UDC 004.4

IMPLEMENTATION OF THE CLIENT PART OF THE MOBILE MMO-GAME IN SPACE STYLISTICS

ROMAN KRASKO, KANSTANTSIN RAKHANAU
Polotsk State University, Belarus

The article deals with the formulation of the task for the client part of the mobile MMO-game in the space setting and the concept of building the client application. It also describes an approach to the software implementation of this application for mobile devices running the Android system and the main structural components of the client application.

Introduction. According to statistics in 2017 about 575 million people around the world were playing online games, which is 3% more than in 2016, and according to forecasts by 2021 the number of gamers in the world will grow to 645 million [1]. According to statistics, there are several popular genres: MOBA (Multiplayer Online Battle Arena, "multiplayer online battle arena"), shooters, MMO and MMORPG, strategy, online gambling [1].

Multiuser online games and games in space stylistics have shown their relevance on the example of such popular games for personal computers as EVE Online, Elite: Dangerous, Star Wars: The Old Republic, Star Citizen. Therefore, for mobile device users, a space MMORPG with quest and arcade elements is capable of gaining popularity.

The world of the game is a galaxy that consists of many systems, which comprise a number of planets that revolve around the star. On the planets, the player can take quests, buy or sell goods, improve or repair equipment. The protagonist acts as the captain of the spacecraft on which he moves around the game world. The player will also have a virtual currency which he can use to improve or repair equipment, complete the team, conduct trade, etc. For the performance of tasks and participation in the battles, the player is credited with the experience that is used to improve the characteristics of the character. Fighting with other players will take place in a step-by-step mode. In the game there will be 3 races: green, red, blue. When the game starts, the user can join one of the above races. Depending on the choice, the player will be given starting characteristics.

Game screens. Based on the content of the game in the mobile client, we can pick up the following game screens:

The "Start screen" is the start point for the application. When it starts, all game resources are loaded and initialized. Data can be obtained from both the server and the local cache. After the completion of the preparatory work, the player moves to the "Main Menu" screen.

The "Main Menu" screen allows you to navigate tothe "Registration and Authorization", "Settings", "About the application" screens, and also allows you to exit the application.

The screen "Registration and authorization". Log in is possible under Facebook, Google+, Twitter. If the player already exists, the user appears on the planet (the "Planet" screen), which he visited the last time. Otherwise, it goes to the "Create Character" screen.

The Settings screen provides the ability to configure notifications, vibration, music volume and sound effects.

The "About application" screen contains a summary of the current version of the game.

The "Character Creation" screen allows the user to select the name, sex, race and icon of the character. After choosing the characteristics player appears on a random planet in the galaxy.

Screen "Planet". Allows you to switch to the "Store", "Workshop", "Bulletin board", "Bar", "System", "Ship" screens.

The "Shop" screen allows you to buy components for the ship using virtual currency.

Screen "Workshop" allows you to repair and modify the components for the ship.

The "Bulletin board" screen contains a list of the quests available to the player for execution.

The "Bar" screen allows you to hire a team for virtual currency.

The "System" screen represents a number of planets that rotate around the star. The ship is controlled using a virtual joystick in the lower left corner. The speed and maneuverability of the ship depends on its technical characteristics. At any point in the game, you can go to the screens "Galaxy", "Ship", "Radar". When you click on someone else's ship, you can open the screens "Trade", "Fight", "Chat".

The screen "Galaxy" is a mini map of the galaxy and allows you to fly to any system that is available to the user. The range of flight depends on the technical characteristics of the ship and the skills of the team.

Screen "Ship" allows you to view the ship's components, the characteristics of the character and the team, a list of active quests.

The Radar screen is a mini map of the current system.

The screen "Fight". Fighting with other players will take place in a step-by-step mode, the essence of which is reduced to the sequential performance of certain actions, for example, the use of weapons, repair droid, etc.

The "Trade" screen allows you to exchange goods between two real players.

The "Chat" screen presents a text chat between two real players.

