

# Гиперказуальные игры

Это простые игры с элементарной механикой, в которых происходит мгновенный переход к геймплею.

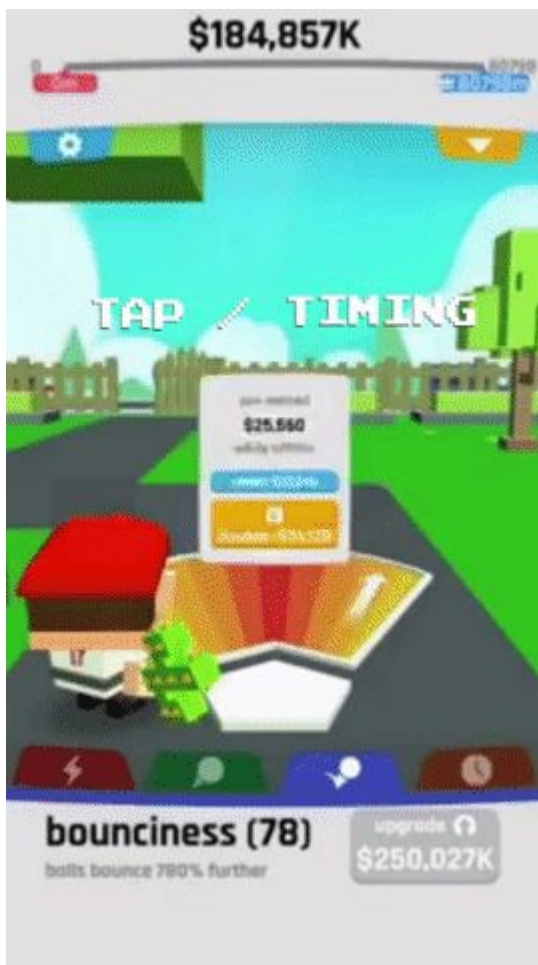


# Преимущества

- Благодаря своей фундаментальной «простоте» гипер-казуальные игры не только легки в прохождении, но в них хочется играть снова и снова.
- Комбинация простой механики с минималистичным UI обеспечивает доступный и невероятно увлекательный пользовательский опыт без какого либо обучения. В отличие от других жанров игроки могут мгновенно погрузиться в процесс и «подсесть» на игру.
- Также, отличие от других жанров, у которых более узкая аудитория, гипер-казуальные игры созданы для всех.

- Механика гипер казуальных игр строится не столько на оригинальности, сколько на уже известных и зарекомендовавших себя идея. Тетрис, три в ряд, раннеры, арканоиды, платформеры - всё это настолько глубоко укоренилось в сознании и современной поп-культуре, что даже человек, никогда не прикоснувшийся к играм ранее, интуитивно поймёт, чего от него хотят. А именно это и определяет успех гипер казуальных игр как наиболее массового, понятного и доступного продукта для широкой аудитории.

# Касание / Тайминговые механики



- В игре которая представляет собой чистый игровой процесс с касанием и таймингом, механика основывается на точном нажатии или точном тайминге. Точность является наиболее важным аспектом игры и цель для пользователя — достичь в этом совершенства. Только идеально точное касание принесет максимальный балл.

# Механика укладки блоков



- Стековая механика продолжает механику касания / тайминга, добавляя результаты предыдущих нажатий к прогрессу раунда. Игра The Tower by Ketchapp является хорошим примером, когда сама башня состоит из ранее сложенных блоков. Каждый раз когда игроку не удастся получить идеальный стэк, сама башня сжимается и становится тяжелее получить следующий стэк.
- Механики стека дают игрокам больше возможности для неудачи, причем каждая неудача имеет меньший эффект, чем в игре с простым тапом. Они смягчают неудачу, позволяя вам продолжить, но сохраняют визуальную ясность того, как произошла эта ошибка. Небольшие наказания за неудачу продлевают раунд, но длинные раунды также передают чувство легкости.

