Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра Информатики

Дисциплина: ООП

Отчет

по курсовому проекту

на тему

Компьютерная Игра

Django Unchained

Студент: гр.253505

Волчецкий А.М.

Руководитель: ассистент кафедры информатики

Юзефович C.В.

Минск 2014

Содержание

[Содержание.......................................................................................................2](#_Toc387579318)

[Введение............................................................................................................](#_Toc387579319)3

Классы.....……………………………………………………………………….7

[Основные методы и алгоритмы.......................................................................](#_Toc387579320)10

[Руководство пользователя..............................................................................1](#_Toc387579321)4

[Заключение…..................................................................................................1](#_Toc387579322)7

[Список использованных источников............................................................1](#_Toc387579323)8

Введение

Созданный компанией Microsoft для поддержки среды .NET Framework, язык C# опирается на богатое наследие в области программирования. Его главным архитектором был ведущий специалист в этой области — Андерс Хейлсберг (Anders Hejlsberg). C# — прямой потомок двух самых успешных в мире компьютерных языков: C и C++. От C он унаследовал синтаксис, ключевые слова и операторы. Он позволяет построить и усовершенствовать объектную модель, определенную в C++. Кроме того, C# близко связан с другим очень успешным языком: Java. Имея общее происхождение, но различаясь во многих важных аспектах, C# и Java — это скорее “двоюродные братья”. Например, они оба поддерживают программирование распределенных систем и оба используют промежуточный код для достижения переносимости, но различаются при этом в деталях реализации.

Опираясь на мощный фундамент, который составляют унаследованные

характеристики, C# содержит ряд важных новшеств, поднимающих искусство программирования на новую ступень. Например, в состав элементов языка C# включены такие понятия, как делегаты (представители), свойства, индексаторы и события. Добавлен также синтаксис, который поддерживает атрибуты; упрошено создание компонентов за счет исключения проблем, связанных с COM (Component Object Model — модель компонентных объектов Microsoft — стандартный механизм, включающий интерфейсы, с помощью которых объекты предоставляют свои службы другим объектам). И еще. Подобно Java язык

C# предлагает средства динамического обнаружения ошибок, обеспечения безопасности и управляемого выполнения программ. Но, в отличие от Java, C# дает программистам доступ к указателям. Таким образом, C# сочетает первозданную мощь C++ с типовой безопасностью Java, которая обеспечивается наличием механизма контроля типов (type checking) и корректным использованием шаблонных классов (template class). Более того, язык C# отличается тем, что компромисс между мощью и надежностью тщательно сбалансирован и практически прозрачен (не заметен для пользователя или программы). На протяжении всей истории развития вычислительной техники эволюция языков программирования означала изменение вычислительной среды, способа мышления

программистов и самого подхода к программированию. Язык C# не является исключением. В непрекращающемся процессе усовершенствования, адаптации и внедрения нововведений

C# в настоящее время находится на переднем крае. Это — язык, игнорировать

существование которого не может ни один профессиональный программист.

**Платформа XNA Framework** – это большой набор системных библиотек, построеннных на базе .NETTбиблиотек и направленных на улучшение и упрощение создания игр для всех продуктов семейства Microsoft. Создавая игру для PC на базе платформы XNA Framework, вы можете быть уверены, что эта игра будет работать на приставке Xbox 360 и в скором времени на новой системе Windows Mobile. Иссключения в этом случае могут составлять некоторые системные классы, которые необходимы только для работы с той или иной платформой. Например, приставка Xbox 360 не имеет мыши, поэтому в программах для консоли нельзя использовать методы и классы, направленные на работу с мышью. В свою очередь, для компьютерных игр можно задавать различные разрешения экрана, тогда как Xbox 360 использует в качестве монитора телевизор, который имеет свою специфику вывода изображения на экран. Эти и другие исключения составляют мышиную долю (примерно 3–5%) от всей библиотеки XNA Framework. При правильном подходе в использовании библиотечных классов можно написать программу, которая на 100% будет работать как на консоли, так и на компьютере и, надеемся, на Windows Mobile тоже.