As for the basic functionality, which is not tied to a specific form, we can point out:

– loading and displaying textures, sprites, fonts and particles, building a user interface;

– loading sound resources, playing background music and sound effects;

– Network communication with the server will be carried out using the protocols UDP, TCP and HTTP. The first is for real-time data transmission, where the maximum speed is important and packet loss is acceptable. It's used interacting with players in the system during the flight. The TCP protocol is used for reliable data transmission, where packet loss is unacceptable. It's used in turn-based combat, performing quests, exchanging goods with other players, text chat, etc. HTTP protocol is used to interact with the server outside the game logic: in-game payments, authorization, registration.

– data storage;

– processing of client-side delays: client-side interpolation, client-side prediction, client move prediction and replay.

– receiving and processing notifications from the server, which inform the user about various events occurring in the game;

– text chat for communication of users during the game;

– purchase of virtual currency for real money;

– integration of bonus advertising (getting any virtual rewards for viewing ads).

Components of the mobile client. To implement the functionality of the mobile client, the MVP pattern (Model-View-Presenter) is selected. It allows you to separate business logic from display, which will facilitate automatic unit testing, editing and code support.

The work with graphics and sound will be implemented using LibGDX, which is a Java framework that provides a cross platform API for developing 2D games and real-time applications [2]. This framework has good documentation, provides a cross-platform API for in-game payments, makes it possible to use both low-level (OpenGL) tools for working with graphics, and high-level, completely free and has open source code.

To work with the network, the Netty framework will be used. This framework is compatible with LibGDX, allows you to work with UDP, TCP and HTTP, has good documentation and is completely free [3]. And Protocol Buffer will act as the protocol of serialization.

Data storage is implemented by the Realm library. The library is a free embedded NoSql database for mobile devices, which has a high speed of operation [4].

Work with notifications, advertisements and social networks is performed through the Firebase SDK, which provides the following tools:

Firebase Cloud Messaging is a cross-platform solution for working with notifications for Android, iOS and web applications, which is completely free.

Firebase Auth is a service that allows you to authenticate via Facebook, GitHub, Twitter, Google or an email address.

Adwords is a service for integrating advertising into mobile applications.

The figure shows a component diagram that represents the partitioning of the mobile client into structural elements.

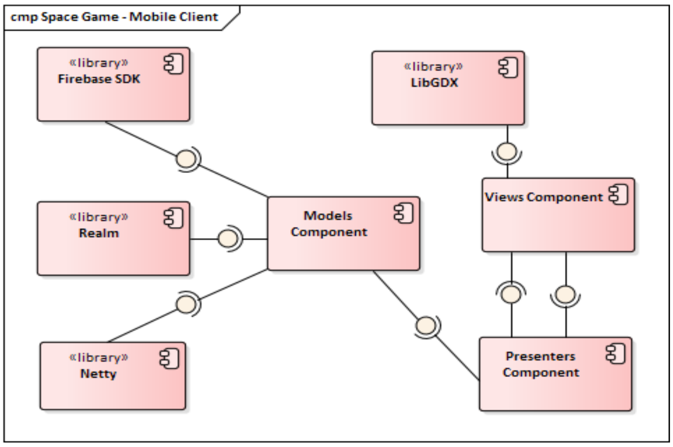

Fig. Mobile client components chart

The conclusion. The task for the client part of the mobile MMO game in the space setting is formulated. An approach to the software implementation of this application for mobile devices running the Android system is proposed. The architectural pattern is MVP. LibGDX was suggested as a tool for working with graphics and sound. Network interaction with the server is carried out with the help of the Netty framework. The storage of local data is realized by the Realm library. Integration with Google services is provided through the Firebase SDK.

REFERENCES

1. Тренды онлайн-игр в 2017 году и дальше [Electronic resource] // Plarium © 2018. – Mode of access: https://plarium.com/ru/sovety-mmo/trendy-onlayn-igr-2017/. – Data of access: 14.01.2018.
2. LiGDX Features [Electronic resource] // LibGDX © 2018. – Mode of access: http://www.libgdx.ru/2013/08/goals-features.html. – Data of access: 14.01.2018.
3. Netty [Electronic resource] // Netty © 2018. – Mode of access: https://netty.io/ © 2018. – Data of access: 15.01.2018.
4. Realm Features [Electronic resource] // Realm © 2018. – Mode of access: https://realm.io/products/realm-database/. – Data of access: 14.01.2018.