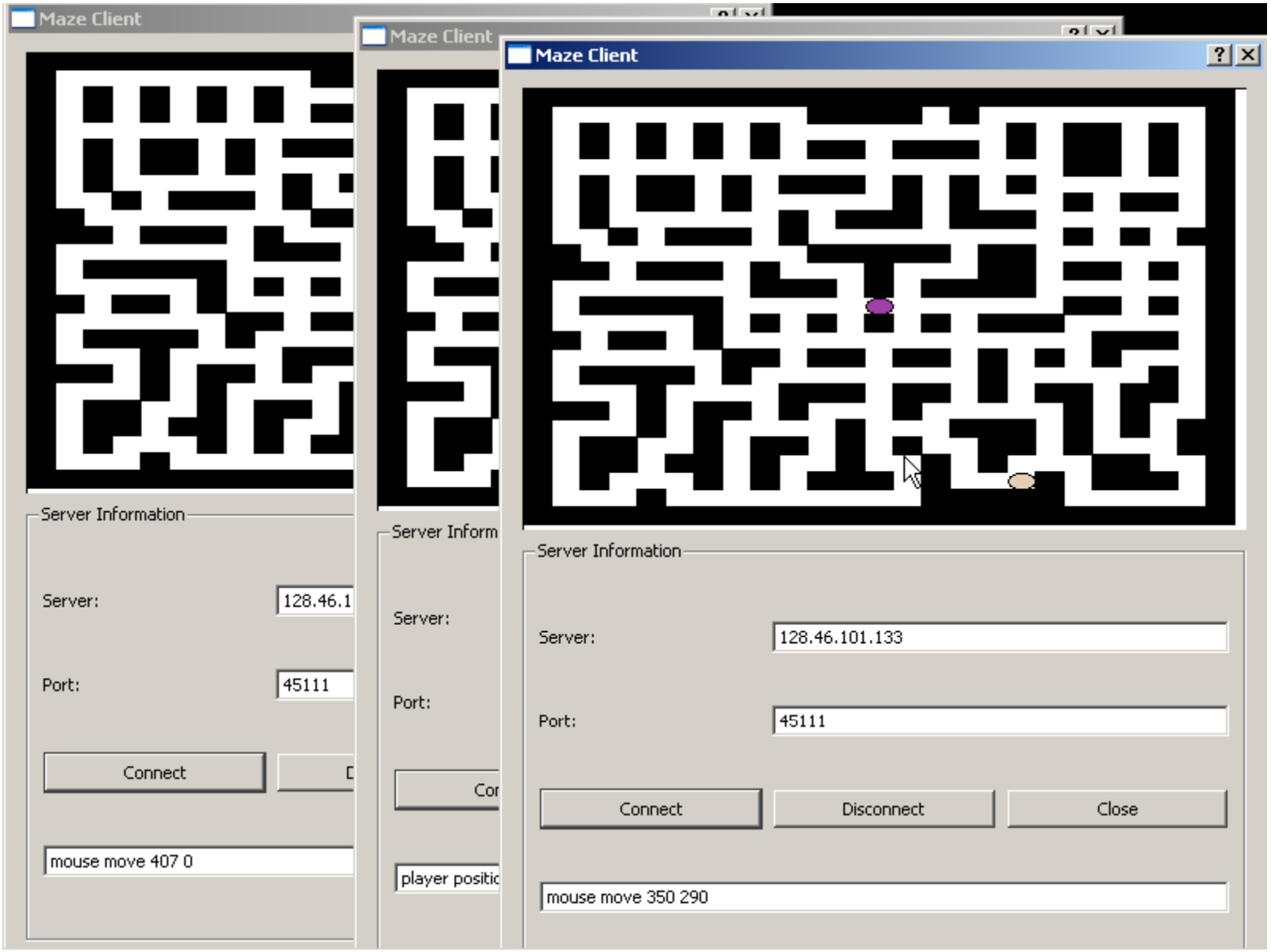
# ECE 462
# Object-Oriented Programming using C++ and Java