Еще одной важной особенностью платформы XNA Framework является значительное упрощение в подходе реализации игр для всех платформ. Прежде чем начать писать именно исходный код игр с использованием DirectX SDK, необходимо создать большую кучу различных объектов, создать окно, настроить видеоадаптер и т. д. Применяя платформу XNA Framework, вы все перечисленное и даже больше создадите за пару строк исходного кода, при этом эти строки сформирует за вас сам компилятор и XNA Framework! Работать стало проще, быстрее, и главное – нет борьбы и драк с DirectX… Роль DirectX в этом деле – это низкоуровневая прослойка API, к которой вы не касаетесь никоим образом. Все манипуляции производятся только через библиотеку XNA Framework.

**Компьютерная игра** (иногда используется неоднозначный термин **видеоигра**) — [компьютерная программа](http://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%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0), служащая для организации [игрового](http://ru.wikipedia.org/wiki/%D0%98%D0%B3%D1%80%D0%B0) процесса ([геймплея](http://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%B9%D0%BC%D0%BF%D0%BB%D0%B5%D0%B9" \o "Геймплей)), связи с партнёрами по игре, или сама выступающая в качестве [партнёра](http://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B0%D1%80%D1%80%D0%B8%D0%BD%D0%B3).

Компьютерные игры часто создаются на основе [фильмов](http://ru.wikipedia.org/wiki/%D0%9A%D0%B8%D0%BD%D0%BE%D1%84%D0%B8%D0%BB%D1%8C%D0%BC) и [книг](http://ru.wikipedia.org/wiki/%D0%A5%D1%83%D0%B4%D0%BE%D0%B6%D0%B5%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5); есть и обратные случаи. С 2011 года компьютерные игры официально признаны в [США](http://ru.wikipedia.org/wiki/%D0%A1%D0%A8%D0%90) отдельным [видом искусства](http://ru.wikipedia.org/wiki/%D0%98%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%BE).

Компьютерные игры оказали столь существенное влияние на общество, что в [информационных технологиях](http://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D1%82%D0%B5%D1%85%D0%BD%D0%BE%D0%BB%D0%BE%D0%B3%D0%B8%D0%B8) отмечена устойчивая тенденция к [геймификации](http://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%B9%D0%BC%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) для неигрового [прикладного программного обеспечения](http://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D0%BA%D0%BB%D0%B0%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)

Компьютерные игры могут быть классифицированы по нескольким признакам:

* **Жанр**: игра может принадлежать как к одному, так и к нескольким жанрам, а в уникальных случаях — открывать новый или быть вне всяких жанров;
* **Количество игроков и способ их взаимодействия**: игра может быть [однопользовательской](http://ru.wikipedia.org/wiki/%D0%9E%D0%B4%D0%BD%D0%BE%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) — рассчитанной на игру одного человека, или [многопользовательской](http://ru.wikipedia.org/wiki/%D0%9C%D0%BD%D0%BE%D0%B3%D0%BE%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) — рассчитанной на одновременную игру нескольких человек; а также вестись на одном компьютере, через интернет, электронную почту, или [массово](http://ru.wikipedia.org/wiki/MMOG);
* **Визуальное представление**: игра может как использовать графические средства оформления, так и напротив, быть текстовой. Игра также может быть двухмерной или трехмерной. Есть и [звуковые игры](http://ru.wikipedia.org/wiki/%D0%97%D0%B2%D1%83%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) — в них вместо визуального представления используются звуки.
* **Платформа**: игра может принадлежать как к одной платформе, так и быть мультиплатформенной.

### Классификация по жанрам

Жанр определяется целью игры. Выделяют следующие жанры:

