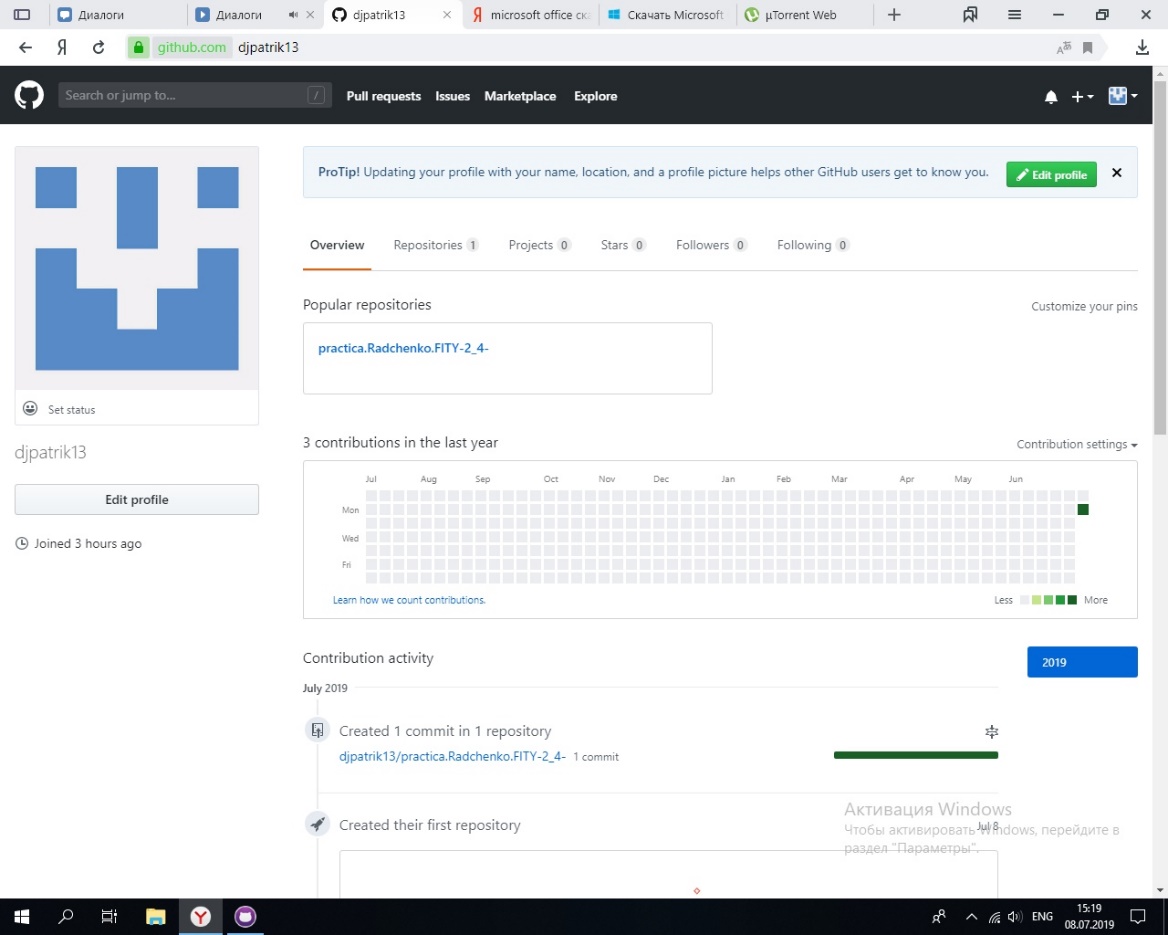
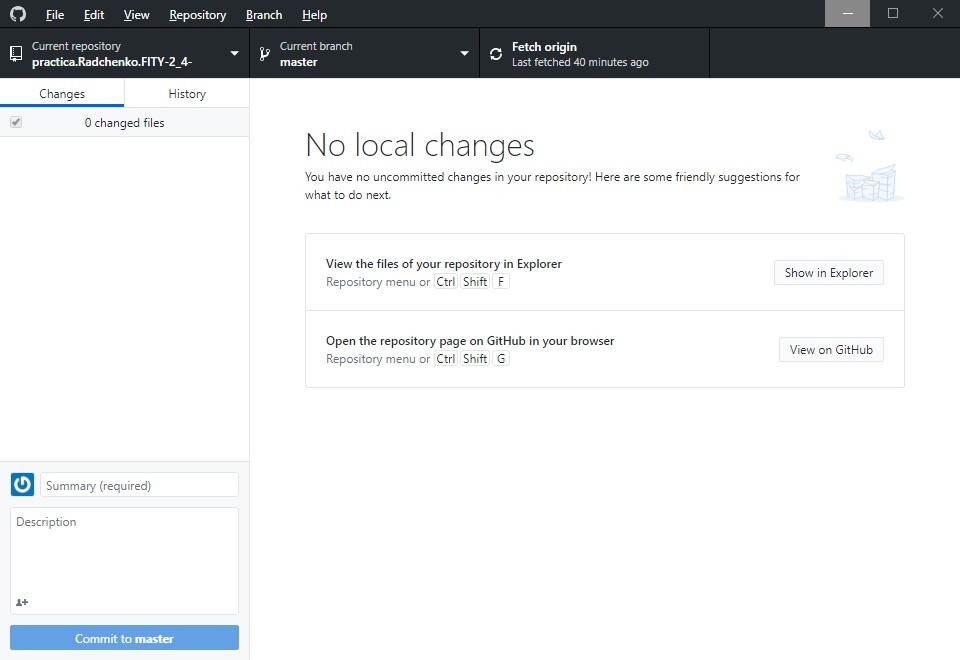
|  |
| --- |
| **МИНИСТЕРСТВО науки и высшего образования  РОССИЙСКОЙ ФЕДЕРАЦИИ**  **федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Южно-Российский государственный**  **политехнический университет (НПИ) имени М.И. Платова»**  ===============================================================  «УТВЕРЖДАЮ»  Заведующий кафедрой ИСТ  \_\_\_\_\_\_\_\_\_\_\_\_\_Д.В. Гринченков  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г.    Факультет **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Кафедра **ИСТ**  Направление *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  **ЗАДАНИЕ**  **на \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ практику**  вид практики  Студенту *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  (ФИО) (курс, группа)   1. Задание на практику: *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  2. Сроки практики: с «\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г. по «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 20\_\_ г.  3. Защита отчета по практике: «\_\_\_» \_\_\_\_\_\_\_\_\_ 20\_\_ г.  Руководитель практики от университета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)  Задание к выполнению принял студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)  Дата выдачи задания «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ г. |

|  |
| --- |
| **МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**  **РОССИЙСКОЙ ФЕДЕРАЦИИ**  **федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Южно-Российский государственный**  **политехнический университет (НПИ) имени М.И. Платова»**  Факультет \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Кафедра ИСТ  Направление \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **ОТЧЕТ**  **о \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ практике**  вид практики  **на \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  название предприятия, организации  **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  Выполнил студент \_\_\_ курса \_\_\_ группы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Фамилия И. О.  Руководитель практики от предприятия Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_м.п.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_20\_ \_г \_\_\_\_\_\_\_\_\_  должность, степень, звание Фамилия И. О. дата подпись  Руководитель практики от университета Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_20\_ \_г \_\_\_\_\_\_\_\_\_  должность, степень, звание Фамилия И. О. дата подпись  **Новочеркасск 2019 г** |

1.Регистрация на <https://github.com>



2.Установка клиента GitHub Deskop



3.Краткая теория

Git — это набор консольных утилит, которые отслеживают и фиксируют изменения в файлах (чаще всего речь идет об исходном коде программ, но вы можете использовать его для любых файлов на ваш вкус). С его помощью вы можете откатиться на более старую версию вашего проекта, сравнивать, анализировать, сливать изменения и многое другое. Этот процесс называется контролем версий. Существуют различные системы для контроля версий.

Git является распределенным, то есть не зависит от одного центрального сервера, на котором хранятся файлы. Вместо этого он работает полностью локально, сохраняя данные в папках на жестком диске, которые называются репозиторием. Тем не менее, вы можете хранить копию репозитория онлайн, это сильно облегчает работу над одним проектом для нескольких людей. Для этого используются сайты вроде github и bitbucket.