# Multi-Player Game using Network

Yung-Hsiang Lu
yunglu@purdue.edu

# Demonstration


Maze Client
Server Information
Server:
128.46.1
Port:
45111
Connect
mouse move 407 0
Maze Client
Server Inform
Server:
Port:
Cor
player positic
Maze Client
Server Information
Server:
128.46.101.133
Port:
45111
Connect
Disconnect
Close
mouse move 350 290

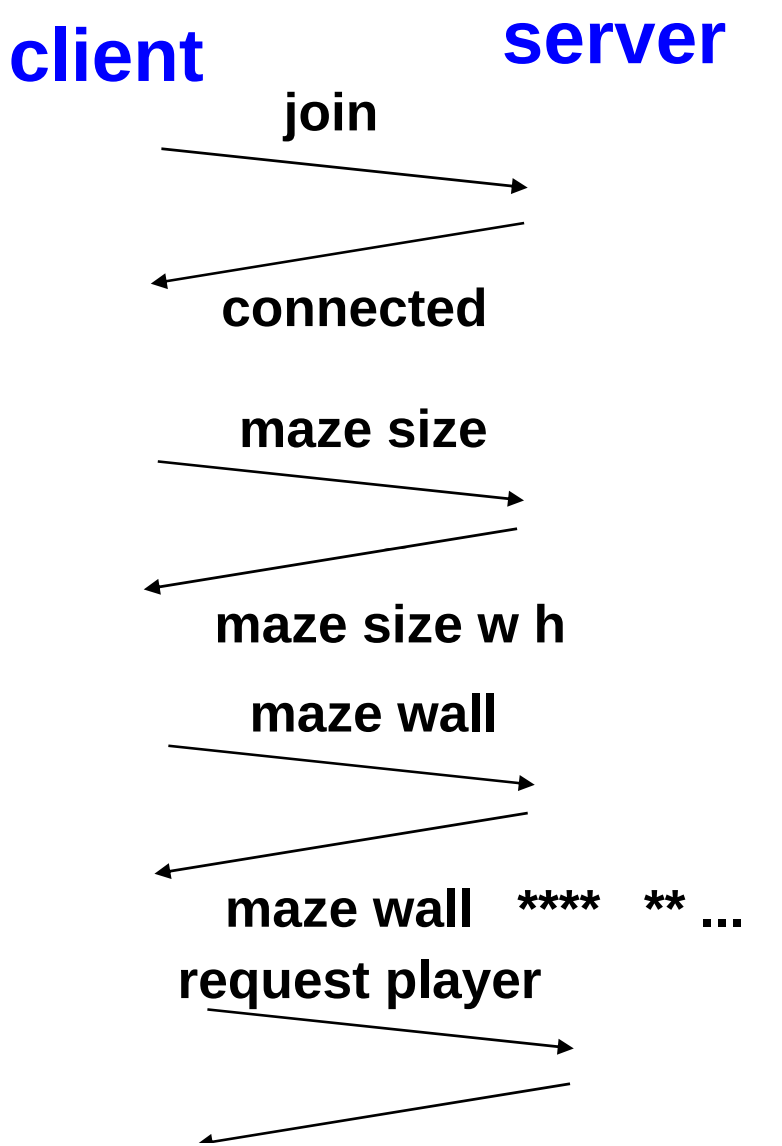
# Design Network Game

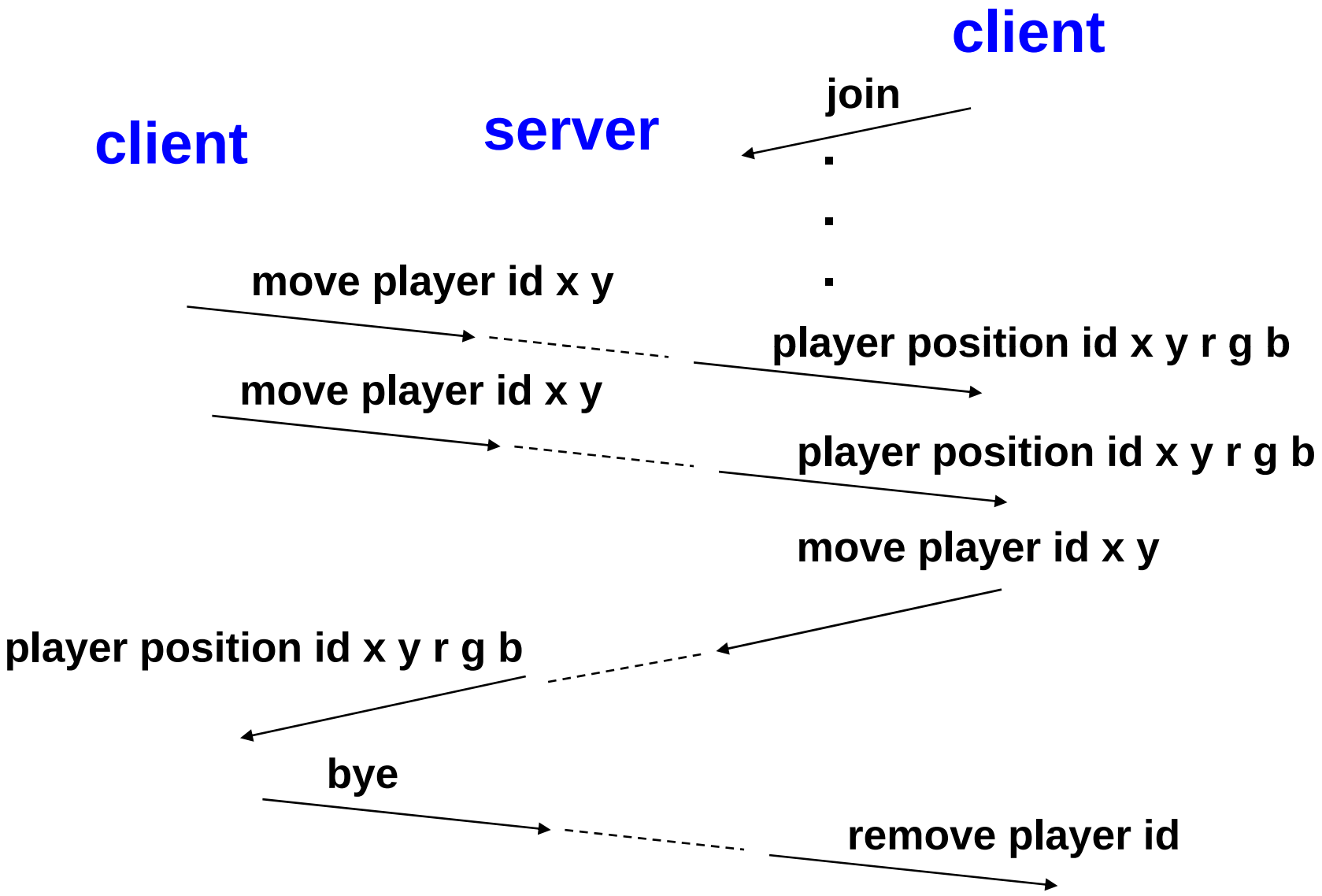
- user interface
- server lookup and connection
- game protocol
- data consistency
- limit of numbers of clients
- exit and error handling

# Game Protocol

- who can talk to whom? what to say? when?
- data types
- data format
- encryption
- push or pull or both?

