

Client Server iOS in Player versus Player (PVP) of “Borneo Snap”

Reza Andrea

Software Engineering, Politeknik Pertanian Negeri Samarinda
Email: reza@politanisamarinda.ac.id

Tommy Bustomi and Muhammad Muhsan

Politeknik Negeri Samarinda, Computer Engineering, STMIK Widya Cipta Dharma
Email: tbustomi@gmail.ac.id, muhammad_muhsan@icloud.com

Received: 25 December 2020; Revised: 05 January 2021; Accepted: 16 January 2021; Published: 08 February 2021

Abstract: Borneo Snap is a Kalimantan’s animals snap card game. Play proceeds with the players taking it in turns to remove a card from the top of their deck and place it face-up on a central pile. If two cards placed consecutively on the pile are identical (same picture) then the first player to shout "Snap!" will get all of it. This game is built for the iOS platform. Game Player Versus Player (PVP) Borneo Snap using peer-to-peer API from Game Kit (XCode Framework) and Wi-Fi or Bluetooth, but actually, Borneo Snap uses client-server architecture model, each player in a player versus player game session only can communicate with server intermediaries. If the player sends data updates when playing cards to other players, this data update will first be through the server, then forwarded to all other players. The result of this research, with client-server framework Borneo Snap can be played by more than 1 player and more iOS gadget too

Index Terms: Player versus Player, Borneo Snap, iOS, Wi-Fi, Bluetooth, Client Server, GameKit.

1. Introduction

One of the most popular game features is the Player Versus Player (PVP) feature [1-3]. This feature uses wireless networks such as Bluetooth, Wi-Fi or the internet. This system allows players to play against other players at the same time from different gadgets or places [2]. Borneo Snap is a snap card game with native animals of Borneo as a theme. This game was developed for the iOS platform which can be played with a maximum of 4 players (multiplayer). One technique that can be used is to use client-server framework technology. The use of peer-to-peer API allows each player to be a server or client in multiplayer games..

Peer-to-peer Application Programming Interface (API) is a framework provided by the iOS SDK Game Kit to connect iOS gadgets with Wi-Fi and Bluetooth [4,5]. Based on its architecture, peer-to-peer allows every iOS gadget to be connected in the same network. Although peer-to-peer, this framework can be converted into a client-server scheme for certain game systems.

This Borneo Snap PVP game is inspired by a card game called "Snap!" published by Metintuzgu in 2012 in the app store [6]. Borneo Snap game was developed to introduce local cultural content, namely native animals of Borneo such as Irrawaddy dolphin, orang-utan, sun bear, hornbill, and others. This game can be played on various iOS platforms, namely the iPhone, iPad, and iPod Touch.

In the snap game, players take turns removing a closed card from the deck of cards into the central pile and opening it [7]. If there is the same card in the central pile, the player who first presses the "snap" button will get all cards from the pool of cards. If the player presses the "snap" button, but the card is different from the central pile, the player must give one card to each other player. The player who loses is the player who first runs out of cards in a deck of cards. While the winner is the last player to get all cards from all other players [8].

Studies on the development of snap card games have been performed [9-12], but most games do not support the PVP [9-11]. So that players can only play with computers which are sometimes easy and sometimes difficult [13]. Several studies showed that several games on the Android platform support the PVP feature, but players must play the game in one gadget together [14]. So, that it becomes an obstacle for many players to compete to press the "snap" button on the gadget screen that is not big.

In this study, Borneo Snap has local cultural content by introducing Borneo's endangered animals, so that players can gain knowledge while playing. PVP feature with client-server framework makes this game can be played by more

than 1 player with more than 1 gadget too and with peer-to-peer API makes every gadget player can be a server (host) without having to be charged an internet connection.

This study has a novelty when players are connected to the same Wi-Fi or Bluetooth network, the players do not need to compete to press the "snap" button on 1 gadget screen only. If there is one player as a server (host), then the other players who join will become clients. This system is called client-server [15]. This study contributes to the field of networking with a card game that has local cultural content. This multiplayer game utilizes a client-server framework through Wi-Fi and Bluetooth network technology

2. Methodology

One the stages of system development can be described as follows:

1. Analysis

Exploring the system of the snap game is very useful to find out the causes and consequences caused by network problems so that it will produce a deeper analysis of the PVP game, especially on the objectives and functions to be performed.

