Corrections for all Printings

Pg	Error	Correction
32	In points 5, 6 and 8, the IP address 18.19.200.1 is incorrect	It should be 18.19.200.2
47	fig 2.15: SND.NXT of Host B is incorrect	The value should be 3051, not 3001
59	Some IP addresses do not match those in the figure	Paragraph should read: As the packet travels from Host A to Host B, it passes through Router A. Router A sees in the NAT table that Host A's address, 192.168.10.2:200, already maps to external port 60000, so it chooses this port for the outgoing packet. It then makes an additional entry stating that 192.168.10.2:200 has sent traffic to 12.12.6.5:62000. This additional entry is the key. The packet will probably never arrive at Host B, but after this has happened, Host N can reply to Host B, telling it to connect directly to Host A at 18.19.20.21:60000. Host B does so, and when the packet arrives at Router A, Router A sees that it is indeed expecting an incoming packet from 12.12.6.5:62000. It rewrites the packet to be targeting 192.168.10.2:200 and sends it to Host A.
69	WinSocket2.h is not a header name	WinSocket2.h should be WinSock2.h
73	sockadd_in is not a struct name	sockadd_in should be sockaddr_in
73	In the note, sa_len is no longer a field in the Mac OS X sockaddr	sa_len in the note should be sin_len

77	Listing 3.4 should not call freeaddrinfo on an interated result	**Listing 3.4** Name Resolution Using the `SocketAddressFactory`
		`class SocketAddressFactory`
		`{`
		`public:`
		`    static SocketAddressPtr CreateIPv4FromString(const string& inString)`
		`    {`

```
        auto pos = inString.find_last_of(':');
        string host, service;
        if(pos != string::npos)
        {
            host = inString.substr(0, pos);
            service = inString.substr(pos + 1);
        }
        else
        {
            host = inString;
            //use default port...
            service = "0";
        }
        addrinfo hint;
        memset(&hint, 0, sizeof(hint));
        hint.ai_family = AF_INET;

        addrinfo* result;
        int error = getaddrinfo(host.c_str(), service.c_str(),
                                 &hint, &result);
        addrinfo* initResult = result;
        if(error != 0 && result != nullptr)
        {
            freeaddrinfo(initResult);
            return nullptr;
        }

        while(!result->ai_addr && result->ai_next)
        {
            result = result->ai_next;
        }

        if(!result->ai_addr)
        {
            freeaddrinfo(initResult);
            return nullptr;
        }

        auto toRet = std::make_shared< SocketAddress >(*result->ai_addr);

        freeaddrinfo(initResult);

        return toRet;
    }
};
```

80	**The 5th paragraph actually describes the parameter named tolen but claims to describe the parameter named len**	**len at the beginning of that 5th paragraph should be changed to tolen**
83	**WASGetLastError is not a function**	**WASGetLastError should be WSAGetLastError**

94	**In listing 3.10, the lines**
	fd_set *writePtr = FillSetFromVector(read, inWriteSet);
	fd_set *exceptPtr = FillSetFromVector(read, inExceptSet);
	are incorrect
		They should read
		fd_set *writePtr = FillSetFromVector(write, inWriteSet);
		fd_set *exceptPtr = FillSetFromVector(except, inExceptSet);

95	**In listing 3.11 there is a missing call to Listen after the socket is bound**	**Right before**
		`vector< TCPSocketPtr > readBlockSockets;`
		should be inserted:
		`if( listenSocket->Listen() != NO_ERROR )`
		`{`
		`    return;`
		`}`

102	**sentdo is not a function**	**sentdo should be sendto**
121	**In the definition of Read(std::vector<T>&), the inner call to Read does not take a const argument, so the iterated variable should not be const T& element**	**const T& element should instead be T& element with no const in front**
126	**The equal signs were printed as plus signs**	**The equation should read:** $P_{OnGround} * Bits_{OnGround} + P_{InAir} * Bits_{InAir} = 0.9 * 1 + 0.1 * 33 = 4.2$
127	**The equal signs were printed as plus signs and the line breaks were omitted**	**The equation should read:**

		$P_{OnGround} * Bits_{OnGround} + P_{InAir} * Bits_{InAir} + P_{OnCeiling} * Bits_{OnCeiling} =$ $0.9*2+0.07*2+0.03*33=$ 2.93
128	**The equal signs were printed as plus signs and the line breaks were omitted**	$(MaxValue - MinValue) / Precision + 1 =$ $(2000 - -2000) / 0.1 + 1 =$ 40001

135	**The Serialize code snippet is missing the case keyword from each case in the switch statement**	void Serialize(MemoryStream* inMemoryStream,
		            const DataType* inDataType, uint8_t* inData)
		{
		    for(auto& mv: inDataType->GetMemberVariables())
		    {
		        void* mvData = inData + mv.GetOffset();
		        switch(mv.GetPrimitiveType())
		        {
		            case EPT_Int:
		               inMemoryStream->Serialize(*(int*) mvData);
		               break;
		            case EPT_String:
		               inMemoryStream->Serialize(*(std::string*) mvData);
		               break;
		            case EPT_Float:
		               inMemoryStream->Serialize(*(float*) mvData);
		               break;
		        }
		    }
		}

141	**In listing 5.2 the class definition does not end with a semicolon**	**There should be a semicolon after the closing brace after the class definition**
151	**In listing 5.9 the code to detect if a game object is missing from the receivedObjects list does not work correctly**	**find(go) != receivedObjects.end() should be changed to find(go) == receivedObjects.end()**
160	**The text incorrectly refers to the**	**PlayerSoundRPCParams should be**

	PlaySoundRPCParams struct as PlayerSoundRPCParams	**PlaySoundRPCParams**
160	**In listing 5.7, the typedef statement is missing a semicolon**	**The first line of the listing should end with a semicolon.**
162	**The text incorrectly refers to the nonexistent ProcessReplicationFunction.**	**ProcessReplicationFunction should be ProcessReplcationAction function.**
221	**HandleDeliverySucess is not a function**	**HandleDeliverySucess should be HandleDeliverySuccess**
225	**ReplicationTransmissionData is not a type**	**ReplicationTransmissionData should be ReplicationManagerTransmissionData**

This errata sheet is intended to provide updated technical information. Spelling and grammar misprints are updated during the reprint process, but are not listed on this errata sheet.