

The background is a dark gray field filled with a dense pattern of isometric cubes. Some cubes are outlined in bright colors like yellow, cyan, and magenta, while others are just gray outlines. The perspective is from an isometric view, looking down at the cubes. The text is centered in the middle of the image.

INTO THE RADIUS VR

Особенности VR игр и инструментарий разработки

План доклада

- Об игре
- Старт разработки
- Особенности VR игр
- Инструменты и процессы

Что такое VR игры и где в них играют

VR - Virtual Reality, ВР - виртуальная реальность. В рамках своего доклада я подразумеваю под VR играми игры, разработанные специально для шлемов виртуальной реальности с 6-dof контроллерами.

VR Игры можно разделить по платформам:

- PC. Открытая платформа, зоопарк устройств
- PlayStation. Закрытая платформа, очень не дружелюбная
- Standalone. Закрытая, но лучше чем Сони, в среднем x4 от ПК рынка

INTO THE RADIUS VR



Into the Radius is a single-player survival shooter in VR. Explore the Pechorsk zone filled with surreal landscapes and dangerous anomalies. Defend yourself with realistic firearms, recover strange artifacts, scavenge for loot, and uncover the mysteries of this unforgiving dystopian environment.

RECENT REVIEWS: **Very Positive** (435)

ALL REVIEWS: **Very Positive** (4,974)

RELEASE DATE: 20 Jul, 2020

DEVELOPER: **CM Games**

PUBLISHER: **CM Games**


Popular user-defined tags for this product:

VR **Survival** **Atmospheric** **Singleplayer** **Horror** +




INTO THE RADIUS VR

[Games](#) [Apps](#) [Entertainment](#) [Wishlist](#)




Into the Radius


 **PEGI 16**
Use of Alcohol/Tobacco

★★★★★ 1,312 Ratings

An atmospheric open-world single-player survival shooter.

Purchased

 Buy for a friend

 Share

INTO THE RADIUS VR

- VR игра от первого лица об исследовании и выживании в аномальной зоне Радиуса.



INTO THE RADIUS VR

- Сюрреалистичные ландшафты, реалистичное взаимодействие с оружием, акценты на атмосфере, саспенсе и вдумчивом геймплее.



Контекст начала разработки

- 2017ый год
- С выходом Vive всем стало ясно, что будущее за VR с 6dof контроллерами. Использование VR без контроллеров - не торт, что ограничивает портирование плоских игр.
- Игры на рынке (кроме Fallout VR) являлись небольшими по длительности аркадами, захватывающими, но быстро приедающимися. Тем не менее, продавались они с неплохим профитом (для небольшой команды).
- Основной идеей проекта стало создание “полноценной” игры, разработанной специально для VR

0 времени разработки

Игра разрабатывается более 5 лет, в разные моменты времени разработка велась силами от 1го до 20 человек.

Весна 2017
Начало прототипа

Май 2018
Первый фулл-тайм
работник

Март 2019
старт проекта, 4 человека

Июль 2020
выход из раннего
доступа

Апрель 2022
анонс MQ2 версии

3 месяца - первый
питч

Февраль 2019
Закрытая альфа на
Steam

Ноябрь 2019
платный ранний
доступ

Ноябрь 2021
2.0 update

Сентябрь 2022
MQ2 релиз



Старт разработки

Весна 2017
Первый прототип игры ->



Особенности разработки игр для VR (1)

- Новая область - нет выработанных стандартов управления, UX, ментальные модели пользователей вступают в конфликт при переходе из игры в игру.
- Многое ,что в обычных играх не важно, в VR становится важным или ключевым. Например, взаимодействие с предметами окружения, масштаб и размер предметов.

Особенности разработки игр для VR (1)

- Производительность. Очень актуальна тк лаги вызывают неприятные ощущения вплоть до тошноты.

Плоские игры работают при частоте кадров от 30 до 60, могут позволить уменьшать разрешение на лету или иногда подлагивать. VR же работает от 72 до 120 кадров в сек (13 - 8мс).

