

# Flash движок в игре Зомби Ферма

Flash  
GAMM!

- Проблемы в процессе разработки и их решения





VIZOR  
INTERACTIVE

# Начало разработки Требование к движку

Flash  
GAMM!

- Обзор существующих Flash-движков
- Основной упор при разработке на оптимизацию



# Способ отрисовки

Flash  
GAMM!

- Display List и Bitmap Biting
- Движок изначально писался на Action Script 2
- Однозначный выбор – использование Display List





VIZOR  
INTERACTIVE

# Изометрия

Flash  
GAMM!

- Разделение объектов в мире на статические объекты и персонажей
- Объекты хранятся в массиве в порядке их отображения
- Сортировка персонажей осуществляется каждые 50 мс
- Многоэтажность





VIZOR  
INTERACTIVE

# Проблемы производительности

Flash  
GAMM!

- При отображении тысяч спрайтов наблюдались сильные тормоза при движении мыши
- Flash шлет события мыши каждому спрайту, добавленному в Display List



# Решение проблемы

Flash  
GAMM!



VIZOR  
INTERACTIVE

Application



mainSprite



UI Controller

- Блокирование всех событий мыши
- Добавление объекта UI Controller
- UI система. UIComponent и UIContainer





VIZOR  
INTERACTIVE

# Система анимирования персонажей

Flash  
GAMM!



- Использование MovieClip тормозит
- Персонаж состоит из слоев
- Кадры персонажа собираются в одну большую растровую картинку
- Под каждый слой создается объект Bitmap
- Анимирование осуществляется путем копирования части растровой картинки в объект Bitmap





VIZOR  
INTERACTIVE

# 50 Зомби!

Flash  
**GAMM!**





# Звуки

- Создание стерео звуков. Использование SoundTransform
- SoundManager для хранения и менеджмента всех звуков в игре



# Кеширование

- Графика объектов собирается группами в swf файлы
- Менеджер объектов следит за количеством созданных копий картинок
- Для подгрузки и кеширования картинок напрямую используется класс `CachedImage`



# Сигналы

- Проблема разработки при разрастании проекта
- Использование сигналов
- Простейший базовый класс логики
- Важность правильной структуры кода на начальных этапах разработки



# Обход препятствий

- Алгоритм обхода в ширину даёт большую нагрузку
- Обход препятствий при необходимости
- Проблемы погрешностей при вычислениях
- Учёт физики карты



# Тестирование

- Обновления нужно делать в короткие сроки
- Все ошибки не возможно найти на этапе тестирования внутри компании
- Критические ошибки Flash отправляются на сервер и записываются в логи



# Организация работы

Flash  
GAMM!

- Программист не должен заниматься добавлением контента
- Редактор объектов и карт
- Конвертер спрайтов





VIZOR  
INTERACTIVE

# Редактор карт и объектов

Flash  
GAMM!



- Редактор карт дает возможность добавлять в игру спрайты, анимированные спрайты, составлять из спрайтов объекты, а из объектов композиции
- В редакторе задаются свойства клеток поля
- Структура композиции на примере могилы дровосека



# Конвертер спрайтов

Flash  
GAMM!

- Кадры сортируются по папкам в удобном для использования порядке
- Создаётся конфиг файл для настройки анимаций
- Анимация конвертируется при сборке проекта







VIZOR  
INTERACTIVE

# Портирование под социальные сети

Flash  
GAMM!

- Каждая сеть имеет свои технические особенности
- Использование препроцессора во flex
- Проект всегда имеет одну ветку в системе контроля версий
- Проект собирается и заливается на сервер одной командой с параметром социальной сети





VIZOR  
INTERACTIVE

# Isomech Engine

Flash  
**GAMM!**



- Быстрый изометрический движок
- Оптимизированная UI-система
- Оптимизированная анимация персонажей
- Автоматизированное добавление контента
- Многоэтажность
- Автоматизированная сборка под все социальные сети





VIZOR  
INTERACTIVE

# Server-side

Flash  
GAMM!



