала медленно угасать. Но, отдавая дань старым традициям, многие разработчики современных файтингов включают в них режимы, напоминающие игры «Избей их всех». Примерами такого включения могут служить режимы «Tekken Force» в игре [Tekken 3](https://ru.wikipedia.org/wiki/Tekken_3" \o "Tekken 3) (см. на рисунок 1.5) от «[Namco](https://ru.wikipedia.org/wiki/Namco" \o "Namco)» и «Boost Mode» в [Guilty Gear Isuka](https://ru.wikipedia.org/wiki/Guilty_Gear_Isuka" \o "Guilty Gear Isuka) от «[Arc System Works](https://ru.wikipedia.org/wiki/Arc_System_Works" \o "Arc System Works)».

С ходом времени и развитием технологий начали появляться и трехмерные файтинги, но двухмерные файтинги не сдали своих позиций и продолжили развиваться.



Рисунок 1.5 – [Tekken 3](https://ru.wikipedia.org/wiki/Tekken_3)

Двухмерные файтинги (серии [Street Fighter](https://ru.wikipedia.org/wiki/Street_Fighter" \o "Street Fighter), [The King of Fighters](https://ru.wikipedia.org/wiki/The_King_of_Fighters" \o "The King of Fighters),  первые игры [Mortal Kombat](https://ru.wikipedia.org/wiki/Mortal_Kombat_(%D1%81%D0%B5%D1%80%D0%B8%D1%8F)" \o "Mortal Kombat (серия))) основаны на [спрайтах](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D1%80%D0%B0%D0%B9%D1%82_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D0%BA%D0%B0)) и представляют собой управляемую анимацию, сходную с [мультфильмами](https://ru.wikipedia.org/wiki/%D0%9C%D1%83%D0%BB%D1%8C%D1%82%D1%84%D0%B8%D0%BB%D1%8C%D0%BC), однако некоторые игры (в частности, кроссовер [Marvel vs. Capcom 3](https://ru.wikipedia.org/wiki/Marvel_vs._Capcom_3:_Fate_of_Two_Worlds" \o "Marvel vs. Capcom 3: Fate of Two Worlds), [Street Fighter 4](https://ru.wikipedia.org/wiki/Street_Fighter_4" \o "Street Fighter 4) и его [апдейт](https://ru.wikipedia.org/wiki/Super_Street_Fighter_IV" \o "Super Street Fighter IV), а также [Mortal Kombat (2011)](https://ru.wikipedia.org/wiki/Mortal_Kombat_(2011)" \o "Mortal Kombat (2011)) (см. на рисунок 1.6)) при сохранении двухмерной механики боя используют трёхмерную графику.



Рисунок 1.6 – [Mortal Kombat (2011)](https://ru.wikipedia.org/wiki/Mortal_Kombat_(2011))

Большая часть двухмерных файтингов использует только один [план](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%BD_(%D0%BA%D0%B8%D0%BD%D0%BE)): персонажи игры могут перемещаться вдоль арены, но не могут отступать вглубь неё. Угол, под которым игрок видит арену и персонажей, всегда остаётся неизменным. Поскольку двухмерные файтинги не могут использовать глубину экрана, они делают упор на использовании высоты. Атаки с прыжка являются необходимыми элементами игры. Другие характерные особенности включают атаки с применением мелких материализуемых предметов или сгустков энергии ([проджектилей](https://ru.wiktionary.org/wiki/%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5:%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%84%D0%B0%D0%B9%D1%82%D0%B8%D0%BD%D0%B3-%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%BE%D0%B2" \l "Projectile" \o "wikt:Приложение:Список файтинг-терминов) на сленге игроков) и двухуровневую систему атак и блоков, делящихся на верхние и нижние. Третий вид блоков — воздушные — присутствует не во всех играх. Поскольку файтинги изначально были именно двухмерными, их игровая система на протяжении долгого времени подвергалась отладкам и дополнениям. В результате двухмерные файтинги часто содержат больше известных и давно применяющихся техник, чем трёхмерные игры того же жанра.