# Механики поворота



- Поворот является последним из механик касания и тайминга. Это добавляет дополнительную сложность к каждому касанию, добавляя сбивающее с толку зрительное восприятие. У зрительной коры человека есть встроенная слабость в оценке длины между горизонтальной и вертикальной формами в трехмерном пространстве. Зрительная кора может быть обманута довольно легко и многие визуальные иллюзии демонстрируют это, The Ponzo Illusion является хорошим примером. Вы все еще просите игрока рассчитать время одного нажатия, но с добавлением путаницы в трехмерном пространстве игроки с большей вероятностью будут ошибаться. Это гораздо сложнее освоить, чем подход на основе 2D.

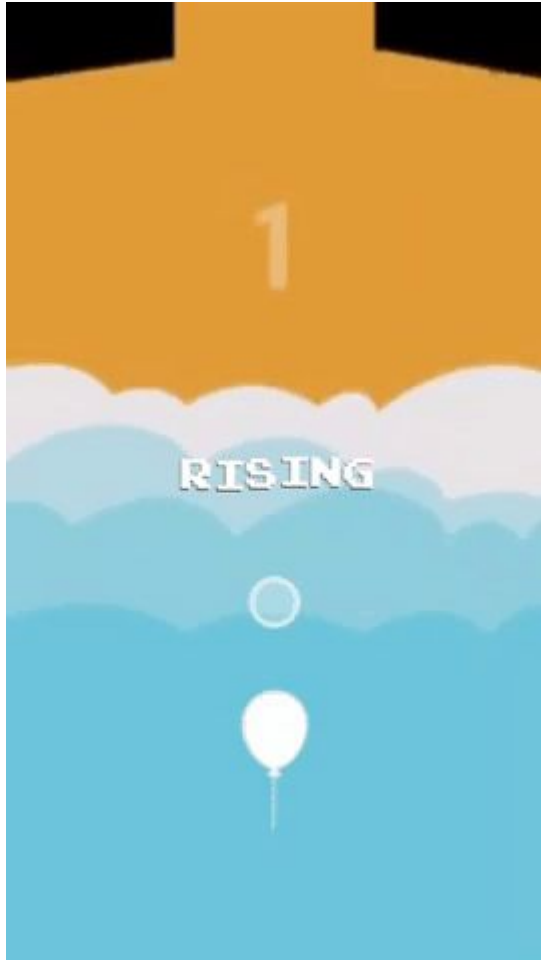
# Механики ловкости



- Эти игры в основном ориентированы на игрока, который выполняет очень простое действие, которое он должен выполнять много сотен раз. С достаточной практикой, эти механики могут быть освоены ловкими игроками и поэтому наивысшие баллы — это справедливое представление о ловкости и мастерстве игрока. Чтобы эти игры были увлекательными, игра обычно должна ускоряться. Механика лёгкая, когда скорость медленная, но под давлением все более и более повышается вероятность, что вы совершите ошибку.



# Механики восхождения и падения



- Механики восхождения и падения предоставляют интересные путешествия для игроков. Постоянная прогрессия уровня приводит к ощущению прогресса без изменения в механике или цели. Чтобы развлечь людей, сам уровень должен развиваться. Rise Up от Serkan Özyilmaz и Helix Jump от Voodoo показывают, как развивается прогресс по мере продвижения вверх или вниз в игре.
- Игрок сосредоточен на решении следующей задачи по ходу игры, а не на точности. Есть много путей, чтобы пройти эти уровни. Часто нужно немного удачи по таймингу или скиллу. Ваша единственная цель — защитить объект даже от одной ошибки.

# Механики уклонения

- В этих играх основное внимание уделяется движению пальца, позволяющее избежать препятствий. Большую часть времени они основаны на механике уклонения в том же духе, что подъем и падения. Но они больше внимания уделяют ловкости, а не времени. Игры с этой механикой позволяют максимально эффективно управлять сенсорным экраном и их сложно воспроизвести на других устройствах. Это дает им оригинальное ощущение и приятное использование сенсорного ввода.

