

SpaceWar Friends Menu Update

BY
ALAN MEYGHAN

Synopsis

In this document, I present the C++ code I developed for the Friends menu in the SpaceWar sample project included as a part of the SteamWorks SDK.

The intention of this work is to offer users additional Steam friends presence information and functionality and to provide developers with the information required to implement these features into their projects as they plan to integrate and publish their games on Steam.

For a quick and convenient browsing experience, C++ functions and contents highlighted in bold in this document are clickable. Clicking on each item will navigate to the desired item.

This addition is developed using Steamworks SDK version 151.

Table of Contents

Introduction	4
Preview	4
Background	5
Purpose	5
Project Scope	6
Code Convention	6
Code Efficiency and Impact	6
Hardware Specification	7
Code Information	7
Asymptotic Notations	7
Item 1: FriendsListMenuItem_t	8
SpaceWarClient.h	8
Item 2: Friends Menu	9
Friends.h	9
Friends.cpp	9
AddFriendsToMenuByStatus	10
SetFriendStatus	10
OnMenuSelection	11
Correction Submission	11
Conclusion	12

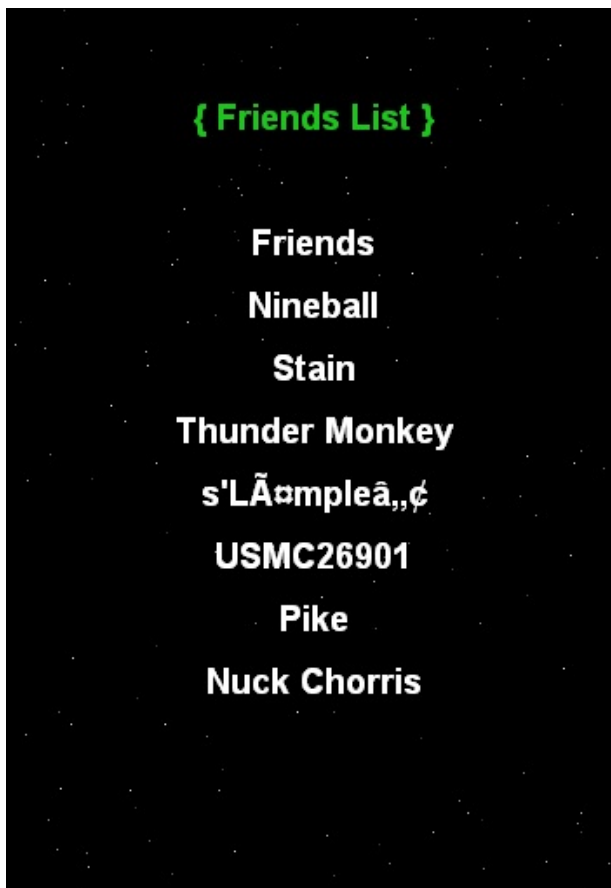
SpaceWar Friends Menu

Introduction

The introduction provides a visual representation of this update and explains the background and the purpose behind it.

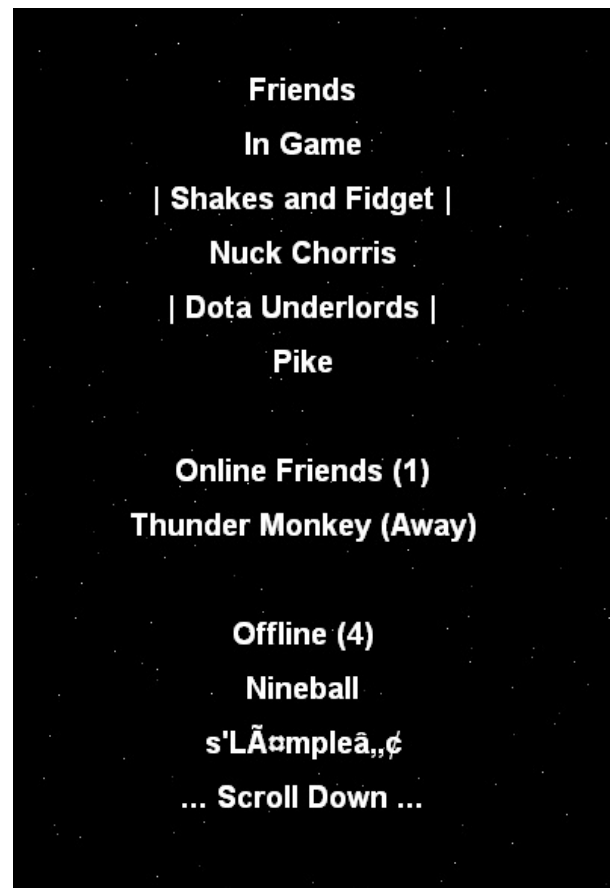
Preview

BEFORE



The Original Friends List Menu

AFTER



The Updated Friends List with Status Information

Background

The friends displayed in the Friends menu section of SpaceWar are sorted based on the SteamID number with no additional menu selection options. This update rearranges the friends list menu in alphabetical order and adds functionality provided by the SteamWorks SDK, matching some the features of the Steam client.