В трёхмерных файтингах ([Virtua Fighter](https://ru.wikipedia.org/wiki/Virtua_Fighter" \o "Virtua Fighter), [Soul Calibur](https://ru.wikipedia.org/wiki/Soul_Calibur" \o "Soul Calibur), [Tekken](https://ru.wikipedia.org/wiki/Tekken_(%D1%81%D0%B5%D1%80%D0%B8%D1%8F)" \o "Tekken (серия)), [Dead or Alive](https://ru.wikipedia.org/wiki/Dead_or_Alive_(%D1%81%D0%B5%D1%80%D0%B8%D1%8F_%D0%B8%D0%B3%D1%80)" \o "Dead or Alive (серия игр)), [Last Bronx](https://ru.wikipedia.org/w/index.php?title=Last_Bronx&action=edit&redlink=1" \o "Last Bronx (страница отсутствует)), [Mortal Kombat 4](https://ru.wikipedia.org/wiki/Mortal_Kombat_4" \o "Mortal Kombat 4)) все графические элементы представляют собой [полигональные](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%B3%D0%BE%D0%BD) модели, которые игрок может рассматривать под любым углом и с любой стороны. [4]

Тем не менее, трёхмерные файтинги зачастую позволяют игроку оптимально использовать только два измерения, делая игру фактически двухмерной (с тем отличием, что 2D-игры используют ширину и высоту, а 3D-игры — ширину и глубину). Геймплей некоторых трёхмерных файтингов сильно напоминает двухмерные игры: третье измерение, глубина арены, используется скорее, как одна из дополнительных возможностей уклонения от атак противника, чем как неотъемлемая часть боевой системы.

В отличие от двухмерных файтингов, персонажи трёхмерных игр способны отходить вглубь арены, прыжки и атаки с применением снарядов имеют намного меньшее значение в игровом процессе, а уровни атак и блоков сложнее и в общем случае делятся на верхний, средний и нижний. Помимо этого, трёхмерные файтинги характеризуются более низкими скоростями атак. Анимированные 3D-модели в играх часто создаются с помощью специального программного обеспечения, которое обрабатывает показания датчиков, укреплённых на теле выполняющего то или иное движение человека. Записанные движения модели требуют для отображения более длинной последовательности [кадров](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%B4%D1%80), чем рисованные спрайты, однако выглядят реалистичнее.

Арены в трёхмерных файтингах намного больше, чем в двухмерных. На арене часто присутствуют объекты, угрожающие [здоровью](https://ru.wikipedia.org/wiki/%D0%9E%D1%87%D0%BA%D0%B8_%D0%B6%D0%B8%D0%B7%D0%BD%D0%B8) персонажей. Такие объекты можно использовать против оппонента, к примеру, столкнуть его с обрыва, швырнуть в сторону оппонента, проходящего мимо человека или бросить на проволоку, по которой проходит [электрический ток](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D1%82%D0%BE%D0%BA). Во многих трёхмерных файтингах победа игроку засчитывается в двух случаях: если у противника не осталось здоровья, или если он оказался за пределами арены. Выбить противника с арены можно удачно попавшей в него атакой или многочисленными ударами по оппоненту в то время, когда он находится в блоке: механика файтингов подразумевает, что блокирующий персонаж после получения удара немного «отъезжает» от атакующего, и таким образом может быть вытеснен с поля боя. В некоторых играх, например, в серии [Super Smash Brothers](https://ru.wikipedia.org/wiki/Super_Smash_Brothers), вытеснение ([англ.](https://ru.wikipedia.org/wiki/%D0%90%D0%BD%D0%B3%D0%BB%D0%B8%D0%B9%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) [ring out](https://ru.wiktionary.org/wiki/%D0%9F%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5:%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D1%84%D0%B0%D0%B9%D1%82%D0%B8%D0%BD%D0%B3-%D1%82%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%BE%D0%B2#Ring_Out)) — основной или даже единственный способ одержать победу. [6]

Изучив программы подобного жанра, мы приняли решение, что нашему приложению необходимо:

* запоминающиеся персонажи и анимации;
* возможность кастомизации приложения;
* соблюдений правил игры жанра файтинг;
* сбалансированность персонажей.