# Механик объединения



- Механики объединения очень легко понять игрокам. Похожие вещи объединяются, разные вещи не сочетаются. Таким образом игра становится очень простой для людей и с каждым последующим слиянием игрокам передается новое знание и сильное чувство прогресса. Сложность и вызов в этой игре, как правило, приходят в виде метагейма. Но для казуальной аудитории метагейм может стать разделителем из-за которого игра покажется слишком сложной и они покинут её.

# Механики кликеров



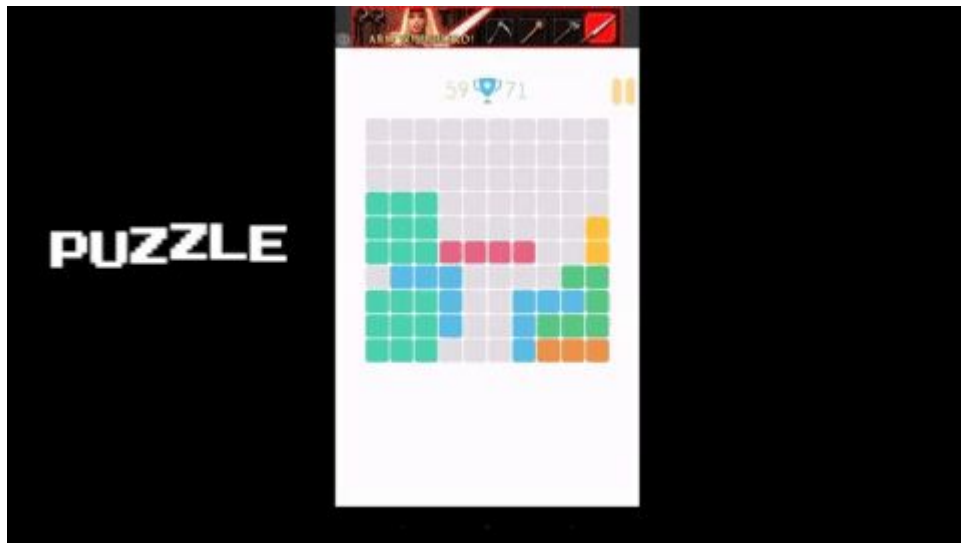
- Кликеры в качестве механики уже несколько лет используется в гиперказуальных мидкорных играх. Сложность и опора на механику — выбор каждого игрового дизайнера. По своей сути, это любая механика которая не требует ввода от игрока, чтобы прогрессировать. Очевидно, что отсутствие ввода делает игру очень казуальной и без цели она может стать скучной. Большую часть времени механика кликера образует вторичную механику, привязанную к мягкой валюте. Это работает хорошо, потому что со временем игроки зарабатывают больше денег, которые они могут потратить в своем кор-геймплее.

# Механика роста



- Механика роста очень похожа на механику кликера в том смысле, что она обычно не зависит от управления. Победители в этом жанре гипер-казуальных игр всегда самые крупные и в некоторых случаях могут съесть других игроков, по сути заканчивая раунд. Механика сама по себе хорошо понятна, но развивает интересный опыт.

# Механики головоломок



- Головоломка — это сам по себе жанр, но гиперказуальные головоломки ориентированы на простоту, а не на сложность. Хорошая гипер-казуальная игра-головоломка обычно не имеет конца. Игроков просто просят продолжать играть в головоломки как можно дольше и игра не увеличивает сложность. Сама механика должна усложняться за счет действий пользователей.

# Пайплайн создания гиперказуальной игры

## Этапы создания игры

### Подготовительный этап

Это первый и очень значимый этап работы над игрой.

### Производство

Это ключевой и самый долгий этап создания игры

### Выпуск

Распространение и выход игры в свет

### Поддержка

Работа над ошибками

Создание игр происходит строго поэтапно, к тому же очень не рекомендуется пропускать какие либо стадии разработки игры.