Создание нового репозитория

Как мы отметили ранее, git хранит свои файлы и историю прямо в папке проекта. Чтобы создать новый репозиторий, нам нужно открыть терминал, зайти в папку нашего проекта и выполнить команду init. Это включит приложение в этой конкретной папке и создаст скрытую директорию .git, где будет храниться история репозитория и настройки.  
Создайте на рабочем столе папку под названием git\_exercise. Для этого в окне терминала введите:



|  |  |
| --- | --- |
| 1  2  3 | $ mkdir Desktop/git\_exercise/  $ cd Desktop/git\_exercise/  $ git init |

Командная строка должна вернуть что-то вроде:



|  |  |
| --- | --- |
| 1 | Initialized empty Git repository in /home/user/Desktop/git\_exercise/.git/ |

Это значит, что наш репозиторий был успешно создан, но пока что пуст. Теперь создайте текстовый файл под названием hello.txt и сохраните его в директории git\_exercise.

Подготовка файлов

В git есть концепция области подготовленных файлов. Можно представить ее как холст, на который наносят изменения, которые нужны в коммите. Сперва он пустой, но затем мы добавляем на него файлы (или части файлов, или даже одиночные строчки) командой add и, наконец, коммитим все нужное в репозиторий (создаем слепок нужного нам состояния) командой commit.  
В нашем случае у нас только один файл, так что добавим его:



|  |  |
| --- | --- |
| 1 | $ git add hello.txt |

Если нам нужно добавить все, что находится в директории, мы можем использовать



|  |  |
| --- | --- |
| 1 | $ git add -A |

Проверим статус снова, на этот раз мы должны получить другой ответ:



|  |  |
| --- | --- |
| 1  2  3  4  5  6 | $ git status  On branch master  Initial commit  Changes to be committed:  (use "git rm --cached ..." to unstage)  new file: hello.txt |

Файл готов к коммиту. Сообщение о состоянии также говорит нам о том, какие изменения относительно файла были проведены в области подготовки — в данном случае это новый файл, но файлы могут быть модифицированы или удалены.

Коммит(фиксация изменений)

Коммит представляет собой состояние репозитория в определенный момент времени. Это похоже на снапшот, к которому мы можем вернуться и увидеть состояние объектов на определенный момент времени.  
Чтобы зафиксировать изменения, нам нужно хотя бы одно изменение в области подготовки (мы только что создали его при помощи git add), после которого мы может коммитить:



|  |  |
| --- | --- |
| 1 | $ git commit -m "Initial commit." |

Эта команда создаст новый коммит со всеми изменениями из области подготовки (добавление файла hello.txt). Ключ -m и сообщение «Initial commit.» — это созданное пользователем описание всех изменений, включенных в коммит. Считается хорошей практикой делать коммиты часто и всегда писать содержательные комментарии.

Политический симулятор (англ. political simulation) или симулятор правительства (англ. government simulation) — жанр игр или компьютерных игр, которые симулируют правительство и политику некоторой страны или какой-то её части. В этих играх может присутствовать геополитическая составляющая (в том числе формирование и проведение внешней политики), создание внутренних политических правил или проведение политических кампаний. Игры в этом жанре отличаются от варгеймов сниженной ролью от военных или экшн-элементов или представление их в абстрактном виде.

Содержание

1 Настольные игры

2 Компьютерные игры

3 Интернет-игры

4 Симуляция политики в других жанрах

5 Образование

Настольные игры

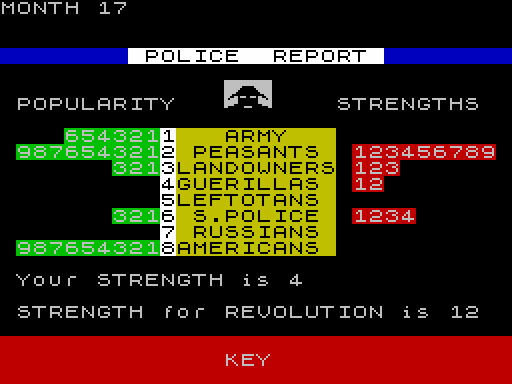
Игры, основанные на политических выборах или геополитике, существовали задолго до появления персональных компьютеров и их возможности быстрой обработки больши́х объёмов статистических данных. Одной из таких ранних игр является The Game of Politics, созданная Освальдом Лордом (англ. Oswald Lord) в 1935 году, которая оставалась популярной до 1960-х годов. В 1954 году создана настольная игра «Дипломатия», которая, в отличие от других варгеймов, содержит фазу «переговоров». В этой фазе игроки могут договариваться друг с другом, чтобы впоследствии проводить военные операции одновременно. Национальная политика оставалась важной частью настольных игр, что обусловило создание игр, похожих на Die Macher (выпущена в 1986 году), основанной на выборах в Германии, или Wreck the Nation, которая высмеивает политику США времен президентства Джорджа Буша.

После долгих лет существования в форме игры по переписке, «Дипломатия» стала одной из первых игр, использовавших преимущества электронной почты. По состоянию на 2007 год «Дипломатия» остается популярной PBEM-игрой.

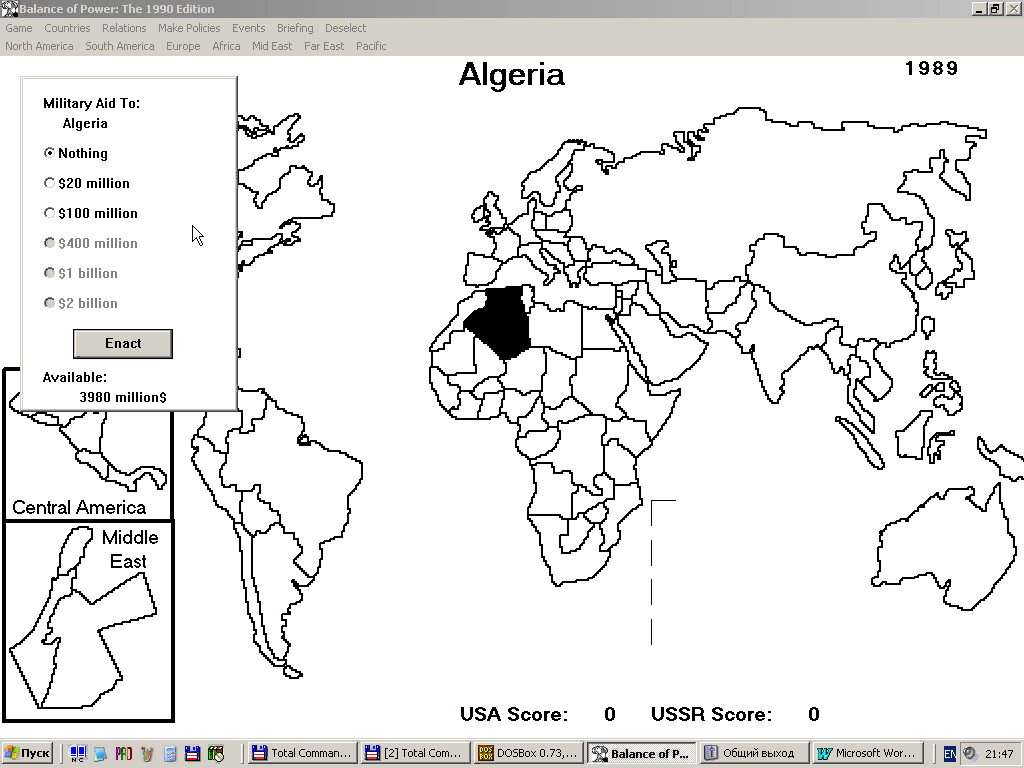
Компьютерные игры

Компьютерная игра Dictator — игровые фракции и их состояния в отчёте тайной полиции.

По мере того, как компьютерная техника усложнялась, компьютерные игры в жанре стали переходить от электронной почты к более сложной симуляции. Для многих европейских пользователей первой получившей известность игрой в жанре политических симуляторов является Dictator. Это игра издана в 1983 году компанией dK’Tronics для компьютеров ZX Spectrum.