## Постановка задачи

В рамках данного курсового проекта планируется разработка веб-приложения «ReWord». В том числе решение таких задач как:

* создание серверной части приложения;
* создание сайта приложения;
* организация взаимодействия с базой данных;
* разработка масштабируемого дизайна.

Для разработки программного средства в большей части будет использован язык программирования Kotlin.

Проанализировав аналоги программного средства, можно поставить задачу: дать пользователю простой и понятный интерфейс взаимодействия с приложением, сделать приложение гибким для пользователя.

В веб-приложении «ReWord» необходимо реализовать следующие функции:

* проверку знания иностранных слов;
* возможность добавлять и удалять свои слова;
* анализ результатов проверки слов;
* выбор категории слов, которые пользователь хочет изучать в данный момент.

# Разработка программного средства

## Структура программного средства

В данном игровом приложении реализованы следующие модули:

* Routing – модуль серверной части веб-приложения, отвечающий за обработку запросов от клиентской части
* Transaction– модуль серверной части веб-приложения, отвечающий за работу с базой данных;
* Table – модуль серверной части веб-приложения, отвечающий за описания отношений базы данных;
* Model– модуль серверной части веб-приложения, отвечающий за описание приложения;
* Api– модуль клиентской части веб-приложения, отвечающий за совершения запросов к серверной части;
* UI – модуль клиентской части веб-приложения, отвечающий за отображение интерфейса приложения.

## Безопасность веб-приложения

Наиболее важной частью любого веб-приложения является его надежность и безопасность.

В данном веб-приложении была использован механизм мезопасности CORS (Cross-origin resource sharing), который позволяет веб-странице с одним доменом обращаться к ресурсу с другим доменом (кросс-доменный запрос). Ниже на рисунке 2.1 изображена схема работы механизма CORS. Данный механизм реализуется в серверной части веб-приложения посредством подключения библиотеки и включением данного плагина, c последующей его настройкой. Ниже приведена часть кода, отвечающая за настройку данного расширения.

*install*(CORS) {

method(HttpMethod.Post)

method(HttpMethod.Get)

header(HttpHeaders.Authorization)

allowCredentials = true

host("http://localhost:8080")

}

В данном фрагменте кода устанавливается расширение CORS для Ktor и указывается запросы с каких методов, с какого хоста и с какими заголовками будут использовать данный механизм. Так же устанавливается флаг allowCredential в значение true, этот флаг отвечает за создание заголовков, по которым браузер определяет, необходимо ли сохранять полученные от сервера cookie, которые в последствии будут использованы для организации сессии.

