

Война вирусов

Походовая онлайн-игра на платформе Android

Лапшин Дмитрий, Степанов Всеволод

СПб АУ РАН

Осень 2015

Содержание

- 1 Содержание
- 2 О проекте
- 3 Клиент
 - Бот
 - Хранение игр
 - Игра по сети
- 4 Сервер
- 5 Сложности

Цель проекта

Реализовать онлайн-версию игры „Война Вирусов“ с возможностью игры с другим игроком из списка знакомых или случайным противником.

Игра проходит на клеточном поле 10×10 , игроки по очереди ставят свои фигуры, захватывая свободные клетки и уничтожая клетки соперника.

Правила игры достаточно просты:

Игроки ходят по очереди, в каждый ход у игрока есть 3 ходика, за каждый из них можно либо занимать клетку в „Зоне влияния“, либо убивать чужого вируса в этой зоне.

„Зона влияния“ – множество клеток, соседних со своими живыми вирусами, а так же с убитыми вражескими клетками в зоне влияния

Цель проекта

Реализовать онлайн-версию игры „Война Вирусов“ с возможностью игры с другим игроком из списка знакомых или случайным противником.

Игра проходит на клеточном поле 10×10 , игроки по очереди ставят свои фигуры, захватывая свободные клетки и уничтожая клетки соперника.

Правила игры достаточно просты:

Игроки ходят по очереди, в каждый ход у игрока есть 3 ходика, за каждый из них можно либо занимать клетку в „Зоне влияния“, либо убивать чужого вируса в этой зоне.

„Зона влияния“ – множество клеток, соседних со своими живыми вирусами, а так же с убитыми вражескими клетками в зоне влияния

0 проекте

Игра заканчивается, когда у кого-то из игроков все вирусы были убиты



Общий план

- 1 Реализация состоит из клиентской и серверной части. Клиент позволяет пользователю играть, в то время как сервер соединяет клиентов и обрабатывает игры.
- 2 Уже прошедшие игры хранятся в базе данных: сервер хранит все, клиент хранит только свои.
- 3 Сервер и клиенты между собой общаются через систему Google Cloud Messaging, которая позволяет клиенту и серверу общаться, не тратя батарею телефона на отдельное сетевое соединение.

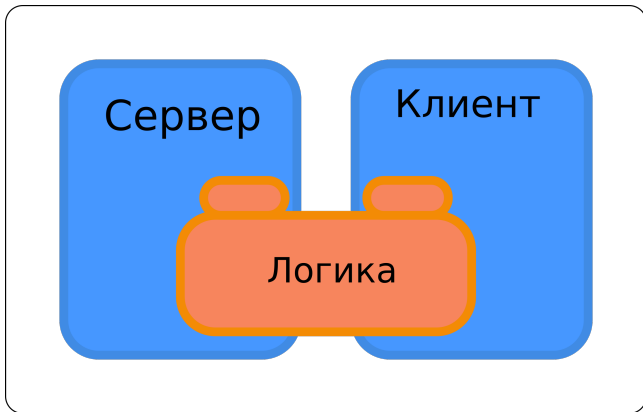
Общий план

- 1 Реализация состоит из клиентской и серверной части. Клиент позволяет пользователю играть, в то время как сервер соединяет клиентов и обрабатывает игры.
- 2 Уже прошедшие игры хранятся в базе данных: сервер хранит все, клиент хранит только свои.
- 3 Сервер и клиенты между собой общаются через систему Google Cloud Messaging, которая позволяет клиенту и серверу общаться, не тратя батарею телефона на отдельное сетевое соединение.

Общий план

- 1 Реализация состоит из клиентской и серверной части. Клиент позволяет пользователю играть, в то время как сервер соединяет клиентов и обрабатывает игры.
- 2 Уже прошедшие игры хранятся в базе данных: сервер хранит все, клиент хранит только свои.
- 3 Сервер и клиенты между собой общаются через систему Google Cloud Messaging, которая позволяет клиенту и серверу общаться, не тратя батарею телефона на отдельное сетевое соединение.