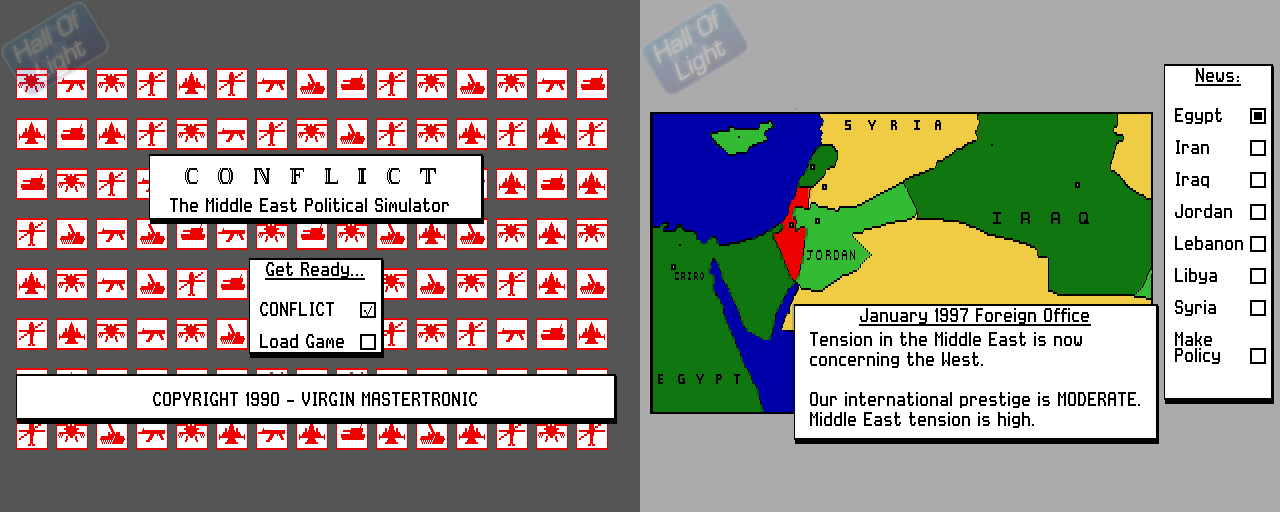


Одной из первых в жанре стала игра Balance of Power, разработанная Крисом Кроуфордом (англ. Chris Crawford) и изданная в 1985 году. В этой игре представлен конфликт на пике холодной войны, для разрешения которого следует использовать политические методы вместо военной силы. В Balance of Power любой вооруженный конфликт между игроком и другой сверхдержавой приводит к ядерной войне, которая является одним из условий поражения.



В игры «эпохи холодной войны» входят также Conflict: Middle East Political Simulator от Virgin Interactive, Crisis in the Kremlin от Spectrum HoloByte и малоизвестная игра Hidden Agenda, вышедшая в 1988 году.

Игра Conflict симулирует гипотетическую ситуацию в 1997 году, где игрок принимает роль израильского премьер-министра и обязан, применяя различные дипломатические и секретные указания, достичь победы над соперничающими народами. Игрок, окруженный враждебными народами, сильно ограничен в военной силе и потому должен придерживаться мирных средств, чтобы удержаться у власти до получения более совершенных вооружений и военной силы.



В игре Crisis in the Kremlin игрок принимает роль протеже одного из следующих советских политиков: Михаил Горбачёв из фракции реформистов, Егор Лигачёв — лидер фракции «твердой линии», и Борис Ельцин — одна из выделявшихся фигур фракции националистов. Игрок может воспользоваться симуляцией для изучения определённых стратегий, которые могли бы либо привести СССР в эпоху роскоши, либо ускорить его распад и интеграцию в новый мировой порядок. В этой игре представлена концепция управления бюджетом, удовлетворенность граждан и фракций, а также множество экономических аспектов и широкий политический спектр.



В Hidden Agenda игрок принимает роль президента государства Химерика (англ. Chimerica) из Центральной Америки, пережившего революцию. Игрок должен жонглировать международными отношениями и потребностями населения страны.

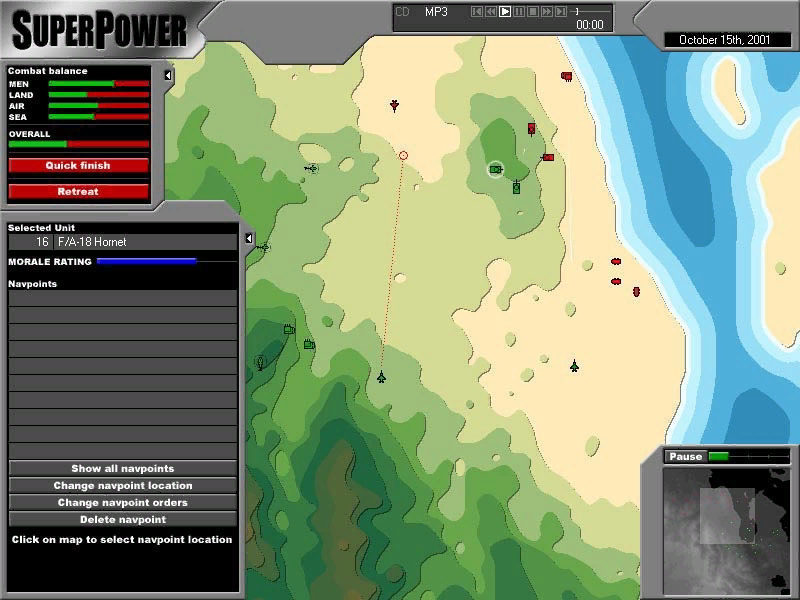


Ранние политические симуляторы преследовали скорее образовательные цели, нежели развлекательные. Так в 1987 году в Университете Кента (англ. Kent State University) была разработана утилита On the Campaign Trail как часть университетской программы политического управления. С помощью этой программы студенты привлекались к принятию решений, касающегося избирательных кампаний в Сенат США с 1970 по 1986 годы.

Игра Power Politics, выпущенная в 1992 году (как и игра President Elect 1981-го года) посвящена внутренним политическим кампаниям в США, но не управлению страной после выборов. В 1996 году эта концепция была использована в игре Doonesbury Election Game, разработанной автором Power Politics Рэнди Чейзом (англ. Randy Chase) и изданной Mindscape. В ней игрок проводит политическую кампанию с помощью ряда советников, которые представлены некоторыми персонажами из комикса «Дунсбери». Следующая игра серии была названа Power Politics III и вышла в 2005 году. В 2004 году компания Stardock выпустила игру Political Machine. В ней игрок управляет избирательной кампанией на должность президента США длительностью в 41 неделю, участвуя в разговорных политических шоу и вырабатывая содержание речей. Игра симулирует современные способы проведения опросов, предлагая обратную связь в реальном времени для понимания того, как проведение кампании влияет на результаты опросов. В 2006 году компания TheorySpark выпустила игру President Forever 2008 + Primaries. Игра представляет собой симулятор избирательной кампании, позволяющий игроку реалистично управлять всей кампанией как в части праймериз, так и в части основных выборов. Игра сама по себе представляет продолжение весьма успешного симулятора основных выборов President Forever, вышедшего в 2004 году.

В некоторых играх жанра для возможности повлиять на избирателей имеются функции введения новых политик или изменений бюджета. Одной из таких игр является Democracy, изданная в 2005 году компанией Positech Games. В ней игрок делает в течение каждого хода выбор, какие политики следует поддержать. С завершением хода игрок может видеть как изменяется его рейтинг среди различных типов избирателей. Перед началом избирательной кампании кандидаты могут давать те или иные обещания. В случае, если они не смогут эти обещания выполнить, на следующих выборах поддержка этих кандидатов будет меньше.

Также существуют игры, в которых игрок принимает роль руководителя государства, например, SuperPower и её сиквел SuperPower 2. Целью игр является создание экономической стабильности и процветания, однако игра в целом связана с внешней политикой и набором разных способов взаимодействия с другими странами. В этих играх присутствует большое количество существующих в реальном мире договоров, влияющих на политику разных стран.



SuperPower 1



SuperPower 2

Интернет-игры

Игры, основанные на возможностях интернета, наподобие NationStates, позволяют игроку влиять на принятие каждодневных решений определённого правительства в условиях конкуренции с другими нациями. Игры с менее формализованной структурой могут проводиться на веб-форумах. В них игроки управляют правительствами и народами на основе набора заранее определённых правил. В этих играх проводится несколько раундов симуляции политики выбранной страны в разных исторических периодах. Не все подобные игры имеют масштаб государства. Они могут иметь и международный «охват», тогда как масштаб других может быть государственный или местный. Игроки в подобных играх принимают роль вымышленных политиков и участвуют в дебатах, работе с прессой и в симуляции выборов. Реализм в этих играх крайне высок, поскольку ключевые темы каждого дня обсуждаются и озвучиваются самими игроками и администраторами. При этом администраторы могут произвольно влиять на игровой мир.