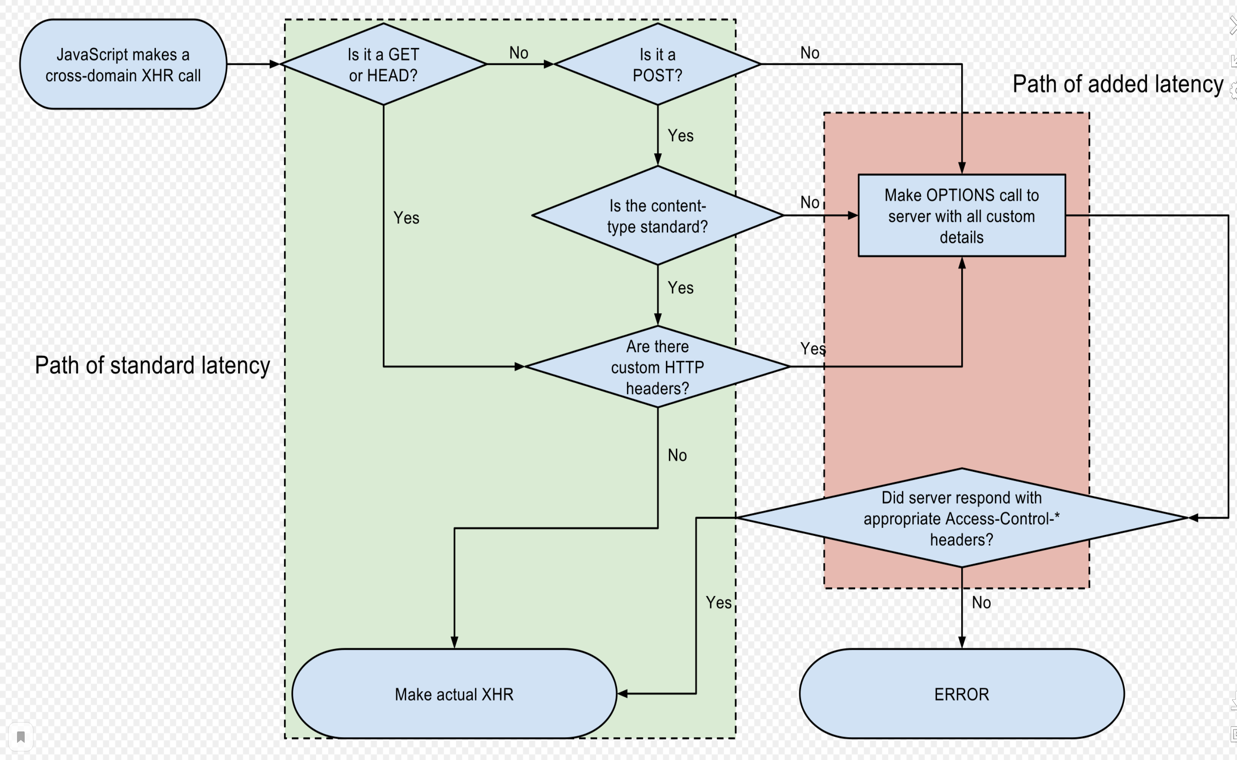


Рисунок 2.1 – схема работы механизма CORS

На стороне клиентской части веб-приложения для всех запросов так же переопределяется значение credentials на значения INCLUDE, что говорит браузеру о необходимости отправки cookie. Фрагмент кода с переопределением поля credentials приведён ниже.

val responsePromise = *window*.fetch(url, object : RequestInit {

override var method: String? = method

override var body: dynamic = body

override var credentials: RequestCredentials? = RequestCredentials.Companion.*INCLUDE*

override var headers: dynamic = *json*(

"Accept" *to* "application/json"

)

})

Данное веб-приложение использует cookie для организации сеанса посредством сессии, однако в последующем планируется перейти на организацию сеанса на клиентской стороне веб-приложения посредством использования Json Web Token.

## Организация клиент-серверного взаимодействия

Одной из основных целей курсового проекта была оптимизация расходов времени и ресурсов процессора на отрисовку и при этом улучшение качетсва графики. В ходе изучения теоретических материалов и разработки приложения были выявлены основные параметры, влияющие на скорость и эффективность программного средства:

* способ отрисовки;
* количество используемых изображений;
* сложность расчетов непосредственно в течение игрового процесса.

Для улучшения качества изображения и увеличения скорости отрисовки используется двойная буфферизация, то есть отрисовка отдельных объектов происходит сначала в контексте графического устройства в памяти, затем изображение переносится на экран. [8]

Оптимизировать способ отрисовки можно за счет неполного копирования байт изображения. В данном приложении такая оптимизация происходит за счет использования функции GdiTransparentBlt, одним из параметров которой является шестнадцатеричное значение цвета, который пропускается при копировании. [10]

Дескрипторы всех изображений, которые используются в приложении, хранятся в разных переменных, и сами изображения (массивы байтов) загружаются в разные области оперативной памяти. При использовании большого количества картинок, работа с ними замедляется из-за сложного доступа к данным, также увеличивается размер программы. Хорошим решением этой проблемы стало использование спрайтов. Спрайтом, называют растровое изображение, содержащее различные состояния одного объекта, что эффективно применяется для работы с анимацией (см. на рисунок 2.9). В идеальном случае необходимо найти баланс между количеством изображений, размещенных на одном спрайте, и сложностью расчета координат. В данном проекте спрайты используются для объедения состояний отдельных объектов.