Архитектура



Клиент

- 1 3 режима игры: вдвоем на одном устройстве, против бота, по сети.
- 2 Возможность просматривать историю уже прошедших игр.
- 3 Текущая игра сохраняется: если приложение случайно закрылось, то всегда можно восстановить текущую игру.
При необходимости, при игре по сети с сервера берется более актуальная версия
- 4 Аскетичный интерфейс
Графика в SVG формате, чтобы идеально масштабировалось под любые размеры экранов. Цвета фигур выбираются игроками.

Клиент

- 1 3 режима игры: вдвоем на одном устройстве, против бота, по сети.
- 2 Возможность просматривать историю уже прошедших игр.
- 3 Текущая игра сохраняется: если приложение случайно закрылось, то всегда можно восстановить текущую игру.
При необходимости, при игре по сети с сервера берется более актуальная версия
- 4 Аскетичный интерфейс
Графика в SVG формате, чтобы идеально масштабировалось под любые размеры экранов. Цвета фигур выбираются игроками.

Клиент

- 1 3 режима игры: вдвоем на одном устройстве, против бота, по сети.
- 2 Возможность просматривать историю уже прошедших игр.
- 3 Текущая игра сохраняется: если приложение случайно закрылось, то всегда можно восстановить текущую игру.
При необходимости, при игре по сети с сервера берется более актуальная версия
- 4 Аскетичный интерфейс
Графика в SVG формате, чтобы идеально масштабировалось под любые размеры экранов. Цвета фигур выбираются игроками.

Клиент

- 1 3 режима игры: вдвоем на одном устройстве, против бота, по сети.
- 2 Возможность просматривать историю уже прошедших игр.
- 3 Текущая игра сохраняется: если приложение случайно закрылось, то всегда можно восстановить текущую игру.
При необходимости, при игре по сети с сервера берется более актуальная версия
- 4 Аскетичный интерфейс
Графика в SVG формате, чтобы идеально масштабировалось под любые размеры экранов. Цвета фигур выбираются игроками.

Бот

- 1 Не являлся основной целью, но хотелось иметь более-менее адекватного противника
- 2 Эвристические подходы не очень понятны, требуют времени и сил на реализацию и проверку, поэтому стратегия переборная
- 3 Проблемы:
 - 1 Большая ветвистость дерева перебора
 - 2 Один ход за три
 - 3 Медлительность телефонов и неоптимизированный код
- 4 Итоговый вариант
 - 1 Перебор трех ходиков
 - 2 Играем за противника жадно еще три ходика
 - 3 Оценочная функция – линейная комбинация количества контролируемых клеток и клеток под ударом
- 5 Можно улучшить, но не было времени.

Бот

- 1 Не являлся основной целью, но хотелось иметь более-менее адекватного противника
- 2 Эвристические подходы не очень понятны, требуют времени и сил на реализацию и проверку, поэтому стратегия переборная
- 3 Проблемы:
 - 1 Большая ветвистость дерева перебора
 - 2 Один ход за три
 - 3 Медлительность телефонов и неоптимизированный код
- 4 Итоговый вариант
 - 1 Перебор трех ходиков
 - 2 Играем за противника жадно еще три ходика
 - 3 Оценочная функция – линейная комбинация количества контролируемых клеток и клеток под ударом
- 5 Можно улучшить, но не было времени.

Бот

- 1 Не являлся основной целью, но хотелось иметь более-менее адекватного противника
- 2 Эвристические подходы не очень понятны, требуют времени и сил на реализацию и проверку, поэтому стратегия переборная
- 3 Проблемы:
 - 1 Большая ветвистость дерева перебора
 - 2 Один ход за три
 - 3 Медлительность телефонов и неоптимизированный код
- 4 Итоговый вариант
 - 1 Перебор трех ходиков
 - 2 Играем за противника жадно еще три ходика
 - 3 Оценочная функция – линейная комбинация количества контролируемых клеток и клеток под ударом
- 5 Можно улучшить, но не было времени.