В других политических интернет-играх после регистрации игрок подает заявку на одну из вакантных должностей (представитель целой страны или должностное лицо внутри страны, например, политик или генерал армии) и выполняет поставленные перед ним задачи либо посредством издания указов, публикования заявлений в газетах и т. п., либо (что чаще встречается) посредством мастера игры. В этих играх обычно имеется сильный уклон в сторону реализма и кооперации. Примерами подобных игр могут быть United States Government Simulation и Virtual Government Simulation. В этих играх после регистрации игрок создает своего персонажа и занимает место в Конгрессе США вместе другими игроками. Игрок может создавать проекты законов и высказывать предложения по ним, добиваясь их подписания и включения в действующее законодательство, увеличивать собственное финансирование и финансирование своей политической партии, получать и сохранять руководящие позиции в Конгрессе для себя или своей партии, или баллотироваться на более высокие посты вплоть до поста президента США.

Симуляция политики в других жанрах

Lincity — градостроительный симулятор, в котором используются элементы жанра политических симуляторов.

Управление правительством необходимо во многих других симуляторах строительства и управления. Например, в градостроительных симуляторах серии SimCity от компании Maxis происходит симуляция должности мэра города. В SimCity представлено существующее в реальном времени окружение, в котором игрок может выделять зоны для развития города, строить дороги, объекты энергетики и водной инфраструктуры и наблюдать за развитием города, происходящим на основе принятых им решений. Оригинальная версия SimCity была выпущена в 1989 году, а в 2013 году вышла пятая часть серии.

Часто в стратегических играх игрок должен решать проблемы с управлением. В играх жанра 4X необходимо управлять правительством, и это независимо от сеттинга. В решаемые задачи входит, например, строительство инфраструктуры или осуществление торговли. В игре Galactic Civilizations II игроку требуется поддерживать определённый рейтинг одобрения, чтобы его партия оставалась у власти. Иногда внутренняя политика существует в более абстрактной форме, нежели международные конфликты. Например, в играх серии Civilization игрок получает полный контроль над ресурсами, и при этом радикальное переустройство цивилизации выполняется простым действием игрока.

В разных стратегических играх управление правительством используется в разной степени. Так, в играх серии Hearts of Iron (сеттинг — Вторая мировая война) гражданское население представлено в виде факторов партизанства и трудового ресурса. В игре Victoria: An Empire Under the Sun игрок должен не только производить завоевания, но и осуществить вторую промышленную революцию, не взирая на имевшие место реальные политические революции типа революций 1848—1849 годов или коммунистической революции.

Правительства и политика часто представлены в играх жанра квест. В игре A Mind Forever Voyaging в жанре текстового квеста, выпущенной в 1985 году компанией Infocom, игрок получает контроль над разумным компьютером, позволяющим ставить эксперименты над сценариями возможного будущего, основываясь на изменениях решений публичной политики. Журнал Newsweek так прокомментировал игру: «Это не „1984“, но, в каком-то смысле, это даже страшнее».

В вышедшей в 2008 году игре Spore представлен этап «цивилизации», на котором игрок может управлять транспортными средствами и взаимодействовать с другими городами, пока под его влиянием не окажется 12 городов.

Образование

Помимо развлечения эти игры имеют практическое применение для тренировки и обучения персонала правительств. Созданы тренировочные симуляции для таких задач как управление политикой правоохранительных органов (например, для расового профилирования), симуляция карьеры военного офицера, симуляция действий больничного учреждения в экстренной ситуации и т. п.

**1. Назначение разработки**

Практическая работа является комплексным проектом, охватывающим различные аспекты разработки ПО;

содержит все основные аспекты компьютерной ролевой игры;

является продуктом сферы компьютерных развлечений.

**2. Требования к программе или программному изделию**

**2.1 Требования к функциональным характеристикам**

Данный проект является компьютерной игрой, вследствие чего предусматривается одна категория пользователей - игроки. В процессе работы приложения пользователь является непосредственным участник игрового процесса и оказывает непосредственное влияние на него.

Программы должна обладать следующим функционалом:

а) графический функционал:

б) звуковой функционал:

в) внутриигровой функционал:

г) интерфейс пользователя:

**2.2 Требования к входным и выходным данным**

Входными данными в компьютерной игре являются игровые настройки пользователя, а также непосредственное управление во время игрового процесса с помощью компьютерной мыши и клавиатуры. Проект относится к играм в реальном времени, где в отличие от пошаговых - действия игрока незамедлительно оказывают влияние на игровой процесс.

Выходными данными являются графическая интерпретация игрового процесса на мониторе игрока и звук, сопровождающий его. Действия игрока влияют на игровой процесс и текущее состояние игровой сцены. Игрок контролирует игрового персонажа с помощью интерфейса пользователя.

**2.3. Требования к надежности**

В программе должна присутствовать проверка входной информации на соответствие типов, принадлежность диапазону допустимых значений и соответствие структурной корректности. В случае возникновения ошибок предусмотреть возможность вывода информативных диагностических сообщений. В программе реализована система логов, а также возможность анализа дампа приложения в случае некорректного завершения.

**2.4. Требования к составу и параметрам технических средств**

Минимальные системные требования:

– ОС (операционная система): Windows XP/Vista/7/8;

– Процессор: Intel Core 2 Duo @ 3.0 Ghz / AMD Athlon 64 X2 6000+;

– Оперативная память: 1 Gb;

– Жесткий диск: 10 Gb свободно;

– Видео память: 512 Mb;

– Видео карта: nVidia GeForce 9800 / AMD Radeon HD 4870;

– Звуковая карта: Совместимая с DirectX;

– DirectX 9.0c;

– Клавиатура, Мышь.

Рекомендуемые системные требования:

- ОС (операционная система): Windows Vista/7/8;

- Процессор: Intel Core i5 @ 3.2 GHz / AMD Phenom II X4 @ 3.6 GHz;

- Оперативная память: 2 Gb;

- Жесткий диск: 10 Gb свободно;

- Видео память: 1 Gb;

- Видео карта: nVidia GeForce GTX 460 / AMD Radeon HD 5850;

- Звуковая карта: Совместимая с DirectX;

- DirectX 11;

- Клавиатура, Мышь.

**2.5. Требования к информационной и программной совместимости**

Программа должна функционировать под управлением ОС семейства Windows следующих версий: Windows XP, Vista, 7, 8. Также требуется установленный DirectX 9.0c или более поздней версии.

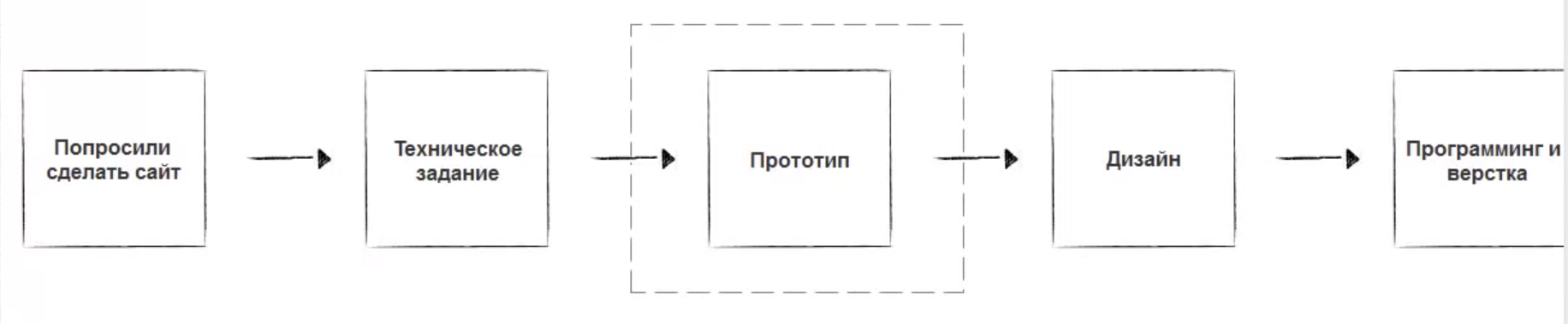
**2.6. Требования к программной документации**

Программная документация должна быть представлена руководством пользователя.

**Прототипирование для менеджеров: зачем это нужно и какие бывают прототипы.**

Прототипирование — это создание наброска, схемы или готового макета пользовательского интерфейса. Прототипы позволяют сохранить время и деньги, которые были бы потрачены на разработку непроверенных решений.

Уметь делать прототипы полезно бизнес-аналитикам, маркетологам, продукт-менеджерам, дизайнерам и тем, кто работает в распределённой команде



## **Для чего нужно прототипирование.**

* Выявление и написание требований к функционалу, интерфейсу, веб-сайту.
* Управление ожиданиями заказчика.
* Иллюстрирование user stories или use cases для того, чтобы они стали более понятными (в том числе и для самого аналитика).
* Создание собственно дизайна, когда в команде нет дизайнера (редко).
* Утверждение основных блоков и расположения с заказчиком.
* Передача требований дизайнеру.
* Тестирование расположения блоков, кнопок, да и валидация идей.