# Подготовительный этап

- концепция. Это первый этап, на котором команда разработчиков придумывает и прорабатывает дизайн проекта, его основные идеи и механики. На этом шаге изначальная идея может претерпеть множество изменений прежде, чем станет оформившейся и подходящей для дальнейшей разработки. Результатом концептирования является концепт-документ - эдакая итоговая компиляция идей;



# Производство

- прототипирование. Следующий шаг - создание прототипа для разработки и оттачивания игровых механик, которые могут привлекательно выглядеть на бумаге, но в реальности столкнутся с множеством проблем. Прототип не нуждается в точной проработке - главное проверить ключевые моменты, вокруг которых строится игра;
- вертикальный срез. Это - минимально готовая версия игры, которая содержит готовый игровой процесс. При этом наибольшее качество проработки должно быть только у тех элементов, которые имеют ключевую важность для проекта, другими можно до поры пренебречь. Такой продукт может оценить [издатель компьютерных игр](#);
- создание контента. После проработки основных игровых механик можно приступить к созданию контента, с которым проект можно выпускать для внешней аудитории. Именно тут художники и дизайнеры начинают выполнять большую часть работы, а остальные специалисты оттачивают баланс;
- закрытая бета. Это - своеобразный стресс-тест, демонстрация игры закрытому кругу лояльных лиц для проверки работоспособности проекта. Критично важно отловить на этом уровне любые баги и недоработки, чтобы исправить их для следующего этапа;

# Выпуск игры

- софт-лонч. Если вы не хотите рисковать, то проект можно выпустить софт-лончем в виде открытого бета-тестирования для уже широкой аудитории. На этом этапе игра уже должна быть монетизирована, поскольку она будет приносить вам прибыль сразу после запуска;
- релиз. После выпуска игры можно анализировать успех и прибыль мобильного проекта. Если у вас полностью отлажено оперирование продукта, комьюнити и поддержка, то всё сделано правильно.

# Концепт-документ

- **Базовый геймплей**

Необходимо выделить для себя, какая часть игры у вас является базовой, то есть, в чём игрок будет проводить больше всего времени. Например, в нашумевшей Gardenscapes игрок в основном играет в match-3, хоть это и не подчёркивается. Там создатели делают упор на обустройство сада, а match-3 как бы второстепенное занятие, служащее для зарабатывания звёзд. Но на самом деле в этом разделе нужно описывать именно его.

# Концепт-документ

Например: 1. Персонаж двигается в изометрии при помощи WASD. 1.1 Если игрок направляет персонажа в стену, тот останавливается при столкновении со стеной. 2. При нажатии на ЛКМ игрок стреляет. И так далее.