## Требования

- Высокие нагрузки
- Линейное масштабирование
- Простота написания логики игры

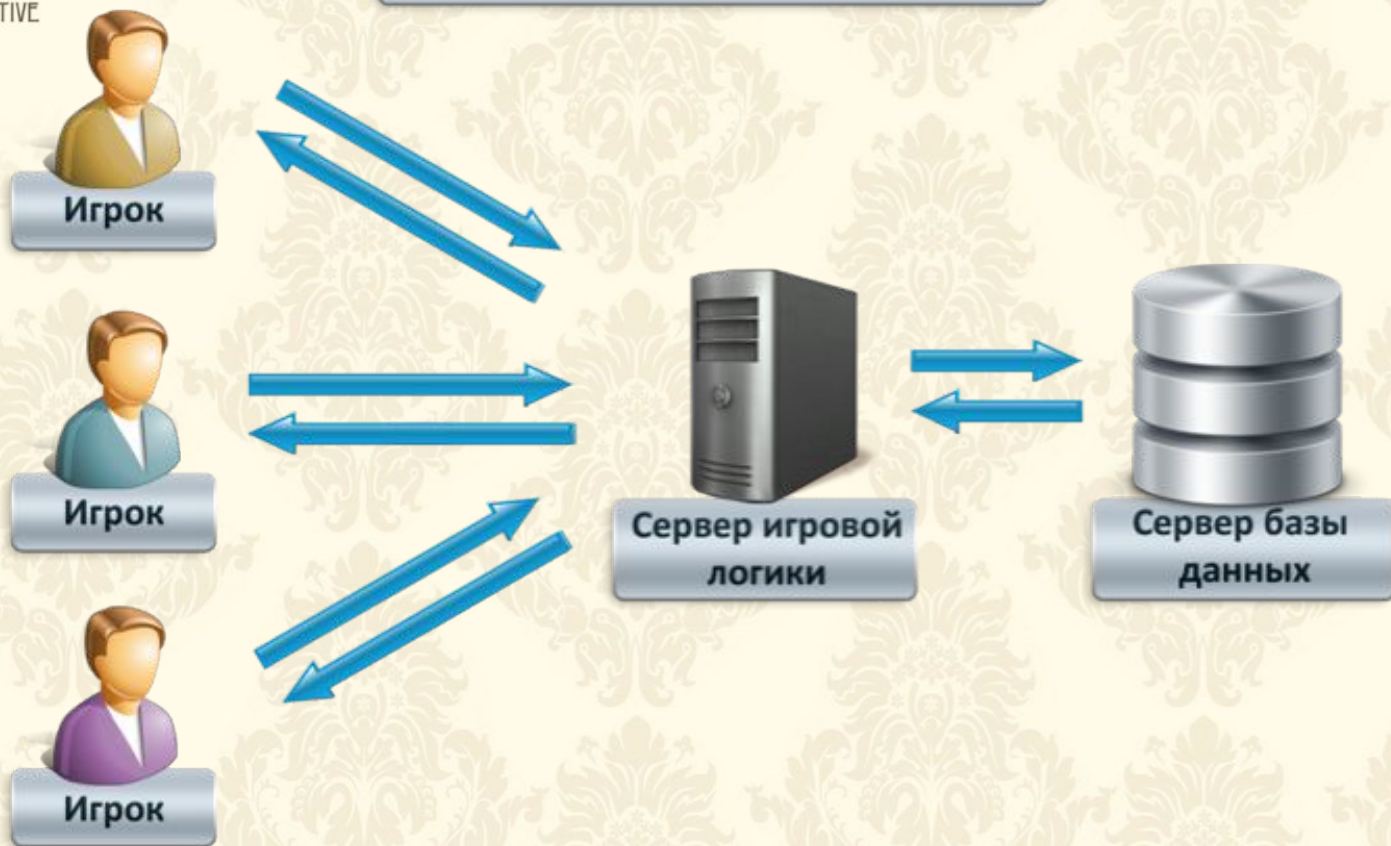




VIZOR  
INTERACTIVE

Flash  
**GAMM!**

## Базовая схема





VIZOR  
INTERACTIVE

Flash  
**GAMM!**

Старт





VIZOR  
INTERACTIVE

# После старта

Flash  
**GAMM!**

- Распределенная система серверов логики
- PostgreSQL в качестве БД
- PostgreSQL в качестве системы синхронизации серверов логики





VIZOR  
INTERACTIVE

# Проблема

Flash  
GAMM!

## База Данных

- не справлялась с нагрузкой
  - хаотичный рост размера данных
  - отсутствие устойчивости к потере сервера
  - использование базы данных не по назначению
- в качестве синхронизатора





VIZOR  
INTERACTIVE

# Выбор базы

Flash  
GAMM!

## Требования

- распределённая база
- удобные инструменты работы с базой, в частности простота расширения кластера
- отказоустойчивость
- возможность простой адаптации текущей модели игры под базу







VIZOR  
INTERACTIVE

# Выбор базы

Flash  
**GAMM!**

## Варианты

- mongodb
- HBase
- Cassandra





VIZOR  
INTERACTIVE

# Кеш и синхронизация

Flash  
GAMM!

## Варианты

- ZooKeeper
- memcached
- hazelcast





VIZOR  
INTERACTIVE

# Статистика

Flash  
GAMM!

## Модуль статистики

- все данные собираются в разрезах по полу, возрасту, стране
- основные параметры данных:
  - идентификатор события
  - числовое значение события
- в качестве дополнительных параметров выступает массив подкатегорий





VIZOR  
INTERACTIVE

Flash  
**GAMM!**





VIZOR  
INTERACTIVE

# ИТОГИ

Flash  
GAMM!

- Распределенная система работы логики
- Hazelcast в качестве системы синхронизации серверов и кэширования данных
- Cassandra в качестве распределенной БД
- Сессия игрока привязана к определенному серверу
- 9 серверов БД + 10 серверов логики = 120000 CCU, 1100000 DAU (RU)
- 450 GB данных игроков (RU)





VIZOR  
INTERACTIVE

# Спасибо за внимание!

Flash  
**GAMM!**

# Вопросы?

E-mail: [office@vizer-interactive.by](mailto:office@vizer-interactive.by)