2. Design

Borneo Snap PVP game paid attention to connections between players, network architecture, data packages and graphics in multimedia including text, sound, animation, and images so that gameplay can look attractive. The design was translated into XCode.

3. Implementation

The result of design was ready-to-use software.

4. Testing

Various kinds of testing related to PVP Snap Game were carried out so that it was expected to be in accordance with the desired results.

A Use Case Diagram of Borneo Snap

Use Case Diagram of Borneo Snap game explains that the system has 2 to 4 actors as users namely player 1 (as host), player 2 (join game), player 3 (join game) and player 4 (join game). Before starting the game, the player must choose the type of role between being a server (host) or client (join game), which will be selected in the initial menu of the game as in Figure 1.

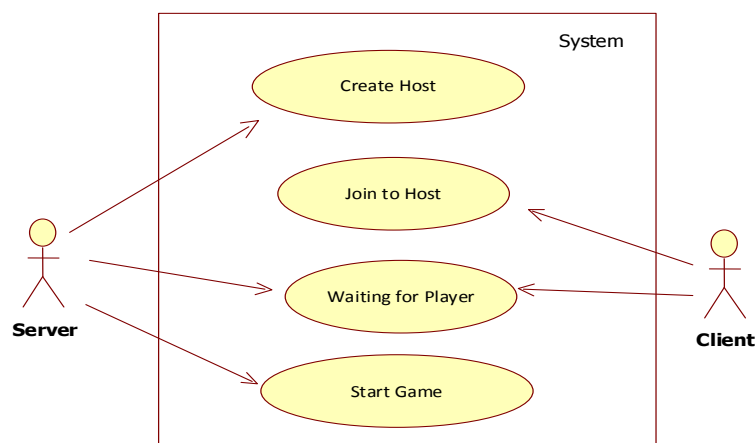


Fig. 1. Use Case Diagram connectivity for PVP, 1 player as a server, and 1, 2 or 3 players become clients

Next, the player as a join game will choose which server table the host wants to choose from, then the player who acts as the host presses the start button then the game will start immediately by distributing all cards to each player. The game will start when all cards have been distributed to all players, the player will be randomly selected to start opening the card for the first time. Use case diagrams of Borneo Snap game can be seen in Figure 2.

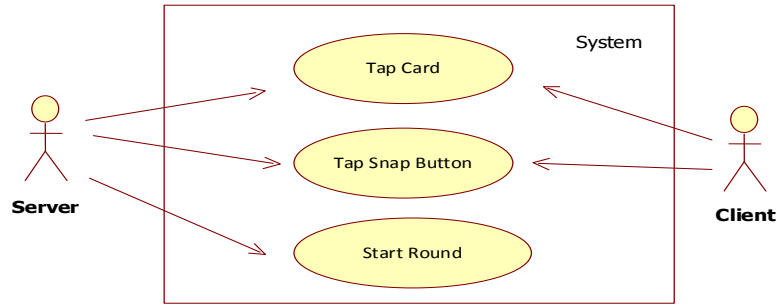


Fig. 2. Use Case Diagram playing Borneo Snap Card, server will start first tapping the card followed by the client in sequence

All players have a turn to tap cards clockwise, players can also press the snap button at any time when the game has started when the game has finished, the server can start a new game round and the game will restart. If three of the four players don't have a card anymore, the remaining one player wins and the program displays the results.

B Client-Server Architecture

Figure 3 shows the Borneo Snap PVP game with a client-server architecture model.

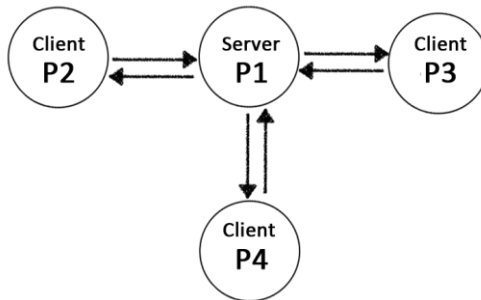


Fig. 3. Borneo Snap client-server architecture with server as P1 and 3 clients as P2 to P4

In this stage, the connection process of the game will be made, when the player presses the "host game" button, the device used will broadcast services to other players within range and the player who presses the "join game" button on the device will immediately search for all services covered within reach by using Wi-Fi or Bluetooth. The process of server connections can be seen in Figure 4 and the client connection process can be seen in Figure 5.