# Концепт-документ

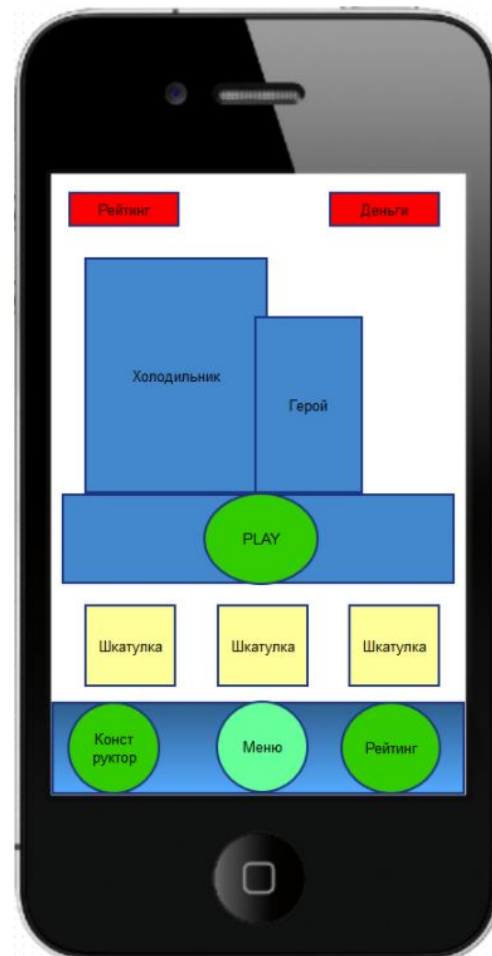
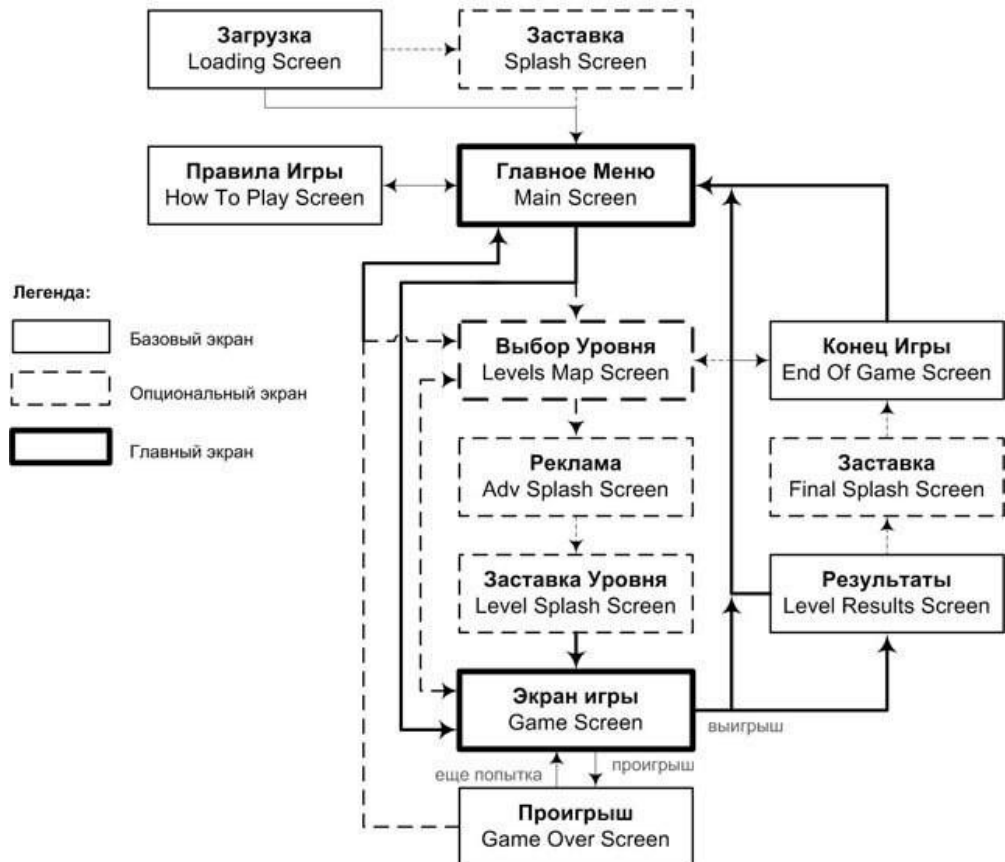
- **HUD или интерфейс базовой игры**

Head-up display — это то, что видит игрок во время базового геймплея. Это одна из самых важных вещей в игре — игрок должен понимать то, во что играет, но при этом оно не должно отвлекать его от геймплея.

# Концепт-документ

## • Интерфейсы и главное меню (эскиз)

Схема Экранов Шаблона Игры



# Концепт-документ

- **Остальные фичи**

Дальше мы просто бьём проект на мелкие составляющие и описываем каждый в своём разделе. Тьюториал, инвентарь, диалоги, глобальная карта, прокачка — всё, что у вас есть. Описывается всё так же, как и базовый геймплей. К каждой из фич в раздел стоит запихнуть мокап того, что с ней связано.

# Концепт-документ

- **Функциональные разделы**

Если что-то не описали в предыдущих разделах, например, балансировку и т.д., пишем сюда