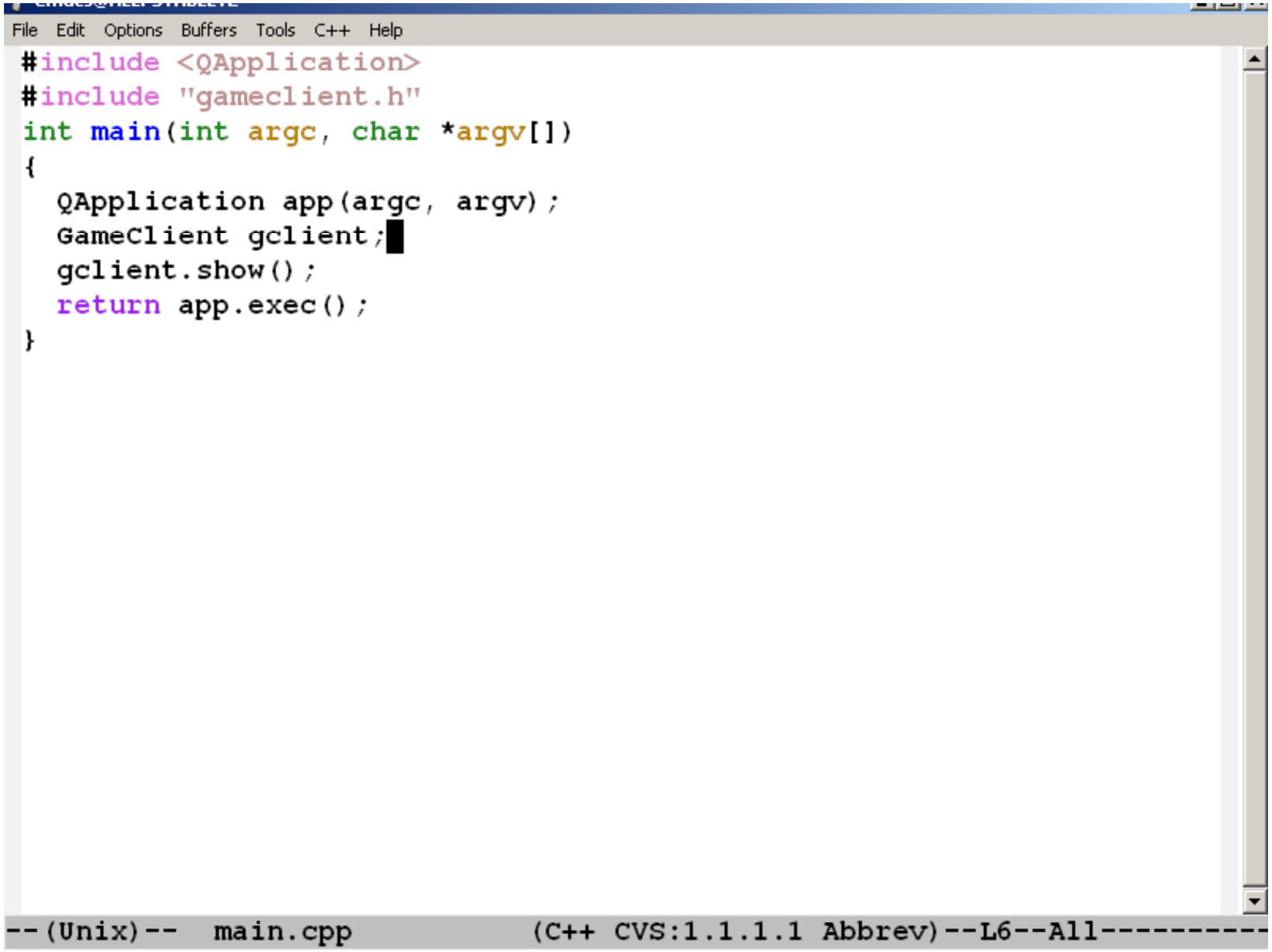
# Communication Sequence

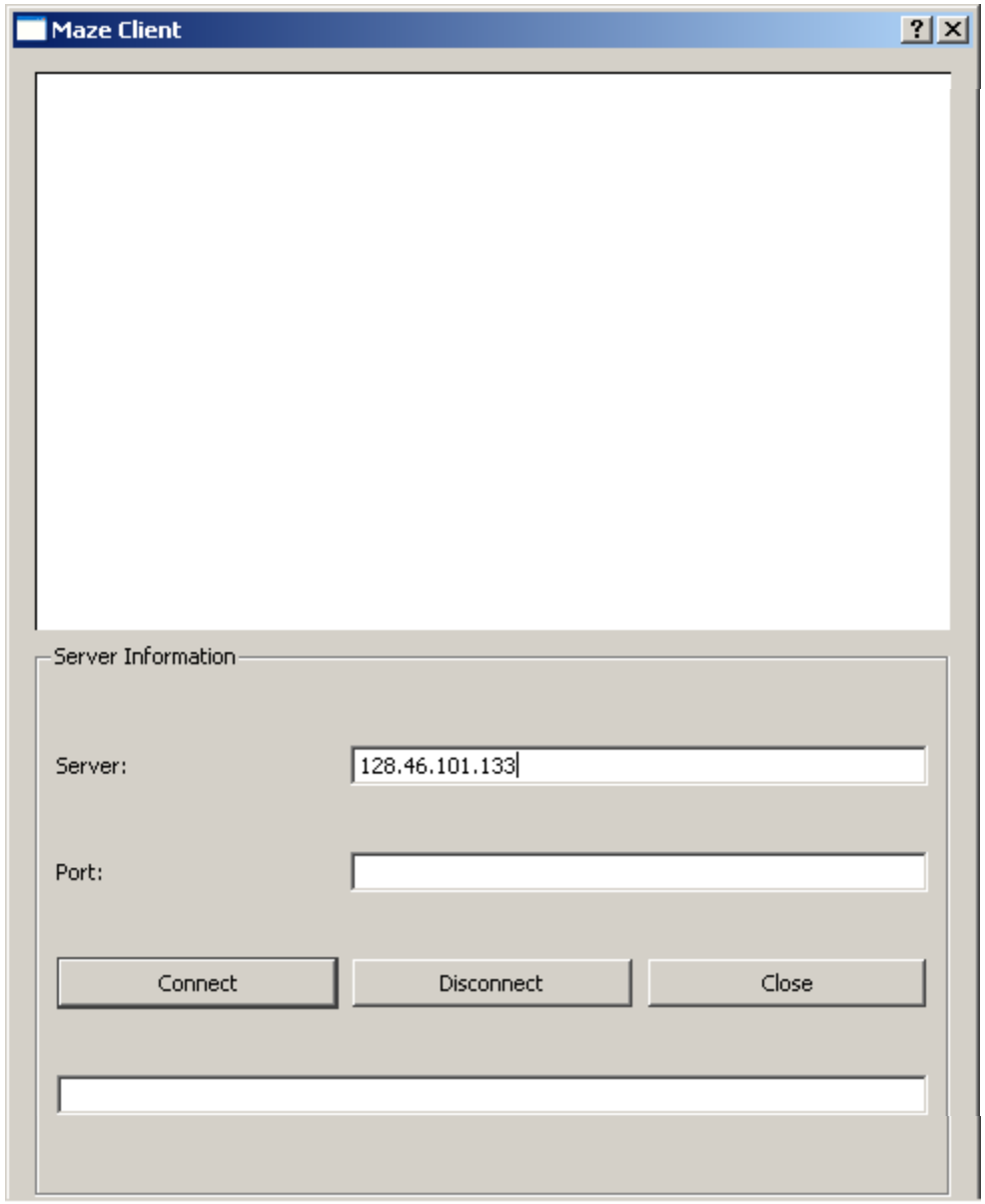

client
server
join
connected
maze size
maze size w h
maze wall
maze wall **** ** ...
request player
player ID location color
id x y r g b