Figure 4 shows that at idle, the server will broadcast service and receive connections from the client because the Borneo Snap game can only be played by 2 to 4 players, so when the connected player reaches 3 players, the server will reject all requests connection from the client because the player has reached 4 people (maximum number of players).

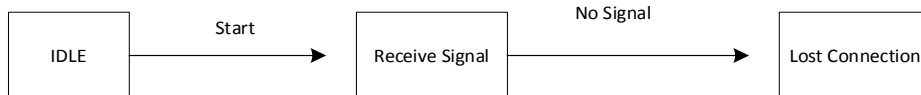


Fig. 4. Server connection diagram performing broadcast service

Figure 5 shows that the client can do two different things: find a server to join and when it determines the server that is wanted, the client will send a request to connect to the server, when the server has received a connection request from the client, the server will first check whether the number player is still less than 3, if the condition is met, then the connection will be accepted and this makes the client and server become connected.

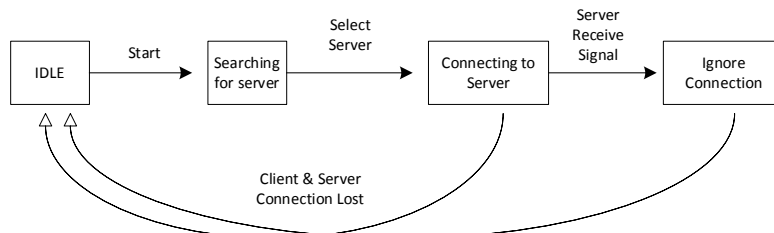


Fig. 5. Client connection diagram connecting with the server

In the development of applications that use the network (LAN or the Internet), the important thing to note is the possibility of errors that occur because the connection is not running as expected [16,17] as in connections that could be lost at the process of connecting. The Borneo Snap PVP game uses 4 scenarios to overcome unpredictable connections, namely:

1. *No connection*

A condition where the device used by the player does not turn on Wi-Fi or Bluetooth which will be used to connect to other players.

2. *Connection failed*

A condition where communication between client and server fails which can be caused by the server being lost out of reach.

3. *User disconnected*

A condition where one of the players closes the Borneo Snap application, then the player will be disconnected from the network.

4. *Server disconnected*

A condition where the server that provides Borneo Snap is disconnected from the application so that all players will also be disconnected from the game.

3. Result and Discussion

Borneo Snap is a mobile game (iOS) with a card games category. This game has simple game rules and displays content designed for children up to adults with an age range of 6 years or more. Players take turns removing one card in the available card stack if there is an identical card then the player who first presses the "snap" button will get all the cards on the deck of cards, but if the player presses the "snap" button even though there is no card identical, the player must give one card to each player. The player who can get all the cards from other players is declared the winner. The game cards feature endangered Bornean animals which can be seen in Figure 6, such as Orangutans, hornbills, and sun bears.



Fig. 6. Borneo Snap cards depict ethnic Kalimantan animals

This game trains the ability to concentrate to see the card and quickly press the "Snap!" to win the game. In Borneo Snap there are several rules, namely:

1. A multiplayer game, so players must choose between being a host (server) or join (client).
2. Borneo Snap can be played by two to four players.
3. The player will get all the decks of cards when pressing the "snap" button when there is an identical card on top of the deck.
4. If there is a player who presses the "snap" button when there is no identical card, then the player will be fined by handing one card to each player.

- The game is declared complete if 3 out of 4 players no longer have a card, and the player with the most cards wins.

There are 2 types of games in Borneo Snap, when running this game, the first displayed View Controller is Main View Controller, then the player is required to choose as a host that is providing Borneo Snap service for other players or as a join that is the player will be able to join the Borneo Snap service provided by the player who acts as the host. There are several menus in this game interface.