Purpose

- Sort the friends list by name, matching the Steam client friends window.
- Provide additional information about a friend's Online/Offline status and In-game presence, grouping friends in the same game under the same game.
- Add basic interactions available in the Steam client using the friends API.

Developers choosing to add friend status, game information and friend interaction functionality to their projects can use the information presented in this update to learn about possible options provided by the friends API.

Project Scope

This section describes the development logic and presents the information gathered by diagnostics tools to ensure that this update imposes no major memory implications to the original project.

Newly-introduced and modified C++ variables and methods are documented as itemized sections. The time complexity of each new function is explained in the closing argument of the corresponding function explanation.

Code Convention

Though there are several ways of developing this update, in conformance with the style used in the original project, I have used pre C++11. In areas where modern C++ provides performance improvements and better code readability, modern C++ code has been added as a comment but not used in the final product. To keep the integrity of the project class methods, all newly added member functions are void and have no return types. The code has been tested in Microsoft Visual Studio 2019, using the following settings:

Platform Toolset: Visual Studio 2017 C++ Language Standard: Default

Window SDK version: 10.0.19041.0 DirectX SDK Version: June 2010 Build: 9.29

Code Efficiency and Impact

Observing the memory usage using Microsoft Visual Studio 2019 Diagnostics Tool, the update exhibits minimal additional memory overhead (≈ 100 Bytes per friend, if the friend is online and in-game). Entering the Friends menu, the amount of memory used to update and show the newly updated friends menu matches the memory consumption of the project before and after (≈ 32 MB).

After rebuilding the project, the memory usage is increased to ≈ 46 MB and upon exiting and reopening Visual Studio, the memory is freed and reset to the original ≈ 32 MB.

Hardware Specification

The hardware used to perform the tests, compile and run the code:

Processor: Intel(R) Core(TM) i7-4790K RAM: 32 GB DDR-3

Operating System: 64-bit Windows 10 Pro Version 20H2

Code Information

The code added for this update to produce the final results can be found in:

- **SpaceWarClient.h:** Starting from line 94, the **FriendsListMenuItem_t** struct has been expanded to include additional information about a friend to initialize each menu item in the CFriendsListMenu class. You can find more information about these changes in the **FriendsListMenuItem_t** item.
- **Friends.h:** At line 37, the STEAM_CALLBACK delegate using the PersonaStateChange_t callback pointer is added to the CFriendsList class for immediate registration of a friend status change to update the friends list menu accordingly.
- **Friends.cpp:** The friends list update starts at line 91 of this file which includes two new methods, the details of which are discussed in **Item 2: Friends Menu**.

Asymptotic Notations

Time complexities presented in this document are based on standard Big-O notations used in computer science and are described with the worst-case as the basis of calculations:

$O(1)$ = Constant $O(\log n)$ = Logarithmic $O(n)$ = Linear

$O(n \log n)$ = Linearithmic

Item 1: FriendsListItem_t

SpaceWarClient.h

To provide the ability of organizing the friends list conveniently, matching the Steam client, a number of variables have been added to this struct to be used by each Friends menu item for initialization and status comparison.

```
struct FriendsListItem_t
{
    enum EFriendsMenuItemCommand
    {
        k_EFriendsListItem,
        k_EFriendsListItemSendMessage,
        k_EFriendsListItemViewProfile,
        k_EFriendsListItemAcceptFriendRequest,
        k_EFriendsListItemIgnoreFriendRequest,
        k_EFriendsListItemRemoveFriend,
        k_EFriendsListItemBack
    };

    CSteamID m_steamIDFriend;
    EFriendsMenuItemCommand m_eCommand;
    EFriendRelationship m_eSteamFriendRelationship;
    FriendGameInfo_t m_steamIDFriendGameInfo;
    bool m_bIsFriendOnline = false;
    bool m_bIsFriendInGame = false;

    //Sort comparator
    bool operator < ( const FriendsListItem_t& rhs ) const
    {
        return _stricmp( SteamFriends()->GetFriendPersonaName( this->m_steamIDFriend ),
                        SteamFriends()->GetFriendPersonaName( rhs.m_steamIDFriend ) ) < 0;
    }
};
```

This change allows each item in the friends menu to be initialized with with additional presence information and state to be used and displayed in the friends menu.