Рисунок 2.9 – пример спрайта

Для вывода фона игры используется процедура DrawBackGround (см. на рисунок 2.10).

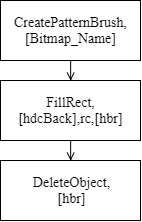


Рисунок 2.10 – отрисовка заднего фона

Функция CreatePatternBrush создает логическую кисть с заданным точечным рисунком узора, куда мы передаем битмап фона. Затем функция FillRect заполняет прямоугольник, используя кисть Brush, до правой и нижней границы. После использование логического пера, кисти, шрифта, необходимо удалять, освобождая все системные ресурсы, связанные с объектом. [11]

Логика вывода анимации выглядит так: при нажатии на копку персонаж выполняет необходимый удар, и после этого переходит в статическое состояние. В программе это выглядит следующим образом: выбирается необходимый спрайт анимации удара, происходит его обработка и отображение на экране.

После выполнения анимации удара, если не нажата другая клавиша, персонаж переходит в статическую стойку до следующего удара.



Рисунок 2.11 – статическая стойка правого персонажа

Сложность расчетов в течение игрового процесса влияет на скорость обновления графики. При разработке данного приложения были применены следующие способы оптимизации:

* проведение загрузки изображений перед началом боя: расчет всех постоянных координат и размеров, загрузка изображений, инициализация переменных происходит при запуске программы;
* использование регистров: доступ к значениям, хранящимся в регистрах, как правило, быстрее, чем доступ к переменным в памяти. Если некоторые переменные в процедуре используются часто, они копируются в регистры в начале процедуры;
* замена сложных операций более простыми;

В программе почти не используется оператор div, так как на его выполнение уходит сравнительно продолжительное время, что при частом использовании влияет на скорость. По возможности вычисления модифицируются таким образом, чтобы достаточно было оператора shr (сдвиг вправо), так как сдвиги выполняются быстрее. По этой же причине оператор mul заменяется на shl (сдвиг влево).

# Руководство по использованию программного средства

При запуске игры пользователям открывается чат, в который необходимо внести данные подключаемого ip адреса или «hostname», далее нажать на . Далее необходимо ввести номер прослушиваемого порта и нажать кнопку «Listen» (оппонент должен ввести номер порта в поле «connect» и нажать «Connect», номера портов должны совпадать). Если соединение прошло успешно, то можно обменяться сообщениями, введя их в специальное поле и нажав на «Send».

Для запуска игры оба оппонента должны нажать «Start game».

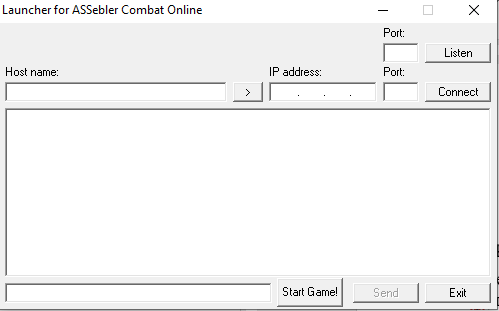


Рисунок 3.1 – Демонстрация чата

Находясь в главном меню, пользователь может начать игру, нажав на интерактивную кнопку «Начать», открыть дисковод, чтобы использовать его как подставку, нажав на кнопку «Получить подставку для пива» либо завершить работу программы нажатием кнопки «Отключиться». Вид главного меню представлен на рисунке 3.2.

Во время пребывания в игровом окне пользователь может приостановить игру, нажав Esc. Из состояния паузы игрок может продолжить игру, нажав кнопку «Играть», или выйти, нажав кнопку «Отключиться».

Управление левого персонажа:

* A – движение влево;
* D – движение вправо;
* W – прямой удар;
* S – апперкот;
* E – удар в прыжке;
* F – удар снизу.