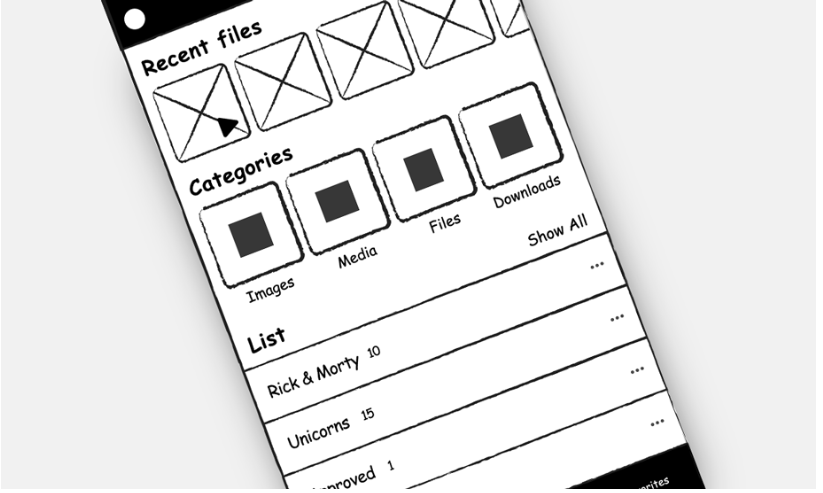
Утверждение смысловых блоков на странице/экране на этапе прототипирования существенно экономит время и деньги на разработку, избавляет от ошибок на начальном этапе.

Визуальное представление, понимание, оценка и изучение идей также лежат в основе методологии Design thinking.

## **Какие бывают прототипы.**

Так как чаще всего мы работаем с иностранными заказчиками, то и термины будут звучать на английском: wireframe и prototype (low-fidelity prototype, high-fidelity prototype).

**Wireframe.**



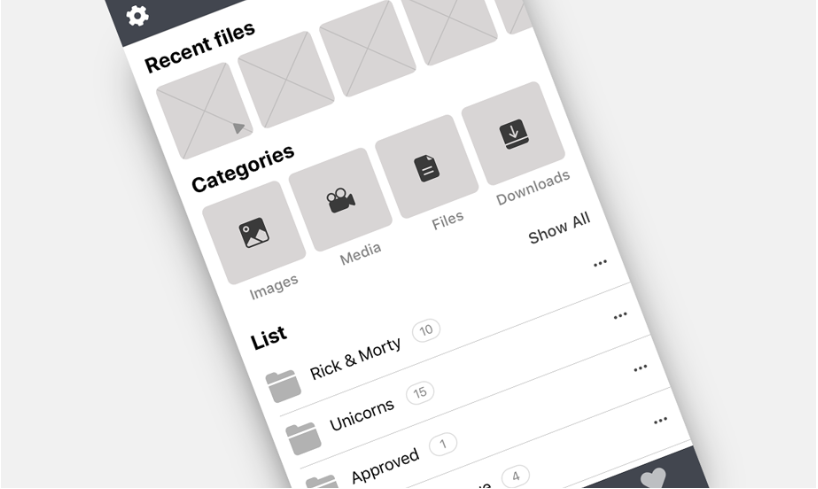
Начальный вид прототипирования. Черновой набросок от руки на бумаге. Некоторые приложения, например [Balsamiq](https://balsamiq.com/products/" \t "_blank), специально имитируют стиль небрежного рисунка, чтобы не фокусироваться на внешнем виде. Такие «наброски» лишь обозначают блоки на странице. Что очень удобно и не отвлекает внимания на самом первом этапе обсуждения. Для wireframes лучше не использовать цвет — это будет отвлекать от сути и совещание пройдёт в обсуждении цветов кнопки. Оставьте цвет дизайнерам для создания прототипов высокой детализации.

Что даёт wireframe:

* предметное отображение функционала;
* описание доступных функций;
* определение правил отображения разных видов информации;
* описание различных сценариев.

Кто делает: менеджеры, бизнес-аналитики, маркетологи, дизайнеры и UX-дизайнеры.

**Low-fidelity prototype (прототип малой детализации).**



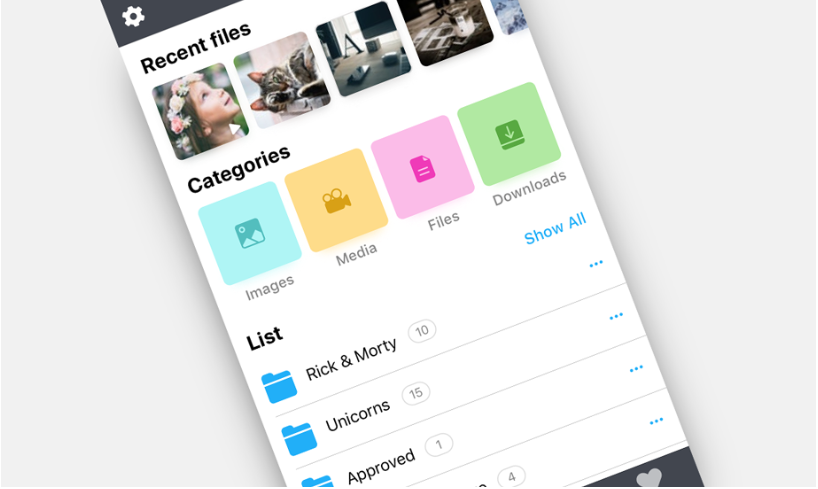
Такой прототип уже отличается от наброска от руки. Расположение элементов проработано, размеры блоков, шрифтов примерно определены. Элементы выровнены друг относительно друга и по сетке. Иерархия блоков определяется оттенками одного цвета, чаще серого.

Прототип малой детализации чаще всего применяется для окончательного утверждения расположения функциональных блоков с заказчиком или другим ответственным лицом (product owner). При создании переходов по кнопкам и ссылкам между страницами используется для пользовательского тестирования. Создать готовый кликабельный прототип такого типа можно в Axure. Отдельно переходы можно сделать в Invision или Marvel.

Если хотите сделать прототип более привлекательным для клиента, но вы не Web- или UI-дизайнер, сделайте равные расстояния между блоками и определите иерархию размеров шрифта (заголовки 1, 2, 3 уровней и основной текст, ссылки).

Кто делает: менеджеры, бизнес-аналитики, маркетологи, дизайнеры и UX-дизайнеры.

**High-fidelity prototype (прототип высокой детализации).**



Этот вид прототипа уже выглядит как готовый продукт с подобранным стилем и pixel perfect элементами. Он тестируется на соответствие требованиям по доступности для людей с разными видами нарушения восприятия цветового диапазона. Утверждается с заказчиком и передаётся в разработку.

Кто делает: только дизайнеры.

**Выводы.**

Любой из представленных видов прототипа можно (и нужно) сделать кликабельным и протестировать на ваших потенциальных пользователях. Для этого есть такие приложения как Balsamiq, Axure, InVision, Marvel и многие другие.

Расскажу об одном продукте, который отправили в продакшн, минуя стадию тестирования. Приложение было предустановлено на устройствах, но люди им не пользовались. Тестирование показало, что большинство респондентов не поняли, что с ним делать. В чём была проблема? Прототипы не были протестированы на реальных людях. Что это значит? Работу пришлось делать дважды, но теперь правильно. Представьте себе рейт минимально необходимой команды из дизайнера, разработчика, тестировщика и менеджера.

Делайте прототипы: экономьте ваше время, деньги проекта и нервы ваших пользователей :)

Laravel — бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (англ. Model View Controller — модель-представление-контроллер). Laravel выпущен под лицензией MIT.

Исходный код проекта размещается на GitHub. В результате опроса sitepoint.com в декабре 2013 года о самых популярных PHP-фреймворках Laravel занял место самого многообещающего проекта на 2014 год.

В 2015 году в результате опроса sitepoint.com по использованию PHP-фреймворков среди программистов занял первое место в номинациях:

Фреймворк корпоративного уровня

Фреймворк для личных проектов

История

Laravel был создан Taylor Otwell как более функциональная альтернатива CodeIgniter, который не предусматривал различные дополнительные функции. Первый бета-релиз Laravel стал доступен 9 июня 2011 года, а Laravel 1 вышел в этом же месяце. Laravel 1 включает в себя встроенную поддержку для аутентификации, локализации, модели, представления, сессий, маршрутизации и других механизмов.