Бот

- 1 Не являлся основной целью, но хотелось иметь более-менее адекватного противника
- 2 Эвристические подходы не очень понятны, требуют времени и сил на реализацию и проверку, поэтому стратегия переборная
- 3 Проблемы:
 - 1 Большая ветвистость дерева перебора
 - 2 Один ход за три
 - 3 Медлительность телефонов и неоптимизированный код
- 4 Итоговый вариант
 - 1 Перебор трех ходиков
 - 2 Играем за противника жадно еще три ходика
 - 3 Оценочная функция – линейная комбинация количества контролируемых клеток и клеток под ударом
- 5 Можно улучшить, но не было времени.

Бот

- 1 Не являлся основной целью, но хотелось иметь более-менее адекватного противника
- 2 Эвристические подходы не очень понятны, требуют времени и сил на реализацию и проверку, поэтому стратегия переборная
- 3 Проблемы:
 - 1 Большая ветвистость дерева перебора
 - 2 Один ход за три
 - 3 Медлительность телефонов и неоптимизированный код
- 4 Итоговый вариант
 - 1 Перебор трех ходиков
 - 2 Играем за противника жадно еще три ходика
 - 3 Оценочная функция – линейная комбинация количества контролируемых клеток и клеток под ударом
- 5 Можно улучшить, но не было времени.

Хранение игр

- 1 Для хранения игр, пользователей, их настроек используется SQLite база данных
- 2 Хранится только история ходов и идентификаторы пользователей: по этой информации можно восстановить всю игру
- 3 Приостановленная игра также хранится в БД.
- 4 Возможна синхронизация с сервером (*не сделано*)

Хранение игр

- 1 Для хранения игр, пользователей, их настроек используется SQLite база данных
- 2 Хранится только история ходов и идентификаторы пользователей: по этой информации можно восстановить всю игру
- 3 Приостановленная игра также хранится в БД.
- 4 Возможна синхронизация с сервером (*не сделано*)

Хранение игр

- 1 Для хранения игр, пользователей, их настроек используется SQLite база данных
- 2 Хранится только история ходов и идентификаторы пользователей: по этой информации можно восстановить всю игру
- 3 Приостановленная игра также хранится в БД.
- 4 Возможна синхронизация с сервером (*не сделано*)

Хранение игр

- 1 Для хранения игр, пользователей, их настроек используется SQLite база данных
- 2 Хранится только история ходов и идентификаторы пользователей: по этой информации можно восстановить всю игру
- 3 Приостановленная игра также хранится в БД.
- 4 Возможна синхронизация с сервером (*не сделано*)

GCM

- 1 Для взаимодействия клиент-сервер используется GCM.
- 2 Эта система связи клиента и сервера не рассчитана на быстрые и большие сообщения, больше подходит для маленьких и не самых быстрых пакетов (например, push-уведомлений). Также она гарантирует доставку и маленькое потребление ресурсов. Именно то, что нам надо.

Игра по сети

- 1** Если была уже сохраненная игра, то посылаем на сервер сообщение с просьбой обновить информацию.
Иначе серверу посылается сообщение, что игрок готов играть.
- 2 Если игрок просит обновить игру, сервер ее обновляет
Если хочет с кем-то поиграть, помечает игрока как ожидающего.
Если уже есть ожидающие, то сервер создает между ними игру, посылает информацию о ней игрокам.
- 3 Когда игрок совершает ход, он посылает информацию об этом серверу, который пересылает ее другому игроку
- 4 Второй ход игрока на клиент может прийти раньше первого. Для этого у хода есть его номер, применяем ходы в порядке возрастания номеров