Управление правого персонажа:

* ◄– движение влево;
* ►– движение вправо;
* ▲– прямой удар;
* ▼– сильный удар;



Рисунок 3.2 – главное меню

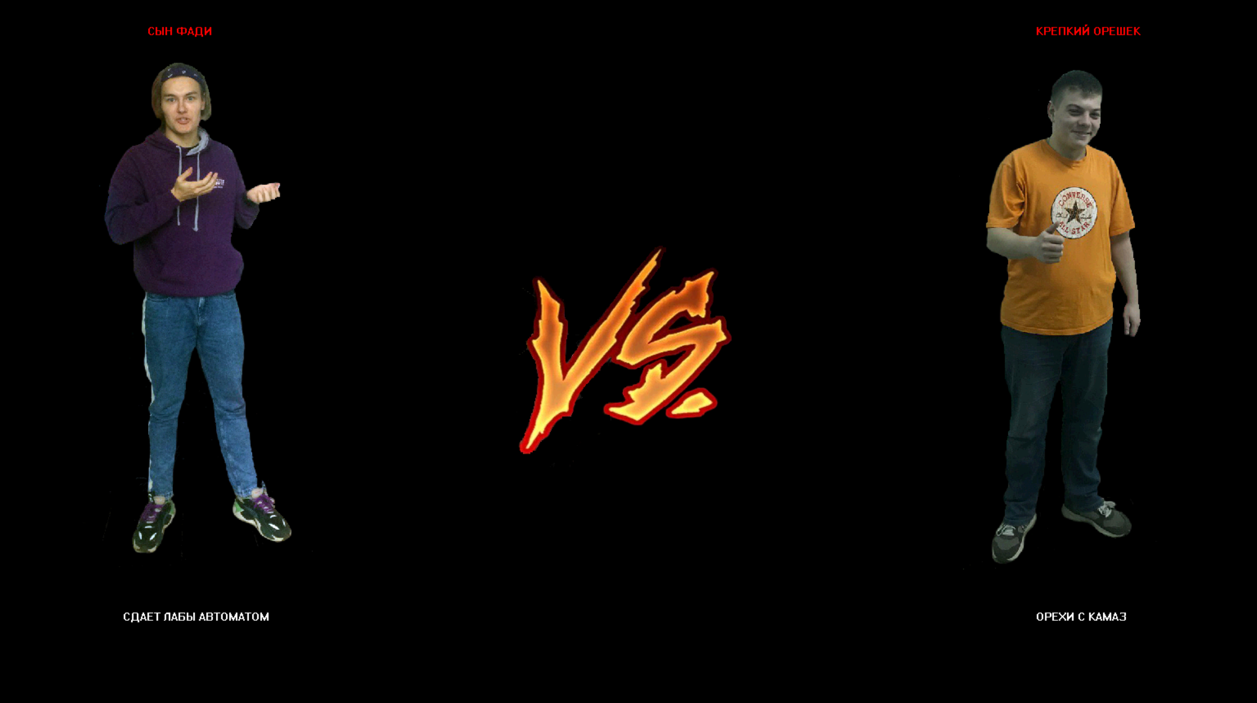


Рисунок 3.3 – экран соперников



Рисунок 3.4 – демонстрация игрового процесса

Заключение

В современном мире существует множество игровых приложений, которые помогают занять свободное время в очереди или в дороге, отдохнуть или просто хорошо провести время с друзьями. Огромное количество людей ежедневно играют в компьютерные игры и спрос на них только растет.

В рамках данной курсовой работы был реализован стиль игрового проекта «Assembler Combat», предназначенного для развлечения пользователей. Разработанное программное средство отличается простым и понятным интерфейсом, плавностью и высокой скоростью работы.

Отдельным плюсом данного приложения является его модифицируемость. Если пользователь предпочтет сменить фон игры, то для этого понадобится заменить соответствующее изображение в папке ресурсов проекта.

Таким же образом можно заменить и анимации самих персонажей, заменив изображения в папке на собственные, но придется учесть количество кадров в спрайте.