Laravel 2 был выпущен в сентябре 2011 года. Основные новые функции включают в себя поддержку контроллеров, которые сделали фреймворк полностью MVC-совместимым, встроенную поддержку для инверсии управления и систему шаблонов Blade.

Laravel 3 был выпущен в феврале 2012 года с набором новых функций, включая интерфейс командной строки (CLI) под именем "Artisan", встроенную поддержку нескольких систем управления базами данных, миграции баз данных в виде контроля версий, обработку событий. Выпуск Laravel 3 получил значительное увеличение числа пользователей, что повлияло на его популярность.

Laravel 4 был выпущен в мае 2013 года. Были произведены существенные изменения структуры Laravel, перенося ее макет в набор отдельных пакетов, распространяемых через Composer. Такая компоновка улучшила расширяемость Laravel 4, которая сочеталась с официальным регулярным расписанием релиза, охватывающим шесть месяцев между небольшими выпусками. Другие новые функции в выпуске Laravel 4 включают в себя выгрузка таблиц базы данных для первоначальной популяции, поддержку очередей сообщений, встроенную поддержку отправки различных типов электронной почты и поддержку "мягкого" удаления записей базы данных.

Laravel 5 был выпущен в феврале 2015 года в результате внутренних изменений, которые закончились перенумерацией релиза Laravel 4.3 в будущем. Новые возможности в выпуске Laravel 5 включают поддержку планирования периодически выполняемых задач через пакет Scheduler, слой абстракции Flysystem, который позволяет использовать удаленное хранилище так же, как и локальные файловые системы, улучшенную обработку активов пакета через Elixir и упрощенная аутентификация с внешней стороны через дополнительный пакет Socialite. Laravel 5 также представил новую внутреннюю структуру дерева каталогов для разработанных приложений.

Laravel 5.1, выпущенный в июне 2015 года. Является первым выпуском Laravel для долгосрочной поддержки (LTS) с запланированной возможностью исправления ошибок в течение трех лет и исправлений безопасности в течение трех лет.

Laravel 5.3, выпущенный 23 августа 2016 года. Новые функции в 5.3 сосредоточены на улучшении скорости разработки, добавив дополнительные усовершенствования для отдельных задач.

Laravel 5.4, выпущенный 24 января 2017 года. В этом выпуске есть много новых функций, таких как Laravel Dusk, Laravel Mix, Blade Components и Slots, Markdown Emails, автоматические фасады, улучшения маршрута.

Laravel 5.5, выпущенный 30 августа 2017 года. Возвращен пакет Whoops для отладки ошибок. Из нововведений, метод обработки запроса, новые директивы для шаблонизатора Blade, новая команда для миграции баз данных - migrate:fresh .

Возможности

Ключевые особенности, лежащие в основе архитектуры Laravel:

Пакеты (англ. packages) — позволяют создавать и подключать модули в формате Composer к приложению на Laravel. Многие дополнительные возможности уже доступны в виде таких модулей.

Eloquent ORM — реализация шаблона проектирования ActiveRecord на PHP. Позволяет строго определить отношения между объектами базы данных. Стандартный для Laravel построитель запросов Fluent поддерживается ядром Eloquent.

Логика приложения — часть разрабатываемого приложения, объявленная либо при помощи контроллеров, либо маршрутов (функций-замыканий). Синтаксис объявлений похож на синтаксис, используемый в каркасе Sinatra.

Обратная маршрутизация связывает между собой генерируемые приложением ссылки и маршруты, позволяя изменять последние с автоматическим обновлением связанных ссылок. При создании ссылок с помощью именованных маршрутов Laravel автоматически генерирует конечные URL.

REST-контроллеры — дополнительный слой для разделения логики обработки GET- и POST-запросов HTTP.

Автозагрузка классов — механизм автоматической загрузки классов PHP без необходимости подключать файлы их определений в include. Загрузка по требованию предотвращает загрузку ненужных компонентов; загружаются только те из них, которые действительно используются.

Составители представлений (англ. view composers) — блоки кода, которые выполняются при генерации представления (шаблона).

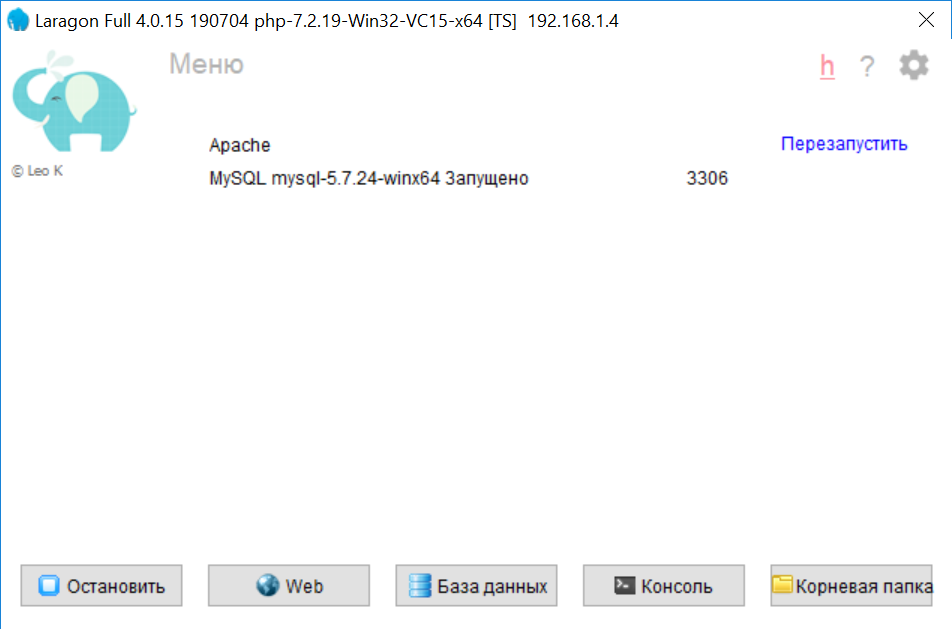
Инверсия управления (англ. Inversion of Control) — позволяет получать экземпляры объектов по принципу обратного управления. Также может использоваться для создания и получения объектов-одиночек (англ. singleton).

Миграции — система управления версиями для баз данных. Позволяет связывать изменения в коде приложения с изменениями, которые требуется внести в структуру БД, что упрощает развёртывание и обновление приложения.

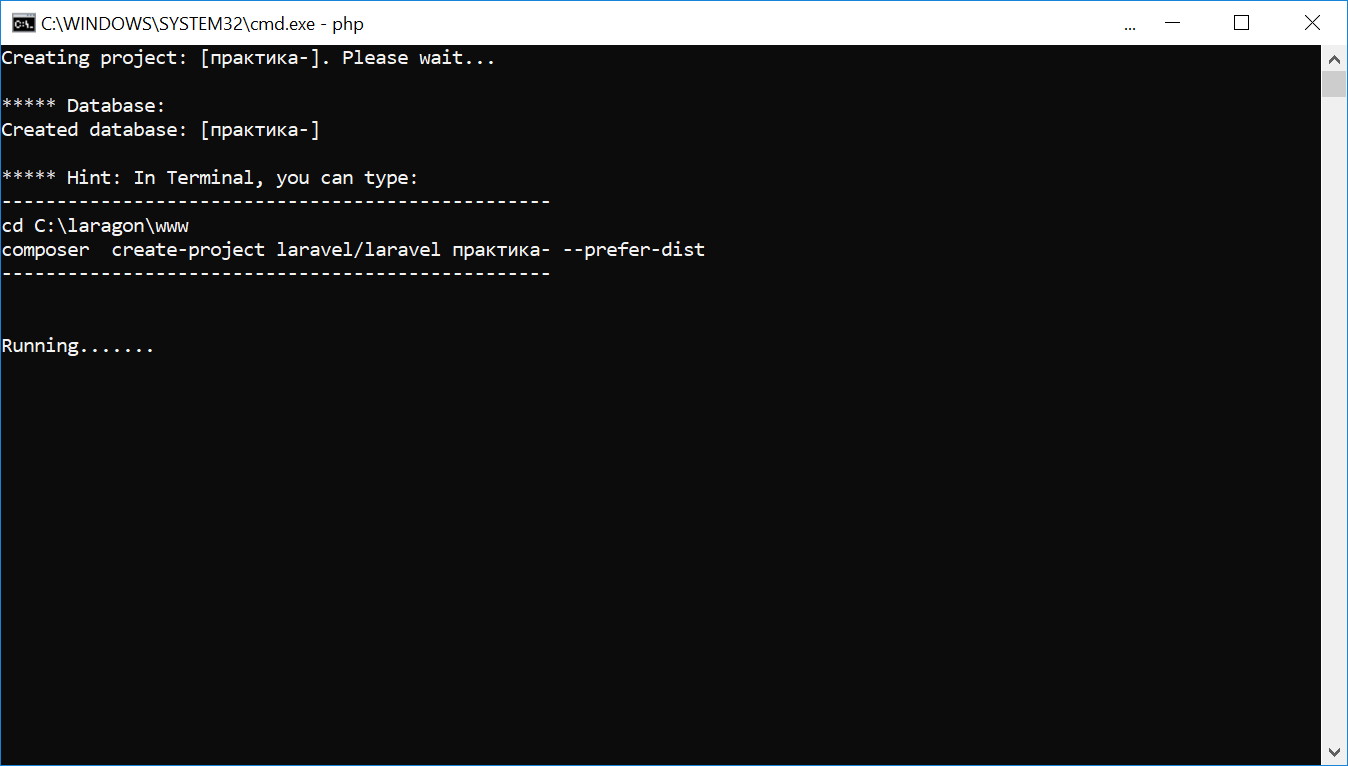
Модульное тестирование (юнит-тесты) — играет очень большую роль в Laravel, который сам по себе содержит большое число тестов для предотвращения регрессий (ошибок вследствие обновления кода или исправления других ошибок).

