



Рисунок 4 — Пример работы приложения

Следует отметить, что объект должен быть дочерним к ImageTarget.

Заключение. Дополненная реальность уже меняет наше настоящее: виртуальные маски, охота за покемонами по городам и болотам, дети, стреляющие друг в друга не из деревяшек, а через экран телефона. Сейчас это уже реальность.

Список использованных источников

1. Vuforia Developer Portal [Электронный ресурс]. — Режим доступа: <https://developer.vuforia.com/>. — Дата доступа: 20.02.2019.
2. Создание контента дополненной реальности в Unity [Электронный ресурс]. — Режим доступа: <https://unity3d.com/ru/partners/vuforia>. — Дата доступа: 20.02.2019.

УДК 004.415.2.043

Д. А. Яроцкий

Учреждение образования «Барановичский государственный университет», Барановичи

КЛИЕНТ-СЕРВЕРНОЕ ПРИЛОЖЕНИЕ ДЛЯ ИГРЫ В «ДВАДЦАТЬ ОДНО»

Введение. Издавна люди предпочитали азартные игры. Азартная игра — игра с целью выигрыша денег или иных материальных ценностей. Азартная игра зависит в большей степени от случая, чем от навыка играющих, причём размер ставок назначается произвольно и может быть изменяем играющими, а главный интерес направлен не на процесс игры, а на её исход.

Основная часть. Двадцать одно — одна из самых популярных карточных игр в казино по всему миру. Большая популярность игры обуславливается простыми правилами, скоростью игры и наиболее простой стратегией в подсчёте карт. Игра пришла к нам из Соединенных Штатов, где имела название «Блэкджек». Целью игры является победа над соперником, набрав большее количество очков, но не более 21, иначе проиграет сам игрок.

Игра «Двадцать одно» популярна своей незамысловатостью. Партии проходят очень быстро, при подсчёте очков не возникает проблем.

Целью работы является создание приложения для реализации карточной игры «двадцать одно» с применением клиент-серверной архитектуры. Достижение поставленной цели требует решения следующих задач: создание сервера для обработки запросов; создание клиентского приложения для отправки запросов

на сервер; создание соединения между клиентом и сервером; контроль передачи хода сервером; создание алгоритма подсчета набранных очков и их сравнение.

При создании данного приложения была использована архитектура клиент-сервер. Суть данной архитектуры в том, что все данные хранятся на сервере. Клиент же посылает запросы, которые обрабатываются сервером. В результате сервер отправляет необходимую информацию клиенту.

Архитектура «Клиент-Сервер» представляет собой взаимодействие структурных компонентов в сети на основе определенных принципов организации данной сети, где структурными компонентами являются сервер и узлы-поставщики определенных специализированных функций (сервисов), а также клиенты, которые пользуются данным сервисом [1].

Преимущества клиент-серверных приложений [2]: 1) делает возможным, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети. Это позволяет упростить обслуживание вычислительной системы. В частности, замена, ремонт, модернизация или перемещение сервера, не затрагивают клиентов; 2) все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа; 3) позволяет объединить различные клиенты. Использовать ресурсы одного сервера часто могут клиенты с разными аппаратными платформами, операционными системами и т.п.

Также клиент-серверные приложения имеют ряд недостатков [3]: неработоспособность сервера может сделать неработоспособной всю вычислительную сеть; поддержка работы данной системы, требует отдельного специалиста — системного администратора; высокая стоимость оборудования.

Для реализации данной цели будем использовать IntelliJ IDEA 2018.

IntelliJ IDEA — интегрированная среда разработки программного обеспечения для многих языков программирования, в частности Java, JavaScript, Python, разработанная компанией JetBrains [4].

Для построения формы клиентского приложения была использована библиотека javax.swing. Java Swing — это легкий инструментальный графического интерфейса пользователя (GUI), который включает в себя богатый набор виджетов. Он включает в себя пакет, позволяющий создавать компоненты графического интерфейса для ваших приложений Java, и он не зависит от платформы.

Для связи сервера с клиентом была использована библиотека java.net.Socket. Благодаря данной библиотеке происходит общение между приложением-клиентом и приложением-сервером.

Схема работы приложения заключается в следующем: на сервере хранится алгоритм выдачи карт и передачи хода другому игроку. Клиент, сделав запрос на выдачу ему карт, получает ответ от сервера в виде выданной карты и ее стоимости в очках. При желании сделать пас, сервер передает ход другому игроку. После хода обоих игроков, сервер сравнивает результаты и выводит их игрокам.

Для начала игры необходимо ввести игровой псевдоним и нажать на кнопку «Играть». Если псевдоним не вводить, сервер назначит его сам. При попытке подключения к серверу, где еще не находится другой игрок, появится сообщение в центре игрового поля, оповещающее игрока об ожидании соперника. Для набора карт необходимо нажимать на кнопку «Еще» до тех пор, пока игрок не приблизится к нужной ему сумме очков.



Рисунок 1 — Приложение «двадцать одно»



Рисунок 2 — Передача хода противнику

После набор необходимой суммы очков игроку требуется нажать на кнопку «Пас» для передачи хода сопернику. На рисунке 2 можно увидеть, что надпись сверху экрана «Ваш ход» изменилась на «Ход противника».

Противник, набрав нужную ему сумму очков, также должен нажать на кнопку «Пас». После нажатия противником кнопки, сервер передает обоим игрокам информацию о набранных очках и подводит итоги. Узнать, сколько набрал очков игрок, а также его соперник, можно слева на игровом поле.

После окончания раунда игровое поле очистится от карт, а очки будут сброшены. Справа игрового поля ведется статистика раундов, где можно узнать какой по счету идет раунд, а также сколько игрок имеет выигрышей, проигрышей и ничьих.

Заключение. В ходе данной работы, было разработано сетевое приложение для игры в «двадцать одно». В результате работы были получены следующие результаты: было разработано приложение в IntelliJ IDEA 2018 на языке программирования Java; в программе реализована возможность игры на двоих игроков; приложение было написано на клиент-серверной архитектуре.

Список цитируемых источников

1. Клиент-серверная архитектура: особенности взаимодействия [Электронный ресурс]. URL: <http://fb.ru/article/101006> (дата доступа: 12.11.2018).
2. Шилдт, Г. Java8. Руководство для начинающих / Г. Шилдт — Спб.: БХВ — Санкт-Петербург, 2015. — 712 с., ил.
3. Корнелл Г. Java. Библиотека профессионала / Г. Корнелл — М.: Гелиос АРВ, 2012. — 813 с., ил.
4. Язык программирования Java [Электронный ресурс]. URL: <https://metanit.com/java/tutorial> (дата доступа: 03.12.2018).