A. Main View Controller

The Main View Controller that can be seen in Figure 7 is a menu with UI View Animation. There are 3 buttons on this menu, namely, Host, Join, and Play. If the player presses the "Host" button then it will enter the Host View Controller and the player will make the device as a server, if the player wants to act as a client then press the "Join" button. Borneo Snap is also equipped with a play together feature in 1 gadget that is by pressing the "Play" button in the middle so that 4 players can play together in 1 gadget without any network connection.

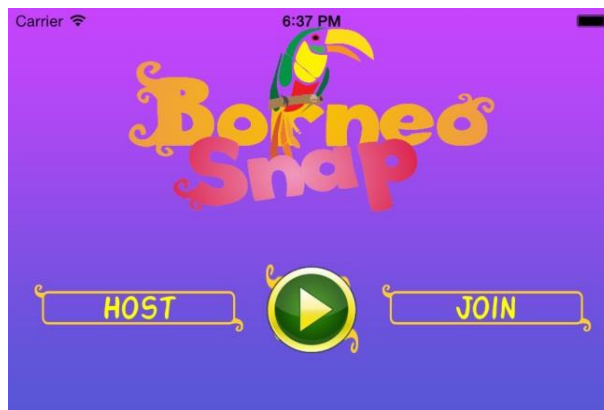


Fig. 7. Main View Controller Borneo Snap's initial interface

B. Host View Controller

Host View Controller is a menu used by the player as a host, which is broadcast service of Borneo Snap with gesture recognizer. This can be seen in Figure 8. "Simulator" is a player that acts as a server, while "Aldy's iPhone" and "Muhammad Muhsan's IPAD" are clients. The host can accept the number of players as clients for a maximum of 3 devices, and only a player who acts as a host can start the game. The number of players is limited to only 4 players because the gadget screen will be full of player cards, and not all iOS gadgets have a large inch screen size.



Fig. 8 The Host View Controller interface is accepting 2 players as clients

C. Join View Controller

Join View Controller is a menu used for the player as client, which can be seen in Figure 9. In this interface, players who act as clients can see a list of hosts that are broadcast by the server. Clients can choose which host they want, then wait for other players to join and wait for the server to start the game. In Figure 9, the device "Muhammad Muhsan's IPAD" as a client is receiving a broadcast signal from a server device called "Simulator".



Fig. 9. The Join View Controller interface is receiving 1 broadcast server

D. Game View Controller

Game View Controller is a display of the gameplay of Borneo Snap which can be seen in Figure 10 where there are 3 players playing cards. Client-server framework will divide the player who acts as a server as P1 (player 1) and client as P2, P3, and P4. The server and client can see the cards open in rotation. The player who first pressed the "Snap!" Button, when there come out 2 cards with the same picture, will get these cards. Players who first run out of decks of cards will lose. Losing players can still watch the game through a client-server connection. The losing player can also disconnect from the host.

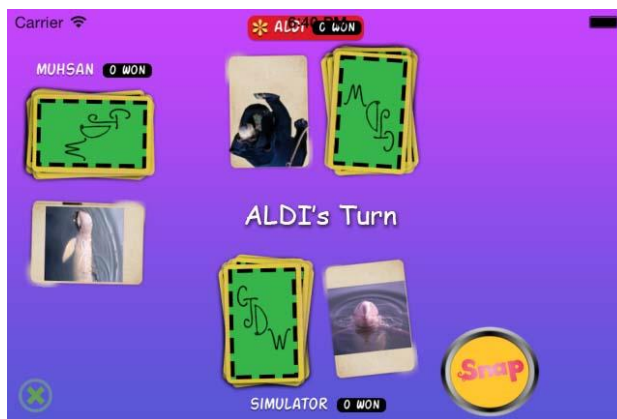


Fig. 10. The Game View Controller interface displays 3 players playing cards

The final result of the game will display the player who won as can be seen in Figure 11, where the player with a large deck of cards will come out victorious.