* [*Приключенческая игра*](http://ru.wikipedia.org/wiki/%D0%9A%D0%B2%D0%B5%D1%81%D1%82) (Adventure) — игра, обладающая полноценным литературным сюжетом, и игрок в процессе игры сам раскрывает все перипетии этого сюжета.
* [*Боевик*](http://ru.wikipedia.org/wiki/Action) (Action) — игра, состоящая, в основном, из боевых сцен, драк и перестрелок. Боевики подразделяются на:
  + «[шутеры](http://ru.wikipedia.org/wiki/%D0%A8%D1%83%D1%82%D0%B5%D1%80" \o "Шутер)» (от [англ.](http://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) *shoot — стрелять*) (пример: [Counter-Strike](http://ru.wikipedia.org/wiki/Counter-Strike" \o "Counter-Strike), серии игр [Call of Duty](http://ru.wikipedia.org/wiki/Call_of_Duty) и [Battlefield](http://ru.wikipedia.org/wiki/Battlefield" \o "Battlefield));
  + «[файтинг](http://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D1%82%D0%B8%D0%BD%D0%B3" \o "Файтинг)» (от [англ.](http://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) *fight — драться*) (пример: [Mortal Kombat](http://ru.wikipedia.org/wiki/Mortal_Kombat" \o "Mortal Kombat));
  + «[ужасы](http://ru.wikipedia.org/wiki/Survival_horror)» (примеры: серия игр [F.E.A.R.](http://ru.wikipedia.org/wiki/F.E.A.R.), [Silent Hill](http://ru.wikipedia.org/wiki/Silent_Hill_(%D1%81%D0%B5%D1%80%D0%B8%D1%8F_%D0%B8%D0%B3%D1%80)), [Resident Evil](http://ru.wikipedia.org/wiki/Resident_Evil));
  + «[стелс](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B5%D0%BB%D1%81-%D1%8D%D0%BA%D1%88%D0%B5%D0%BD)» (от [англ.](http://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) *stealth — скрытный*) (пример: серии игр [Hitman](http://ru.wikipedia.org/wiki/Hitman_(%D1%81%D0%B5%D1%80%D0%B8%D1%8F_%D0%B8%D0%B3%D1%80)" \o "Hitman (серия игр)), [Metal Gear Solid](http://ru.wikipedia.org/wiki/Metal_Gear_Solid) и [Splinter Cell](http://ru.wikipedia.org/wiki/Splinter_Cell)).
* [*Ролевая игра*](http://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_%D1%80%D0%BE%D0%BB%D0%B5%D0%B2%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) (RPG — [англ.](http://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) *Role Playing Game*) — игра, отличительной особенностью которой является наличие у персонажей определённых навыков и характеристик, которые можно обрести, а впоследствии развивать, выполняя какие-либо действия. К этому жанру относятся и многопользовательские ролевые игры ([ММОРПГ](http://ru.wikipedia.org/wiki/%D0%9C%D0%9C%D0%9E%D0%A0%D0%9F%D0%93)), которые, в отличие от однопользовательских, не имеют ни конечной цели, ни законченного сюжета. Примеры: [The Elder Scrolls](http://ru.wikipedia.org/wiki/The_Elder_Scrolls" \o "The Elder Scrolls) (серия), [Gothic (серия)](http://ru.wikipedia.org/wiki/Gothic_(%D1%81%D0%B5%D1%80%D0%B8%D1%8F)" \o "Gothic (серия)).
* [*Стратегическая игра*](http://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_%D1%81%D1%82%D1%80%D0%B0%D1%82%D0%B5%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D0%B8%D0%B3%D1%80%D0%B0) (Strategy) — игра, представляющая собой управление масштабными процессами, как, например, строительство городов, ведение бизнеса, командование армией и т. д. Игровой процесс может идти как в реальном времени ([RTS](http://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D0%B0%D1%82%D0%B5%D0%B3%D0%B8%D1%8F_%D0%B2_%D1%80%D0%B5%D0%B0%D0%BB%D1%8C%D0%BD%D0%BE%D0%BC_%D0%B2%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%B8) — real time strategy), так и в пошаговом режиме (TBS — turn based strategy).
* [*Компьютерный симулятор*](http://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D0%B9_%D1%81%D0%B8%D0%BC%D1%83%D0%BB%D1%8F%D1%82%D0%BE%D1%80) (Simulator) — игра, полностью имитирующая какую-либо область реальной жизни, например, имитация управления гоночным автомобилем или самолётом.
* [*Головоломка*](http://ru.wikipedia.org/wiki/%D0%93%D0%BE%D0%BB%D0%BE%D0%B2%D0%BE%D0%BB%D0%BE%D0%BC%D0%BA%D0%B0_(%D0%B6%D0%B0%D0%BD%D1%80_%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D1%8B%D1%85_%D0%B8%D0%B3%D1%80)) (Puzzle) — игра, полностью или более чем наполовину состоящая из решения различных логических задач и головоломок.
* *Образовательная игра* — игра, включающая в себя элементы обучающих программ, которые подаются через сам игровой процесс и, благодаря повышению интереса к ним в связи с необычным антуражем, впоследствии хорошо запоминаются.
* *Забавы* — игры, рассчитанные, в основном, на детей, где психологическое впечатление от происходящего на экране гораздо важнее самого процесса игры — например, вид лопающихся пузырьков.