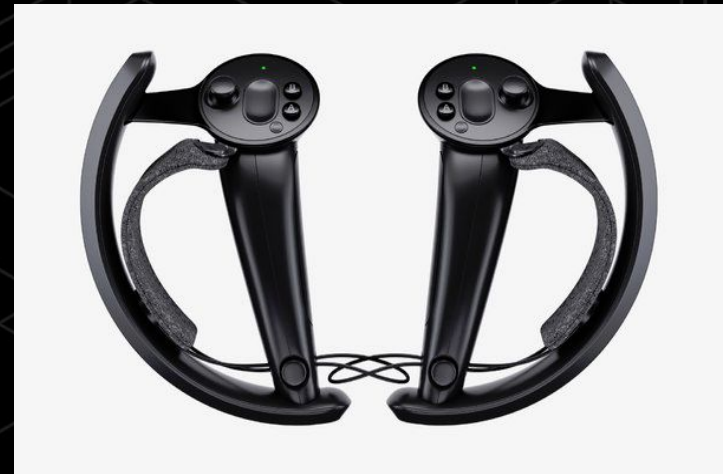
В шлеме - изображение стерео, то нужно отрисовывать каждый кадр дважды, а разрешение выше чем FullHD.

В Standalone VR это все еще происходит на 3х ядрах мобильного чипа из флагманского телефона 3х летней давности.

По этому внимание к оптимизации необходимо как графике так и игровой логике.

Особенности разработки игр для VR (2)

- Физически разные контроллеры у разных производителей, с разным набором кнопок, джойстиков итд. Постоянно выходят новые и поддерживать нужно фактически ВСЕ.



Особенности разработки игр для VR (3)

- Звуки играют куда большую роль в VR, чем в плоских играх. Правильное позиционирование звука, а также звуковая обратная связь очень важны для восприятия.
- Больше погружение игрока, ближе к его физиологии. Фобии, тошнота, размеры тела. Без тестирования на больших аудиториях многие проблемы не будут замечены вплоть до релиза.
- Труднее тестировать и обеспечивать качество из-за того, что пользователь выполняет многие действия сам. Автотесты либо сильно ограничены, либо требуют поддержки полностью функционального альтернативного управления.

Инструментарий

!DISCLAMER!

Дальнейшая информация относится конкретно к нашей ситуации и нашим проблемам в конкретный момент времени.



Game Engine

Это основной софт в котором происходит разработка игры. Огромный набор разных подсистем, которые отвечают за то, как и где будет работать и как выглядеть ваша игра. Как происходит разработка, какие инструменты используются итд основываются на его выборе.

Смена движка в дальнейшем == начало разработки фактически с чистого листа.

Game Engine: Unreal Engine 4

Почему UE а не Unity?

- Изначально я - художник, я писал скрипты и инструменты для 3ds MAX, но визуальное программирование для меня более наглядно и позволяет быстрее прототипировать.
- Большая интегрированность инструментов - террейн, рендерер, редактор материалов, аи, анимация итд присутствуют из коробки и не требуют напильника, чтобы работать вместе. Так же “гарантированно” поддерживаются и развиваются в новых версиях.

Тк концепт игры предполагал открытые пространства, деревья, продвинутых врагов, то наличие этих систем из коробки было значительным плюсом.

Разработка начиналась на 4.15 и до 4.27 велась на бинарных дистрибутивах. Сейчас у нас своя модифицированная версия (тк исходный код открыт).

Game Engine: Unreal Engine 4

Недостатки этого выбора:

- Визуальный язык куда сложнее мерджить, чем обычный код. Но как и с обычным кодом решается организацией производства и стандартами написания.
- Глубокая оптимизация материалов требует лезть в исходный код движка = очень высокий порог входа.
- Система префабов Unity в целом удобней в работе, особенно с появлением вложенности префаб в префабе.
- C++ разработчика найти сложнее чем C#

Version Control

Как движок является основой вашей игры, так и система контроля версий (VCS) является основой вашей командной работы. Даже прототипы, которые делаются одним человеком, лучше сразу вносить в VCS.

Цель: Обеспечивать параллельную работу распределенной, а главное **мультидисциплинарной** команды команды без потери работы и дополнительных усилий на синхронизацию.

Version Control: Perforce Helix

Всего 3 актуальных софта на рынке - пробовали все.

- 1) Git - UE4 в основном бинарные данные - уровни, блюпринты итд - то нужно использовать Git LFS. Так же чтобы не было конфликтов, нужен lock. Нет нормальной поддержки этого функционала в UI клиентов, движка итд. Постоянно вмешивается человеческий фактор с потерей работы и нервов.
- 2) Plastic SCM - более дружелюбен для "не кодеров", имеет lock, НО он сквозной для всех бранчей, что не дает нормально работать с бранчами.
- 3) Perforce Helix - все хорошо, но стоит приличных денег. Для маленьких команд до 5 юзеров бесплатен.

ART

Конкретные инструменты не так важны и зависят скорее от бюджетов и навыков специалистов. Все современные пакеты обладают достаточным функционалом для создания 3д моделей для игры.