Fig. 11. The Game View Controller interface displays 3 players playing cards

3.1 PVP Client-Server framework

Connected client-server frameworks can be divided based on connection types as follows:

1. Server Connection

Server Connection is an API that has a dual function which is to start broadcast service of Borneo Snap and accept connections from clients (other players), functions used in server connections can be seen in the script:

```
if (_ConditionServer == ConditionServerIdle) {
    _ConditionServer = ConditionMenerimaConnection;
    _playerConnected = [NSMutableArray arrayWithCapacity:self.maxPlayer];
    _session = [[GKSession alloc
        initWithSessionID:sessionID
        displayName:nil sessionMode:GKSessionModeServer];
    _session.delegate = self;
    _session.available = YES;
}
```

From the function above it can be explained that, before the listing is run, then the server condition is checked in the server idle state (`_ConditionServerIdle`), then the server condition will be changed to the condition of accepting the connection, after that the connected player array will be filled by the list of players who will join the service of Borneo Snap with the `maxPlayer` variable to check when the number of players exceeds the specified number of players (3 players as clients), while the `GKSession` object functions to operate as a server (`GKSessionModeServer`) or in other words, this algorithm will only broadcast services of Borneo Snap with unique identification is the `sessionID` parameter. Delegate functions as a means of communicating the server connection algorithm with other classes and is closed by `GKSession available = YES` which will immediately start the broadcast service of Borneo Snap.

2. Client Connection

Client Connection is an API that functions to search for services of Borneo Snap, the function of client connections can be seen in the script:

```
if (_ConditionClient == ConditionClientIdle) {
    _ConditionClient = ConditionClientSearchServer;
    _serverAvailable
        [NSMutableArray arrayWithCapacity:10];
    _session = [[GKSession alloc
        initWithSessionID:sessionID
        displayName:nil
        sessionMode:GKSessionModeClient];
    _session.delegate = self;
    _session.available = YES;
}
```

From the above function it can be explained that, before the listing is run, the condition of the client will be checked in the condition of the client idle condition (`_ConditionClientIdle`), then the client condition will be changed to the condition of looking for service of Borneo Snap, the list of services of Borneo Snap detected will be stored on an array of available servers, while the `GKSession` object functions to operate as a client (`GKSessionModeClient`) in other words this API will only search for a service of Borneo Snap that uses a unique identification that is the `sessionID` parameter. Delegate functions as a communication tool for client connection algorithms with other classes and is closed by `GKSession available = YES`, which will immediately search for the service of Borneo Snap.

3. Connection Error Client-Server

For each application that uses a network, it is necessary to create a mechanism to resolve the disconnected condition of the network, the function of the client-server connection error can be seen in the script:

```
(void)session:(GKSession *)session peer:(NSString *)peerID ConditionChange:(GKPeerConnectionState)state
    case GKPeerStateDisconnected:
        if (_ConditionClient == ConditionConnected) {
            [self DisconnectedFromServer];
        }
        break;
(void)DisconnectedFromServer{
    NSLog(@"ConditionClient != ConditionClientIdle, @"Condition salah");
    _clientState = ConditionClientIdle;
    [_session disconnectFromAllPeers];
}
```

```

_session.available = NO;
_session.delegate = nil;
_session = nil;
_serverAvailable = nil;
[self.delegate matchmakingClient:self
DisconnectedFromServer:_serverPeerID];
_serverPeerID = nil;
}

```

This function process occurs after the connection process, where one player is disconnected from the network due to various factors such as a server disconnected from the application that causes all players to be disconnected from the server, in other conditions if the connection process has occurred, the server and client are connected but because the distance is too far away can cause an error in the connection. If one player (client) is disconnected from the network, the GKPeerStateDisconnected from the GameKit framework will call DisconnectedFromServer function which will then clear the contents of _serverPeerID which stores information about the connected server, and the final process is to notify the delegate of the client that the client has been disconnected from the server. Further explanation about GKPeerState connectivity can be seen in Table 1.

Table 1. GKPeerState Numeric Value Code List

State	GKPeerState	Explanation
0	GKPeerStateAvailable	Conditions can receive / seek service
1	GKPeerStateUnavailable	Conditions can't receive / seek service
2	GKPeerStateConnected	Condition is connected
3	GKPeerStateDisconnected	Connection condition lost
4	GKPeerStateConnecting	Conditions for connection between client and server

3.2 PVP Connectivity Testing

Testing was performed on the client-server connection API, the connection function is lost on the data transfer between the client and server that occurs in the game.