Страничный вывод (англ. pagination) — упрощает генерацию страниц, заменяя различные способы решения этой задачи единым механизмом, встроенным в Laravel.

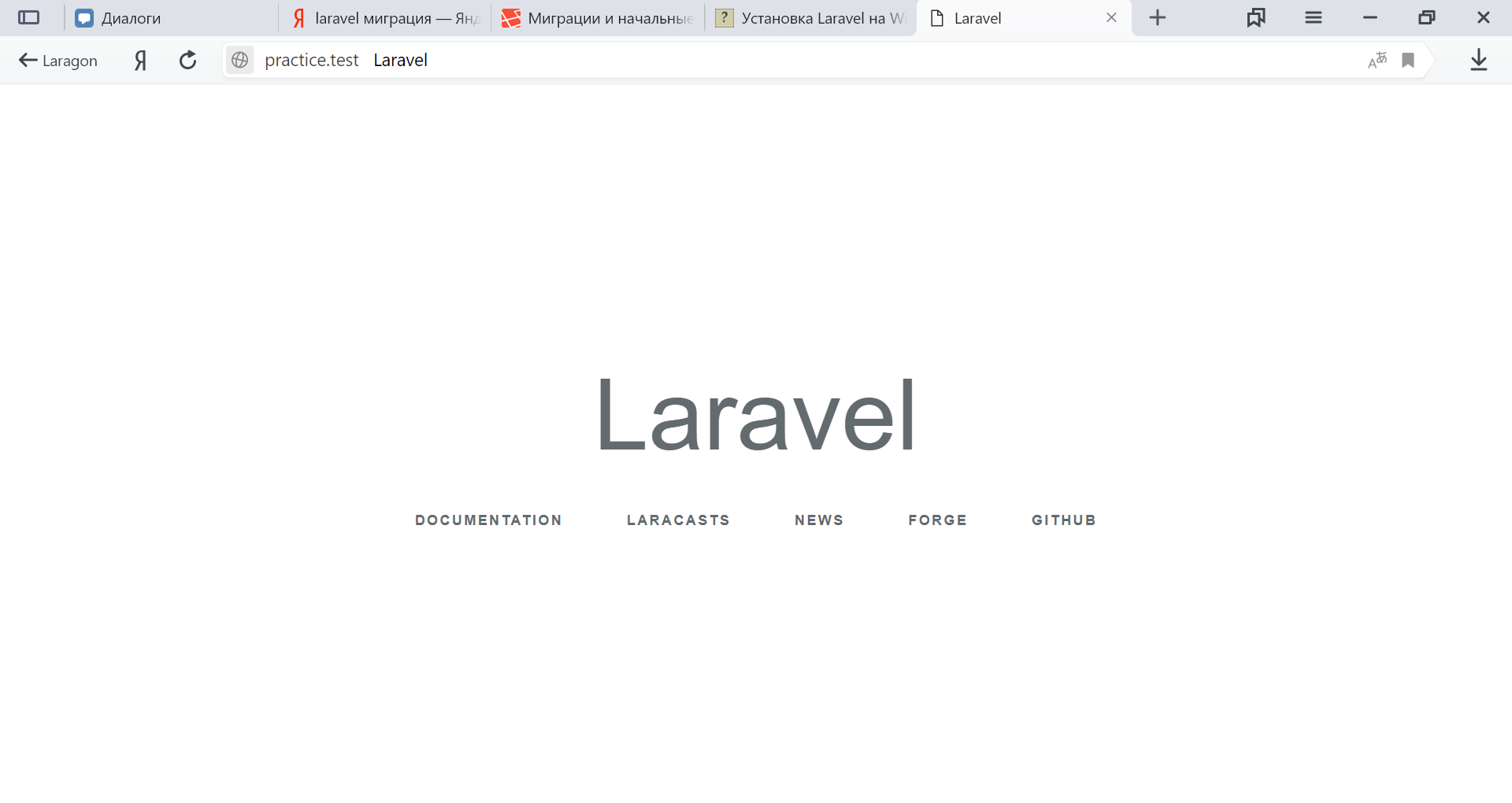
**Установка Laragon**

****

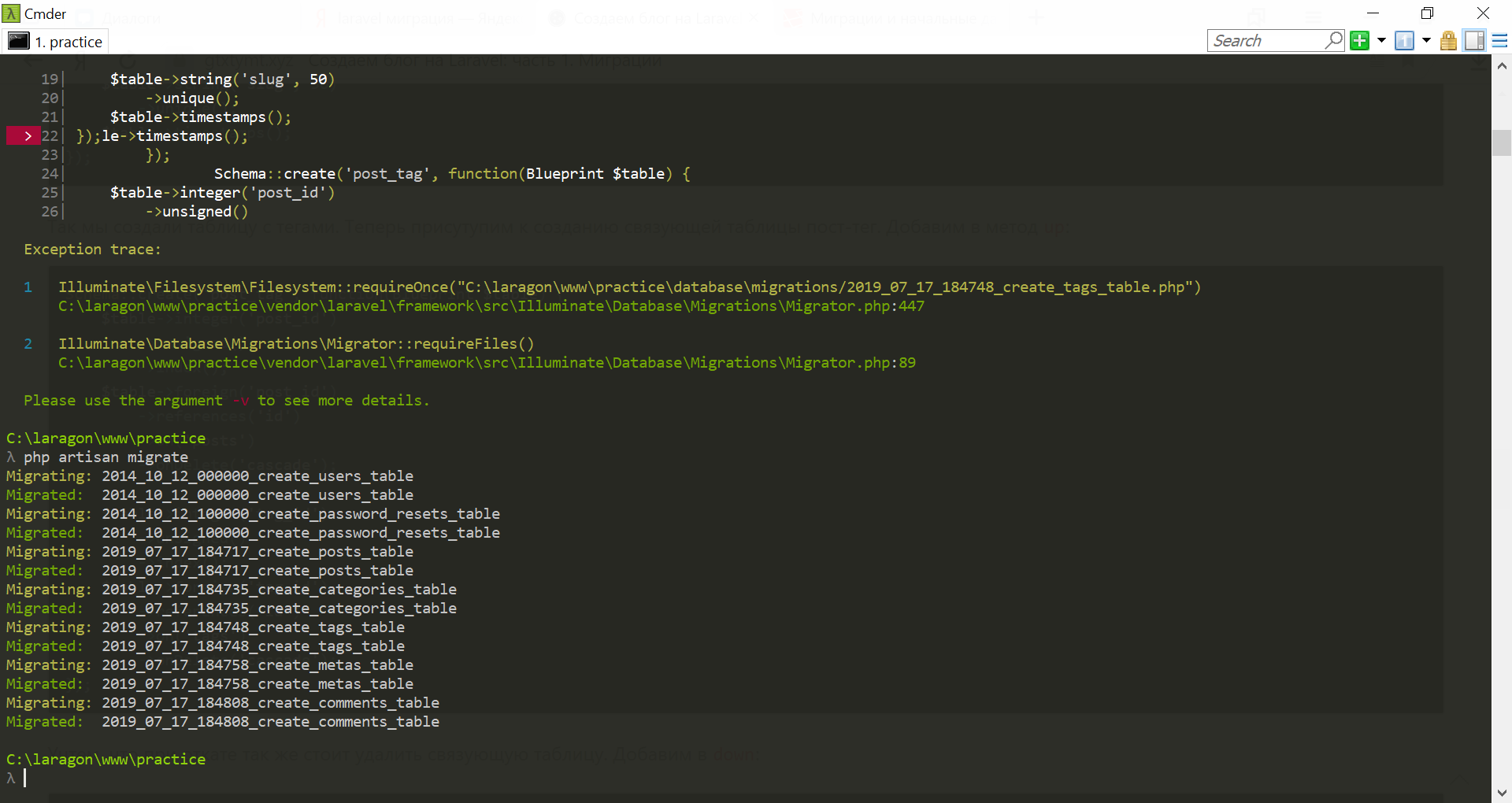