The overloaded operator < is used as the comparator argument passed into std::sort to alphabetically sort the friends list, matching the Steam client. Additionally, the EFriendsListItemCommand enums make it possible for items to receive different commands and perform actions based on the corresponding command on which they are initialized through.

Item 2: Friends Menu

Friends.h

STEAM_CALLBACK delegate is added for immediate registration of a friend's status change, game update and incoming/outgoing friend requests to be displayed in the friends menu.

```
STEAM_CALLBACK( FCFriendsList, OnPersonaChanged, PersonaStateChange_t )
```

The callback pointer in the OnPersonaChanged method defined in Friends.cpp at line 378, performs a user status change, ignoring the user self and updating the the friends list accordingly by rebuilding the friends list.

Time Complexity: This functions performs the status check in $O(1)$ time.

Friends.cpp

The first addition can be observed at line 91. By creating a vector container to hold the friends already in a user's friends list, this container is passed into the newly created method **AddFriendsToMenuByStatus** to sort, check for status change and perform the required actions necessary.

Added Member Methods:

Line 121: `void AddFriendsToMenuByStatus(std::vector<FriendsListMenuItem_t> *vecMenuFriends)`

Line 226: `void SetFriendPersonaStatus(const FriendsListMenuItem_t &menuItemFriend,
const &steamFriendPersonaState, const char *pszFriendName)`

Modified Member Methods

Line 301: `void OnMenuSelection(FriendsListMenuItem_t selection)`

AddFriendsToMenuByStatus

This void method takes a pointer to a vector of menu friends as an argument and is entered at line 122 by sorting the received argument at line 124 using `std::sort`, taking $O(n \log n)$ time.

Traversing through the input sequence, it adds each friend to their corresponding container according to their status, taking linear $O(n)$ time, with the constant as the number of friends.

To keep the integrity of the class structure, this method performs the operations using raw loops. I have included a better, cleaner and faster comment by employing modern algorithms on iterator pairs and lambdas at line 130 through 139, which requires less memory allocation and performs the operations by back-inserting the desired content, without manually iterating through the input sequence.

Subsequently, the function performs a check on friend containers (In-game, Online, Offline). If the container contains any friends, the information is then added to the corresponding menu and displayed on the screen. For the In-game friends, an additional check is performed to group friends in the same game under the same game name (line 178 through 183) taking logarithmic $O(\log n)$ time.

Time Complexity: Adding up the function algorithms and iterations, this function will run in $O(n \log n)$ time.

SetFriendStatus

This void method takes three arguments, a menu friend item, a friend state and a friend name and displays the corresponding friend status. If the friend is online, presence information is appended to the friend's name by checking a switch case and for offline friends, only the friend's name is displayed in the menu.

Time Complexity: This function deals with name and status assignments and takes $O(1)$ time.

OnMenuSelection

Newly added actions to this method include:

- Checking friendship status, if we sent a friend request, the function returns.
- Prompt an overlay window to accept or ignore a friend request if we received a request.
- Provide the following options If the friend is already in our friends list, matching some of the options presented in the Steam client:
 - Send Message : Opens the Steam overlay to send a message to the friend.
 - View Profile : Opens the Steam overlay to view the friend's Steam page.
 - Remove as Friend: Prompts the user for removal of the selected friend.
 - Go back : Returns to the Friends menu.

These menu options use the enum command switches from the member variable `m_eCommand` within the `FriendsListMenuItem_t` struct.

Time Complexity: The actions performed in this function include assignments and the switch statement, checking for a single command at a given time, taking $O(1)$ time.

Correction Submission

SpaceWarClient.cpp line 1012, is related to the friends menu selection. The purpose stated as "Viewing a leaderboard" is a repeated line from the previous function at line 1003 and should be changed to viewing "the Friends list"

Conclusion

This update provide developers with the information required to include these additional features into their projects as they prepare to publish their games on Steam.

These features are among those commonly used in most games on the Steam platform. The information provided in this document and the code added to the project will educate developers with ways to implement these additional features provided by the SteamWorks SDK.