The debug output below shows the sample when the host (server) is broadcasting the Borneo Snap service. Based on the function created, the server will accept connections from clients when there are less than 3 players. It can be understood that the initial state is 4 (Table 1 state 4, the condition of connecting between client and server), after the server accepts the connection the state will change to state 2 (already connected).

```

2019-03-06 16:59:18.753 BorneoSnap[357:907] Connection Server: peer 1626531790 Condition state 4
2019-03-06 16:59:18.756 BorneoSnap[357:907] Connection Server: request Connection from peer 1626531790
2019-03-06 16:59:18.762 BorneoSnap[357:907] Connection Server: Connection received from peer 1626531790
2019-03-06 16:59:18.806 BorneoSnap[357:907] Connection Server: peer 1626531790 Condition state 2

```

Testing was carried out on client to host connections. The debug output below shows the debug output sample at the client's condition, the client's condition will initially be at position 0 (Table 1 state 0, where this condition indicates that the client has detected a service of Borneo Snap in the coverage area), then if the player has selected server from the list of available servers, the client algorithm will connect to the selected server, and if the server accepts the connection the client condition will change to 2 (client and server conditions have been connected) then after that the client will wait until the server starts the game.

```

2016-03-06 17:05:50.195 BorneoSnap [14094:60b] BTM: attaching to BTServer
2016-03-06 17:05:55.121 BorneoSnap [14094:60b] Connection Client: peer 2033237104 Change state 0
2016-03-06 17:06:00.343 BorneoSnap [14094:60b] BTM: attempting to connect to service 0x00000800 on device "Muhsan iPhone" 64:B9:E8:03:B4:BA
2016-03-06 17:06:00.508 BorneoSnap [14094:60b] BTM: connection to service 0x00000800 on device "Muhsan iPhone" 64:B9:E8:03:B4:BA succeeded
2016-03-06 17:06:04.619 BorneoSnap [14094:60b] Connection Client: peer 2033237104 Change state 2
2016-03-06 17:06:05.180 BorneoSnap [14094:60b] dealloc <JoinViewController: 0x14da5280>

```

Testing was carried out when the connection API was lost, based on this function, it will produce state 3 in the debug output panel. In the first sample, when the player does not turn on Bluetooth, the player will get a notification message like in Figure 12.



Fig. 12. No connection notification message.

Whereas in the debug output there will be a GKSession error domain notification message when no communication media (Wi-Fi or Bluetooth) is available, debug output can be seen as follows:

```

2016-03-08 14:04:43.535 BorneoSnap [381:907] BTM: attaching to BTServer
2016-03-08 14:04:48.196 BorneoSnap [381:907] dealloc <HostViewController: 0x1fdc48d0>
2016-03-08 14:04:52.969 BorneoSnap [381:907] Connection Client: session failed Error
Domain=com.apple.gamekit.GKSessionErrorDomain Code=30509 "Network not available." UserInfo=0x1fd87a60
{NSLocalizedFailureReason=Wi-Fi dan Bluetooth and/or Bluetooth is required., NSLocalizedDescription=Network
no available.}
2016-03-08 14:04:54.323 BorneoSnap [381:907] dealloc <MatchmakingClient: 0x1fd75220>
2016-03-08 14:05:52.741 BorneoSnap [381:907] dealloc <JoinViewController: 0x1fddb2c0>
    
```

The second test sample was when the client and server are connected but then the server is disconnected from the application, so there will be a notification message in Figure 13.

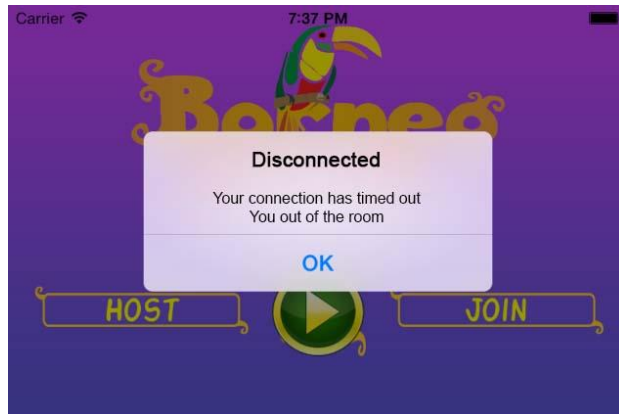


Fig. 13. Connection lost notification message.