Игра по сети

- 1 Если была уже сохраненная игра, то посылаем на сервер сообщение с просьбой обновить информацию.
Иначе серверу посылается сообщение, что игрок готов играть.
- 2 Если игрок просит обновить игру, сервер ее обновляет
Если хочет с кем-то поиграть, помечает игрока как ожидающего.
Если уже есть ожидающие, то сервер создает между ними игру, посылает информацию о ней игрокам.
- 3 Когда игрок совершает ход, он посылает информацию об этом серверу, который пересылает ее другому игроку
- 4 Второй ход игрока на клиент может прийти раньше первого. Для этого у хода есть его номер, применяем ходы в порядке возрастания номеров

Игра по сети

- 1 Если была уже сохраненная игра, то посылаем на сервер сообщение с просьбой обновить информацию.
Иначе серверу посылается сообщение, что игрок готов играть.
- 2 Если игрок просит обновить игру, сервер ее обновляет
Если хочет с кем-то поиграть, помечает игрока как ожидающего.
Если уже есть ожидающие, то сервер создает между ними игру, посылает информацию о ней игрокам.
- 3 Когда игрок совершает ход, он посылает информацию об этом серверу, который пересылает ее другому игроку
- 4 Второй ход игрока на клиент может прийти раньше первого. Для этого у хода есть его номер, применяем ходы в порядке возрастания номеров

Игра по сети

- 1 Если была уже сохраненная игра, то посылаем на сервер сообщение с просьбой обновить информацию.
Иначе серверу посылается сообщение, что игрок готов играть.
- 2 Если игрок просит обновить игру, сервер ее обновляет
Если хочет с кем-то поиграть, помечает игрока как ожидающего.
Если уже есть ожидающие, то сервер создает между ними игру, посылает информацию о ней игрокам.
- 3 Когда игрок совершает ход, он посылает информацию об этом серверу, который пересылает ее другому игроку
- 4 Второй ход игрока на клиент может прийти раньше первого. Для этого у хода есть его номер, применяем ходы в порядке возрастания номеров

GSON

- 1 По сети данные надо пересылать в каком-то формате. GCM использует для этого JSON, но из него так легко все нужные данные не достать...

```
User cross = new User(data.getLong(WoVProtocol.CROSS_USER_ID), data.getString(WoVProtocol.CROSS_GOOGLE_TOKEN),
    data.getInt(WoVProtocol.CROSS_USER_TYPE), data.getString(WoVProtocol.CROSS_NICK_NAME_STR),
    data.getString(WoVProtocol.CROSS_NICK_NAME_ID), data.getInt(WoVProtocol.CROSS_COLOR_CROSS),
    data.getInt(WoVProtocol.CROSS_COLOR_ZERO), null);

User zero = new User(data.getLong(WoVProtocol.ZERO_USER_ID), data.getString(WoVProtocol.ZERO_GOOGLE_TOKEN),
    data.getInt(WoVProtocol.ZERO_USER_TYPE), data.getString(WoVProtocol.ZERO_NICK_NAME_STR),
    data.getString(WoVProtocol.ZERO_NICK_NAME_ID), data.getInt(WoVProtocol.ZERO_COLOR_CROSS),
    data.getInt(WoVProtocol.ZERO_COLOR_ZERO), null);
```

- 2 Но тут приходит на помощь GSON! Являясь, по сути, реализацией JSON от Google, предоставляет значительно более удобные возможности для сериализации/десериализации

```
User cross = gson.fromJson(jsonData.get(WoVProtocol.CROSS_USER), User.class);
User zero = gson.fromJson(jsonData.get(WoVProtocol.ZERO_USER), User.class);
```