Список использованной литературы

[1] Windows API [Электронный ресурс] – Режим доступа:

<https://docs.microsoft.com/en-us/windows/win32/apiindex/windows-api-list/> . – Дата доступа: 10.12.2020.

[2] Учебник по FASM [Электронный ресурс] – Режим доступа:

<http://asmworld.ru/uchebnik/>. – Дата доступа: 10.12.2020.

[3] Windows GDI [Электронный ресурс] – Режим доступа:

<https://docs.microsoft.com/en-us/windows/win32/api/_gdi>. – Дата доступа: 10.12.2020.

[4] Файтинг [Электронный ресурс] – Режим доступа:

<https://ru.wikipedia.org/wiki/Файтинг>. – Дата доступа: 20.12.2020.

[5] Аркадный автомат [Электронный ресурс] – Режим доступа:

[https://ru.wikipedia.org/wiki/Аркадный\_автомат](https://ru.wikipedia.org/wiki/Файтинг). – Дата доступа: 20.12.2020.

[6] Компьютерная игра [Электронный ресурс] – Режим доступа:

[https://ru.wikipedia.org/wiki/Компьютерная\_игра](https://ru.wikipedia.org/wiki/Файтинг). – Дата доступа: 20.12.2020.

[7] Спрайт [Электронный ресурс] – Режим доступа:

[https://ru.wikipedia.org/wiki/Спрайт\_(компьютерная\_графика)](https://ru.wikipedia.org/wiki/Файтинг). – Дата доступа: 20.12.2020.

[8] Буферизация [Электронный ресурс] – Режим доступа:

[https://ru.wikipedia.org/wiki/Буферизация\_(информатика)](https://ru.wikipedia.org/wiki/Файтинг). – Дата доступа: 20.12.2020.

[9] Функция GetSystemMetrics[Электронный ресурс] – Режим доступа:

<https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-getsystemmetrics/> . – Дата доступа: 20.12.2020.

[10] Функция GdiTransparentBlt [Электронный ресурс] – Режим доступа:

<https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-gditransparentblt/> . – Дата доступа: 20.12.2020.

[11] Функция CreatePatternBrush [Электронный ресурс] – Режим доступа:

[https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-createpatternbrush/](file:///Users/helicoptera/Desktop/Course/%20https:/docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-createpatternbrush) . – Дата доступа: 20.12.2020.

Приложение А

(обязательное)

Исходный код программы

proc TextDraw hdc,left,top,string,strlen,color ;Вывод текста

invoke SetBkColor,[hdc],0h

invoke SetTextColor,[hdc],[color]

invoke TextOut,[hdc],[left],[top],[string],[strlen]

ret

endp

proc StopWalking,Player

mov ecx,[Player]

mov [ecx+PLAYER.Walk\_Stopped],1

mov [ecx+PLAYER.Action\_State],3

stdcall InitBitmap,[Player]

ret

endp

proc AnimateBitmap seg\_w,seg\_h,x0,y0,iterat,hbmImage,color,Player ;Анимация битмапа взависимости от числа замены кадров

invoke GetObjectA,[hbmImage],sizeof.BITMAP, bm

mov ecx,[Player]

inc [ecx+PLAYER.Cadr\_Change]

mov ebx,[iterat]

cmp [ecx+PLAYER.Cadr\_Change],ebx

jne @f

stdcall Dinamics,[seg\_w],[seg\_h],[bm.bmWidth],[bm.bmHeight],[Player]

mov [ecx+PLAYER.Cadr\_Change],0

@@:

invoke SelectObject,[hdcMem], [hbmImage]

mov ecx,[Player]

invoke GdiTransparentBlt,[hdcBack],[x0],[y0],[seg\_w],[seg\_h],[hdcMem],[ecx+PLAYER.Bitmap\_begX],[ecx+PLAYER.Bitmap\_begY],[seg\_w],[seg\_h],[color]

ret

endp

proc Dinamics w,h,full\_w,full\_h,Player ;Нарезка битмапа на необходимые по текущему времени части