In the debug output there will be a notification message indicating the client will change state to state 3 (see debug output below). It is known from Table 1, that state 3 is a condition with a disconnected connection as in Figure 1, then after that, the player will be returned to the initial display of the application.

```

2016-03-08 14:12:44.110 BorneoSnap [439:907] Game: peer 63400799 changed state 3
2016-03-08 14:12:44.745 BorneoSnap [439:907] dealloc <GameViewController: 0x1ede5cf0>
    
```

In the third sample test, the client is in the process of connecting to the server, and suddenly the server disappears from coverage. The test results can be seen in the debug output below, the state client will also change to state 3 (disconnected from the server).

```

2016-03-05 17:02:44.962 BorneoSnap [357:907] dealloc <Player: 0x1f59f6a0> peerID = 1454017797, name =
Muhsan's iPad, position 2
2016-03-08 17:02:45.002 BorneoSnap [357:907] Game: peer 1454017797 changed state 3
    
```

The final test was the process of transferring data between client and server. Transfer data on the Borneo Snap game using the header validation method, if the data received has the same header (in the form of hexadecimal 10 bytes) then it is certain that the data sent is from the Borneo Snap service, but if the data is different (the size is too small or

header does not match) then the data will be ignored, the data transfer process in the debug output window is seen in the following:

```

2016-03-06 17:07:00.157 BorneoSnap [14094:60b] Game: receive data from peer: 2033237104, data: <534e4150
ffffffff 0064>, length: 10
2016-03-06 17:07:00.265 BorneoSnap [14094:60b] Game: receive data from peer: 2033237104, data: <534e4150
ffffffff 00660231 36323635 33313739 30004d75 6873616e e2809973 20695061 64000232 30333332 33373130 34004d75
6873616e 20695068 6f6e6500 00>, length: 65
2016-03-06 17:07:00.332 BorneoSnap [14094:60b] Game: receive data from peer: 2033237104, data: <534e4150
ffffffff 00683230 33333233 37313034 00313632 36353331 37393000 1a000803 0d020203 01020900 0a020402 0c020303
0a030c03 05010c02 0b010900 0d020701 06010801 0d00c030 07030603 09000301 0a323033 33323337 31303400 1a010102
0a000b03 08020503 04020d02 06030202 01030301 03000701 0b000401 05000101 02020800 05000201 04000603 0b000301
07>, length: 149
2016-03-06 17:07:06.753 BorneoSnap [14094:60b] Game: receive data from peer: 2033237104, data: <534e4150
00000000 006a3230 33333233 37313034 00>, length: 21
2016-03-06 17:07:13.718 BorneoSnap [14094:60b] Game: receive data from peer: 2033237104, data: <534e4150
00000000 006a3136 32363533 31373930 00>, length: 21

```

From all the connectivity tests above, it can be concluded that all client-server connection frameworks are running well.

4. Conclusion

Client-server architecture can be used to develop the Borneo Snap PVP game so that it can be played by more than 1 player with different gadgets. Connections and communications between client and server are carried out via the client-server and delegate connection APIs of the GameKit framework. Client-server framework will intelligently divide server and client as P1, P2, P3, and P4. Testing on connectivity also proves that the host and join game can function properly.

The architecture used is expected to be equipped with a shake feature that is useful for finding other players who are around and can send invitations to invite to play Borneo Snap. This game application uses a client-server connection API, it would be better if there is an increase in terms of security and encryption of data sent over the network, to avoid cheating. This game application still uses Wi-Fi media and Bluetooth as media, it is hoped that in the future it will be able to use the internet as a medium of connecting between users on a large scale.