**Lavarel. Создание проекта**

****

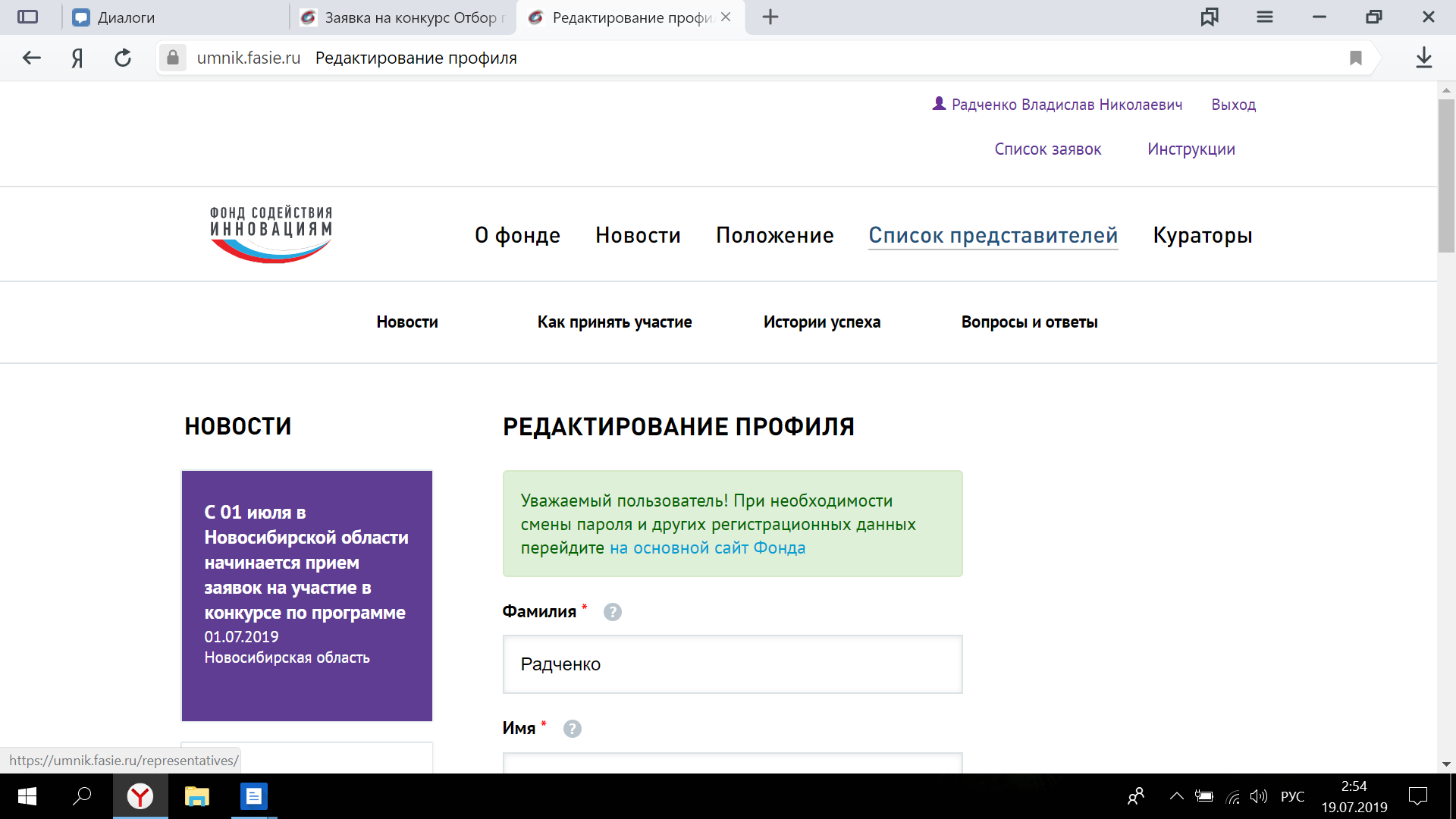
**Лаварел. Браузер.**

****

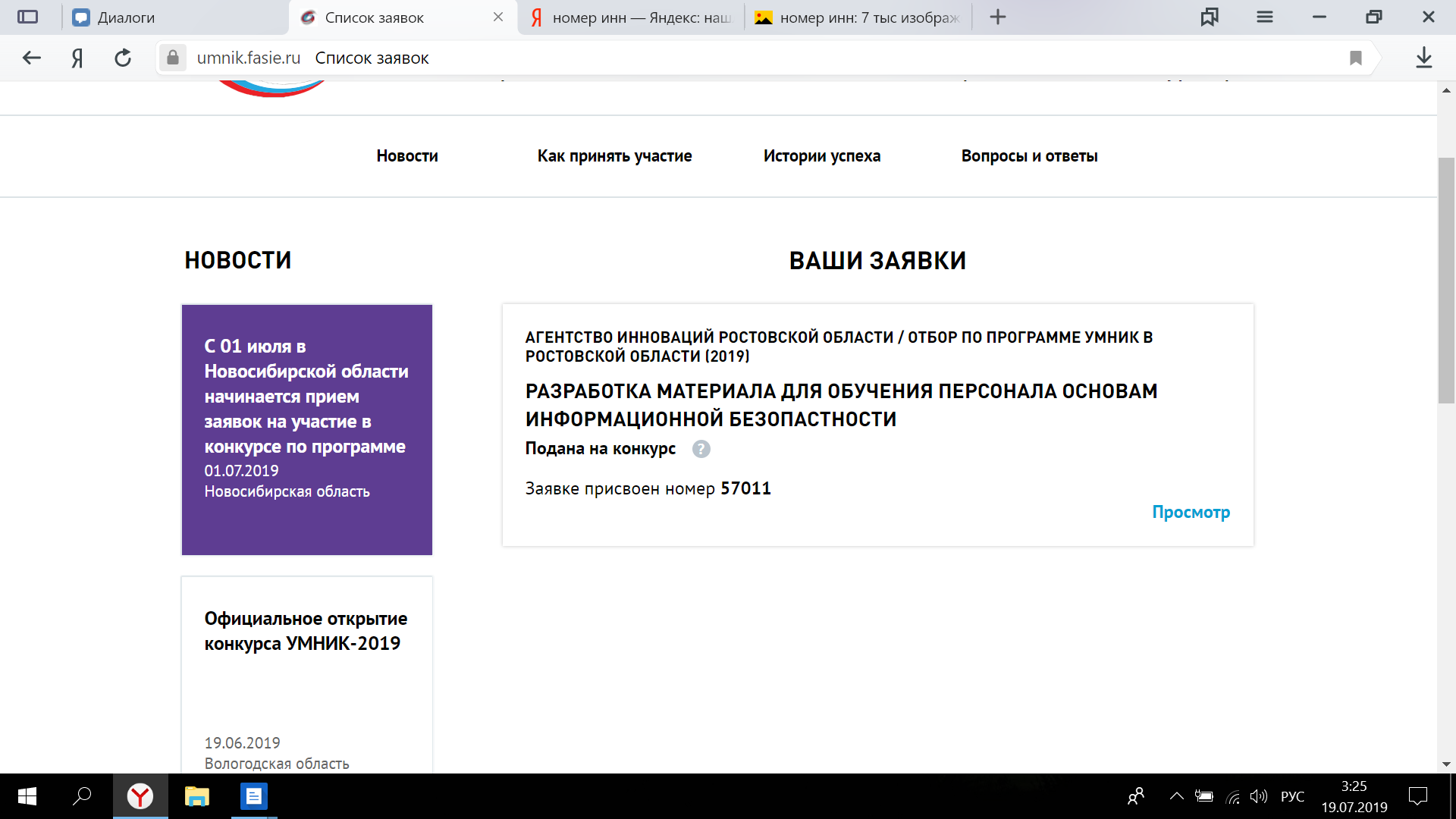
Создание таблиц миграции



**Личный кабинет**

****

**Заявка**

****