Жанровая классификация компьютерных игр предполагает введение нескольких дополнительных оснований:

* динамика — игровой процесс может происходить в условиях «реального времени» или пошагово;
* перспектива — игра может вестись как от первого, так и от третьего лица.

**Аркады** – это простейшие компьютерные игры, которые не требуют мощных ресурсов процессора, а также специальной компьютерной графики. Как правило, аркады это достаточно простые, как в плане графики, так и в плане сценария игры. Их особенность заключается в их простоте.

Играя в игры-аркады, игроку приходится принимать сиюминутные решения, полагаться на свою скорость и реакцию. Как правило, все аркады характеризуются большой системой бонусов. Начисление бонусов происходит как во время игры, так и после прохождения того или иного уровня.

Следует заметить, что сам термин появился тогда, когда в игровых залах начали устанавливать первые игровые автоматы. Игры были простыми для того, чтобы как можно большее количество людей смогли освоить их. В последствие все аркады стали использоваться на игровых приставках.

Аркады могут быть различных направлений. Например, аркады стрелялки, или аркады гонки, также существуют классические аркады. К ним относятся такие популярные игры как Pac Man, Mario или Icy Tower.

Следует сказать, что сегодня мода на аркадные игры снова возвращается. Они просты в управлении и не требуют больших затрат времени. В такую игру можно играть от 5 минут до бесконечности, именно поэтому такие игры сегодня становятся популярными.

**Классы**

Entity-базовый класс для всех игровых объектов. Все классы для представления объектов в игре являются наследниками этого класса. В нем происходит отрисовка объекта на экране и проверка на взаимодействие с платформами

Platform- класс служит для отрисовки платформы, по которой могут передвигаться игровые персонажи и стоять другие игровые объекты. Если предмет не взаимодействует с платформой, он будет падать вниз. В случае вылета за пределы экрана объект будет удален из игры. Если Джанго упадет в пропасть, игра закончится.

BulletsBonus- класс служит для отрисовки бонуса на пули, который будет давать Джанго 10 патронов. В нем идет отслеживание подбора бонуса и, после подбора его главным героем, удаление его из игры.

HealthBonus- класс служит для отрисовки бонуса на жизни, который будет давать Джанго 20 очков здоровья. В нем идет отслеживание подбора бонуса и, после подбора его главным героем, удаление его из игры.

HealthBonus- класс служит для отрисовки бонуса на скорость, который будет увеличивать скорость передвижений Джанго в 2 раза. В нем идет отслеживание подбора бонуса и, после подбора его главным героем, удаление его из игры.

Cactus- класс служит для отрисовки кактуса, который будет уменьшать здоровье Джанго на 10 очков при соприкосновении с ним, а также отталкивать его на определенное расстояние вверх, влево или вправо, если главный герой напрыгнул на кактус, подошел к нему слева или справа соответственно.

Mine- класс служит для отрисовки мины. Если Джанго сталкивается с ней, то мина активизируется и взорвется. Если Джанго или другой персонаж игры попадет в зону поражения, то у него отнимется 50 очков здоровья. Если в зону поражения попадет пороховая бочку, о которой будет рассказано ниже, то она взорвется.

PowderKeg – класс служит для отрисовки пороховой бочки. Если в нее попала пуля, либо она оказалась в зоне поражения гранаты или мины, то бочка взорвется. У тех персонажей, что попадут в зону поражения отнимется 50 очков здоровья

Bullet - класс служит для отрисовки пули, которыми умеет стрелять Джанго и его враги. При попадании пули в одного из персонажей у него отнимется 5 очков здоровья. Количество пуль в игре ограничено, пополнять его можно с помощью бонусов на пули.

Grenade- класс служит для отрисовки гранаты, бросать которые умеет только Джанго. Спустя 3 секунды граната взорвется. У тех персонажей, что попадут в зону поражения отнимется 25 очков здоровья

Character –класс является базовым для персонажей в игре. Персонаж—это объект, который умеет двигаться, стрелять и имеет очки здоровья. В этом классе находятся методы для передвижения персонажей по игровому уровню, а также отрисовки анимации. Все классы, которые представляют персонажа в игре являются наследниками этого класса(класс Character наследуется от класса Entity).