References

- [1] Breuer, J., Pietschmann, D., Liebold, B., & Lange, B. P. (EDS.). 2018. *Evolutionary Psychology and Digital Games: Digital Hunter-gatherers*. Routledge.
- [2] Schell, J. 2014. *The Art of Game Design: A book of lenses*. AK Peters/CRC Press.
- [3] Glazer, J., & Madhav, S. 2015. *Multiplayer game programming: Architecting networked games*. Addison-Wesley Professional.
- [4] Clayton, C. 2016. *iOS 10 Programming for Beginners*. Packt Publishing Ltd.
- [5] Bourg, D. M., & Bywalec, B. 2013. *Physics for Game Developers: Science, math, and code for realistic effects*. " O'Reilly Media, Inc."
- [6] Langley, K. 2015. *Learning Unity iOS Game Development*. Packt Publishing Ltd.
- [7] Wang, M., & Zhu, L. 2014. Designing and implementing an online card game based on Android 2D graphics. In 2014 International Conference on Audio, Language and Image Processing (pp. 817-821). IEEE.
- [8] Rigal, B. 2011. *Card Games for Dummies*. John Wiley & Sons.
- [9] Fachriyani, I., & Syafe'i, A. F. R. 2018. Using Karuta Game in Teaching Vocabulary to young Learners. *Journal of English Language Teaching*, 7(1), 32-39.
- [10] Monaghan, J. 2016. Games: Artefacts in gameplay. In *Tools and Mathematics* (pp. 417-431). Springer, Cham.
- [11] Garraffa, M., Coco, M. I., & Branigan, H. P. 2015. Effects of immediate and cumulative syntactic experience in language impairment: Evidence from priming of subject relatives in children with SLI. *Language Learning and Development*, 11(1), 18-40.
- [12] Hunt, S., Letchford, C., Lloyd, M., Manning, L., & Plummer, R. 2018. The Virtue of Variety: Opening the Doors to Wider Pedagogical Practices in UK Schools and Universities. *Journal of Classics Teaching*, 19(38), 53-60.
- [13] Gonsalves, T. 2019. Board Games AI. In *Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction* (pp. 68-80). IGI Global.
- [14] Andrea, R., & Kopel, M. 2018. Design and development of “battle drone” computer-based trading card game (CTCG). In *International Conference on Multimedia and Network Information System* (pp. 574-585). Springer, Cham.
- [15] Brown, R., & Anderson, M. 2019. Client-Server, Unstable Theory. *Computer Science and Information Technology Journal*, 3(1).
- [16] Pittman, D., & Gauthierdickey, C. 2013. Match+ Guardian: a secure peer-to-peer trading card game protocol. *Multimedia systems*, 19(3), 303-314.
- [17] Yahaya, M. O. 2015. Mitigating Free Riding in Peer-To-Peer Networks: Game Theory Approach. *Nigerian Journal of Technology*, 34(2), 375-384.

Authors' Profiles



Reza Andrea, earned his Bachelor's degree from Computer Science, Mulawarman University in 2012. In 2013 he was involved in a game development project as a leader of Bibir Design Studio and he was awarded the best 2D game developers of Indonesia Game Show (IGS). In 2014, Reza Andrea earned his Master's Degree in Business Engineering from Department of Computer Engineering, STMIK Eresha, Jakarta. Now, he is a lecturer in Software Engineering Department in Polytechnic Agriculture of Samarinda and Expert Staff of Research Department in STMIK Widya Cipta Dharma.



Tommy Bustomi, was born in Jakarta on November 27, 1971. Completed Bachelor studies at Universitas Gunadarma graduated (1997) and Masters studies at the Institut Teknologi Sepuluh Nopember. Computer Engineering Study Program graduated (2006). Lecturer at Politeknik Negeri Samarinda from 2001 until now, and also served as vice-rector for academic at STMIK Widya Cipta Dharma.



Muhammad Muhsan, Was Born In Tenggaraong Seberang September 13, 1992. Completed Formal Education In Elementary School (2004), Junior High School (2007), Senior High School (2010), In Tenggaraong Seberang Kutai Kartanegara East Kalimantan. Proceed To STMIK Widya Cipta Dharma Samarinda. And Is Currently Working In the Hydroponic Agribusiness Sector

How to cite this paper: Reza Andrea, Tommy Bustomi and Muhammad Muhsan, " Client Server iOS in Player versus Player (PVP) of “Borneo Snap”", International Journal of Information Engineering and Electronic Business(IJIEEB), Vol.13, No.1, pp. 25-35, 2021. DOI: 10.5815/ijieeb.2021.01.03