GSON

- 1 По сети данные надо пересылать в каком-то формате. GCM использует для этого JSON, но из него так легко все нужные данные не достать...

```
User cross = new User(data.getLong(WoVProtocol.CROSS_USER_ID), data.getString(WoVProtocol.CROSS_GOOGLE_TOKEN),
    data.getInt(WoVProtocol.CROSS_USER_TYPE), data.getString(WoVProtocol.CROSS_NICK_NAME_STR),
    data.getString(WoVProtocol.CROSS_NICK_NAME_ID), data.getInt(WoVProtocol.CROSS_COLOR_CROSS),
    data.getInt(WoVProtocol.CROSS_COLOR_ZERO), null);

User zero = new User(data.getLong(WoVProtocol.ZERO_USER_ID), data.getString(WoVProtocol.ZERO_GOOGLE_TOKEN),
    data.getInt(WoVProtocol.ZERO_USER_TYPE), data.getString(WoVProtocol.ZERO_NICK_NAME_STR),
    data.getString(WoVProtocol.ZERO_NICK_NAME_ID), data.getInt(WoVProtocol.ZERO_COLOR_CROSS),
    data.getInt(WoVProtocol.ZERO_COLOR_ZERO), null);
```

- 2 Но тут приходит на помощь GSON! Являясь, по сути, реализацией JSON от Google, предоставляет значительно более удобные возможности для сериализации/десериализации

```
User cross = gson.fromJson(jsonData.get(WoVProtocol.CROSS_USER), User.class);
User zero = gson.fromJson(jsonData.get(WoVProtocol.ZERO_USER), User.class);
```

Сервер

- 1 Поскольку сервер, по сути, только хранит и пересылает данные, то его реализация довольно компактна и проста.
- 2 Реализация сервера состоит из доступа к БД, доступа к GCM, и обработки разных видов запросов.

Сервер

- 1 Поскольку сервер, по сути, только хранит и пересылает данные, то его реализация довольно компактна и проста.
- 2 Реализация сервера состоит из доступа к БД, доступа к GCM, и обработки разных видов запросов.

Сложности

1 API Google.

Наш проект сильно завязан на эти API. К сожалению, официальная документация, хотя и является самой часто обновляемой, содержала внутренние противоречия и неточности.

Официальные примеры, что ещё хуже, имели свою «точку зрения».

2 Постоянная сериализация данных.

Архитектура Android даёт мало возможностей для передачи данных в исходном виде, их приходится постоянно передавать в форме строк или байт. Также данные приходится сохранять при любом скрывании активности, причём не ясно, что с ним произошло.

3 Отличия структуры API Android и обычной Java для доступа к сети, БД,

Сложности

1 API Google.

Наш проект сильно завязан на эти API. К сожалению, официальная документация, хотя и является самой часто обновляемой, содержала внутренние противоречия и неточности.

Официальные примеры, что ещё хуже, имели свою «точку зрения».

2 Постоянная сериализация данных.

Архитектура Android даёт мало возможностей для передачи данных в исходном виде, их приходится постоянно передавать в форме строк или байт. Также данные приходится сохранять при любом скрывании активности, причём не ясно, что с ним произошло.

3 Отличия структуры API Android и обычной Java для доступа к сети, БД,

Сложности

1 API Google.

Наш проект сильно завязан на эти API. К сожалению, официальная документация, хотя и является самой часто обновляемой, содержала внутренние противоречия и неточности.

Официальные примеры, что ещё хуже, имели свою «точку зрения».

2 Постоянная сериализация данных.

Архитектура Android даёт мало возможностей для передачи данных в исходном виде, их приходится постоянно передавать в форме строк или байт. Также данные приходится сохранять при любом скрывании активности, причём не ясно, что с ним произошло.

3 Отличия структуры API Android и обычной Java для доступа к сети, БД,

Не успели

- 1 Красивый интерфейс клиента.
- 2 Честная сетевая игра – нет регистрации.
- 3 Синхронизация данных.

Не успели

- 1 Красивый интерфейс клиента.
- 2 Честная сетевая игра – нет регистрации.
- 3 Синхронизация данных.

Не успели

- 1 Красивый интерфейс клиента.
- 2 Честная сетевая игра – нет регистрации.
- 3 Синхронизация данных.

Спасибо за внимание

<https://github.com/LDVSOFT/War-Of-Viruses>