Enemy-класс служит для отрисовки врага Джанго. Цель врага в игре—убить главного героя. Враги имеют 10 очков здоровья. Они не могут взять ни один игровой бонус, а также активировать мину. Если Джанго попадет в «зону слежения» врага (т.е. он его увидит), в главного героя будет произведен выстрел.

Django- класс служит для главного героя игры—Джанго. Игрок управляет этим героем. Его цель—пройти до конца игры и убить всех врагов. Он взаимодействует со всеми объектами в игре. Для того, чтобы убить врага, игрок может воспользоваться различными способами: выстрелить из пистолета, бросить гранату, взорвать бочку с порохом, а так же активировать мину.

Основные методы и алгоритмы

1)

private void Gravity()

{

if ((PositionRectangle.Bottom < groundCoordinateY) && (IsGravity))

{

PositionRectangle = new Rectangle(PositionRectangle.X, PositionRectangle.Y + FallSpeed,

PositionRectangle.Width, PositionRectangle.Height);

IsInAir = true;

}

else

{

IsInAir = false;

}

}

Здесь обрабатывается гравитация в игре. Если игровой объект не взаимодействует с платформой и он может упасть, т.е. имеет гравитацию (например платформа не имеет гравитации, поэтому можно ее разместить в воздухе), он будет падать вниз

2)

private void Collisions()

{

var collidedEntities = CollisionWithEntities();

foreach (var entity in collidedEntities)

{

if ((entity.Key is Platform) || (entity.Key is PowderKeg) || (entity.Key is Enemy))

{

CollisionWithGameObject(entity);

}

else if (entity.Key is Cactus)

{

CollisionWithCactus(entity);

}

else if (entity.Key is HealthBonus)

{

HealthPoints += ((HealthBonus)entity.Key).GetBonus();

}

else if (entity.Key is BulletsBonus)

{

AllBullets += ((BulletsBonus)entity.Key).GetBonus();

}

else if (entity.Key is SpeedBonus)

{

var bonus = ((SpeedBonus)entity.Key).GetBonus();

MoveSpeed \*= bonus.Key;

lifeTimeOfSpeedBonus = bonus.Value;

}

else if (entity.Key is Mine)

{

((Mine)entity.Key).Activate();

}

}

}

В этом методе происходит проверка на взаимодействие Джанго с игровыми объектами. Вначале мы получаем объекты, с которыми он взаимодействует, а далее, в зависимости от того, чем является этот объект, обрабатываем столкновение

3)

protected void Animation(GameTime gameTime, int firstFrame, int lastFrame)

{

timerMove += TimeSpan.FromMilliseconds(gameTime.ElapsedGameTime.Milliseconds);

if (timerMove > timeForOneFrame)

{

timerMove = TimeSpan.Zero;

if ((currentFrame.X >= lastFrame) || (currentFrame.X < firstFrame) || IsInAir)

//в воздухе анимации не будет

{

currentFrame.X = firstFrame;

}

else

{

currentFrame.X++;

}

}

}

Этот метод позволяет сделать так, чтобы при движении персонажей, игрок видел анимацию. Смена кадров производится через определенное время, выделенное для одного кадра.

Руководство пользователя

Цель игры: пройти все уровни, убить всех врагов и остаться при этом живым

Управление персонажем:

**Стрелка вверх** – прыжок;

**Стрелка влево** – передвижение влево;

**Стрелка вправо** – передвижение вправо;

**F** – выстрелить из пистолета;

**R** **–**перезарядить пистолет

**G**—бросить гранату

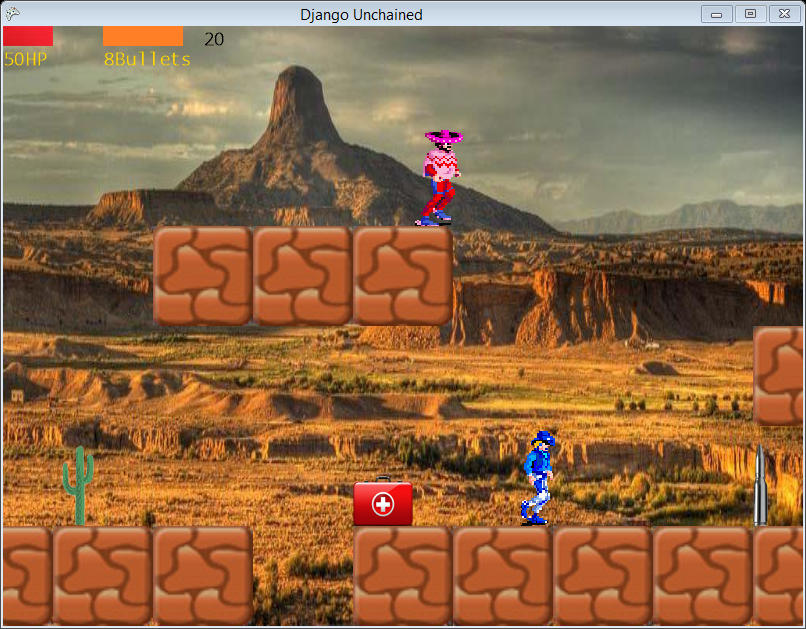
Карта представляет собой полотно с разрешением, которое игрок может настроить по своему усмотрению в игровом меню.

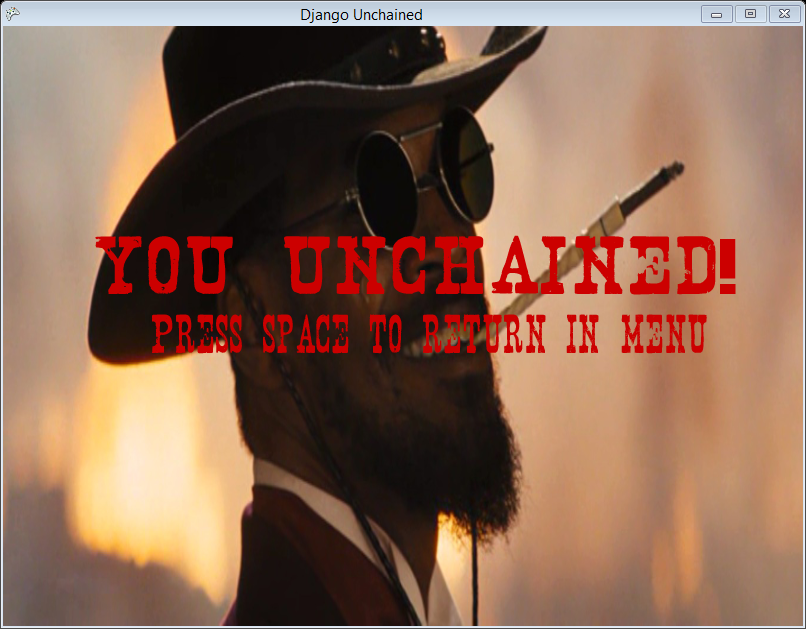
Объекты на карте:

* + 1. Джанго
    2. Враги
    3. Платформы
    4. Бонусы на пули, здоровье, скорость
    5. Кактусы
    6. Пороховые бочки
    7. Мины
    8. Пули (появляются после выстрела Джанго или врага)
    9. Гранаты (появляются после того, как Джанго их бросит)

Скриншоты из игры





После прохождения игры появится игрок увидит на экране: 

В случае смерти:



Для того, чтобы научиться пользоваться этой игрой, начинающему игроку достаточно этого руководство. Более опытному не составит труда разобраться в управлении и без руководства

Заключение

В результате выполнения курсового проекта был разработан простая аркадная игра—Django Unchained

Игра очень проста в управлении.

В процессе создания проекта были изучены основные концепции ООП, работа с файлами, базовый синтаксис языка программирования C# и основы платформы XNA Framework

Список использованных источников

1. Материалы ресурса Wikipedia – Свободная Энциклопедия. ([http://wikipedia.org](http://wikipedia.org/)).
2. Горнаков С. Г. Программирование компьютерных игр под Windows в XNA Game Studio Express. – М.: ДМК Пресс, 2008. – 384 с.: ил.
3. Шилдт, Герберт. Полный справочник по C#. : Пер. с англ. — М. : Издательский дом “Вильямс”, 2004. — 752 с. : ил. — Парал. тит. англ.
4. Троелсен, Эндрю. Язык программирования С# 2010 и платформа .NET 4.0, 5-е изд. : Пер. с англ. — М. : ООО "И.Д. Вильяме", 2011. — 1392 с. : ил. — Парал. тит. англ.