Проблемы, стоящие перед арт.отделом:

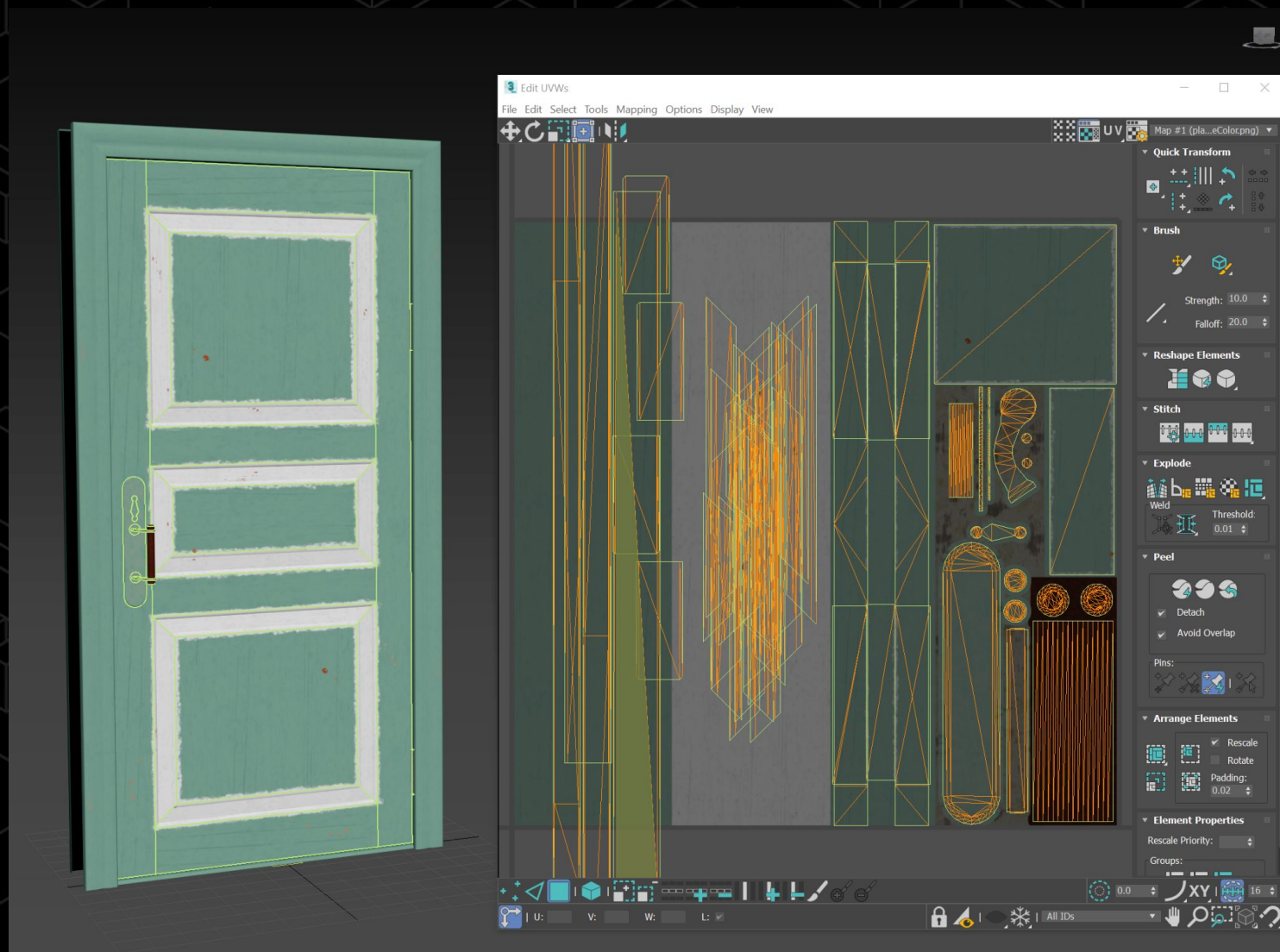
- 1) Пользователь может подносить предметы или рассматривать окружение вплотную.
- 2) Важно соответствие всего своим реальным масштабам и ощущению.
- 3) Большое количество контента для наполнения мира, чтобы он не ощущался пустым.

В свете данных проблем есть несколько особенностей:

- Активы с маркетплейсов почти никогда не подходят для окружения из-за проблем с масштабом, сложных шейдеров и не оптимального изготовления.
- Но как референсы, отдельные предметы или спецэффекты вполне могут использоваться для экономии ресурсов команды.

ART

Tile-trim подход. Когда все поверхности модели текстурируются с помощью наложений на специальную текстуру-атлас.



Спасибо за внимание!

aleksei.shulga@cm.games