mov ebx,[w]

mov ecx,[Player]

add [ecx+PLAYER.Bitmap\_begX],ebx

sub [full\_w],50

mov ebx,[full\_w]

cmp [ecx+PLAYER.Bitmap\_begX],ebx

jb .Stop

mov [ecx+PLAYER.Bitmap\_begX],0

mov ebx,[h]

add [ecx+PLAYER.Bitmap\_begY],ebx

sub [full\_h],30

mov ebx,[full\_h]

cmp [ecx+PLAYER.Bitmap\_begY],ebx

jb .Stop

mov [ecx+PLAYER.Bitmap\_begY],0

.Stop:

ret

endp

proc InitBitmap Player ;Сброс кадров битмапа

mov ecx,[Player]

mov [ecx+PLAYER.Bitmap\_begX],0

mov [ecx+PLAYER.Bitmap\_begY],0

mov [ecx+PLAYER.Action\_Time\_End],0

ret

endp

proc DetectHitKey Animation\_Priority,Animation\_Interval,Player ;Инициаизация данных по клавише удара

mov ebx,[time]

mov ecx,[Player]

cmp ebx,[ecx+PLAYER.Action\_Time\_End]

jb WindowProc.defwndproc

je @f

stdcall InitBitmap,[Player]

mov ebx,[Animation\_Priority]

mov [ecx+PLAYER.Action\_State],ebx

mov eax,[time]

add eax,[Animation\_Interval]

mov [ecx+PLAYER.Action\_Time\_End],eax

jmp WindowProc.finish

@@:

mov [ecx+PLAYER.Action\_Time\_End],0

mov [ecx+PLAYER.Action\_State],3

jmp WindowProc.finish

ret

endp

proc DetectMoveKey Animation\_Priority,Animation\_Interval,Player ;Инициализация данных под клавишу движения

mov ecx,[Player]

mov ebx,[time]

cmp ebx,[ecx+PLAYER.Action\_Time\_End]

jb WindowProc.defwndproc

stdcall InitBitmap,[Player]

mov ebx,[Animation\_Priority]

mov [ecx+PLAYER.Action\_State],ebx

mov eax,[time]

add eax,[Animation\_Interval]

mov [ecx+PLAYER.Action\_Time\_End],eax

mov [ecx+PLAYER.Walk\_Stopped],3

jmp WindowProc.finish

ret

endp

proc CheckForZero Player ;Проверка окончания анимации

mov ecx,[Player]

mov ebx,[time]

cmp ebx,[ecx+PLAYER.Action\_Time\_End]

jbe .Next

mov [ecx+PLAYER.Action\_State],3

mov [ecx+PLAYER.Cadr\_Change],0

mov [ecx+PLAYER.Action\_Time\_End],0

stdcall InitBitmap,[Player]

.Next:

ret

endp

proc DrawBackGround Bitmap\_Name,f ;Вывод фона

cmp [f],0

jne @f

invoke CreatePatternBrush,[Bitmap\_Name]

mov [hbr],eax

jmp .Next

@@:

invoke CreateSolidBrush,0h

mov [hbr],eax

.Next:

invoke FillRect,[hdcBack],rc,[hbr]

invoke DeleteObject,[hbr]

ret

endp

proc InitPlayer Player

mov ecx,[Player]

mov [ecx+PLAYER.Action\_Time\_End], 0

mov [ecx+PLAYER.Action\_State],3

mov [ecx+PLAYER.Walk\_Stopped],1

mov [ecx+PLAYER.Cadr\_Change],0

mov [ecx+PLAYER.x],0

mov [ecx+PLAYER.y],0

mov [ecx+PLAYER.Bitmap\_begX],0

mov [ecx+PLAYER.Bitmap\_begY],0

mov [ecx+PLAYER.Health],300

mov [ecx+PLAYER.Blood\_Time],0

mov [ecx+PLAYER.Kicks\_for\_Fall],0

mov [ecx+PLAYER.Delta\_Health],0

mov [tolpaduration],120

ret

endp