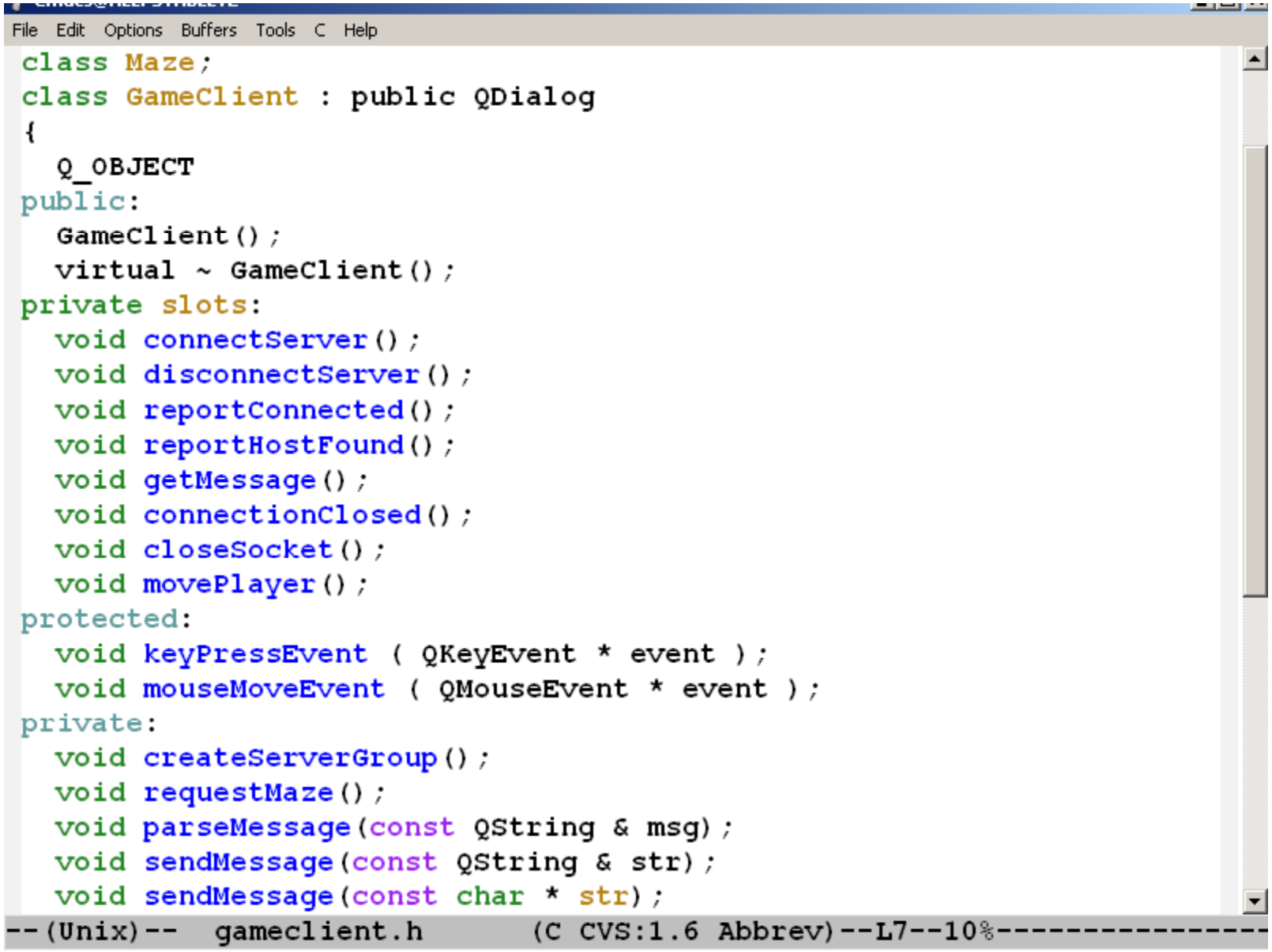

client
client
server
join
move player id x y
player position id x y r g b
move player id x y
player position id x y r g b
move player id x y
player position id x y r g b
bye
remove player id

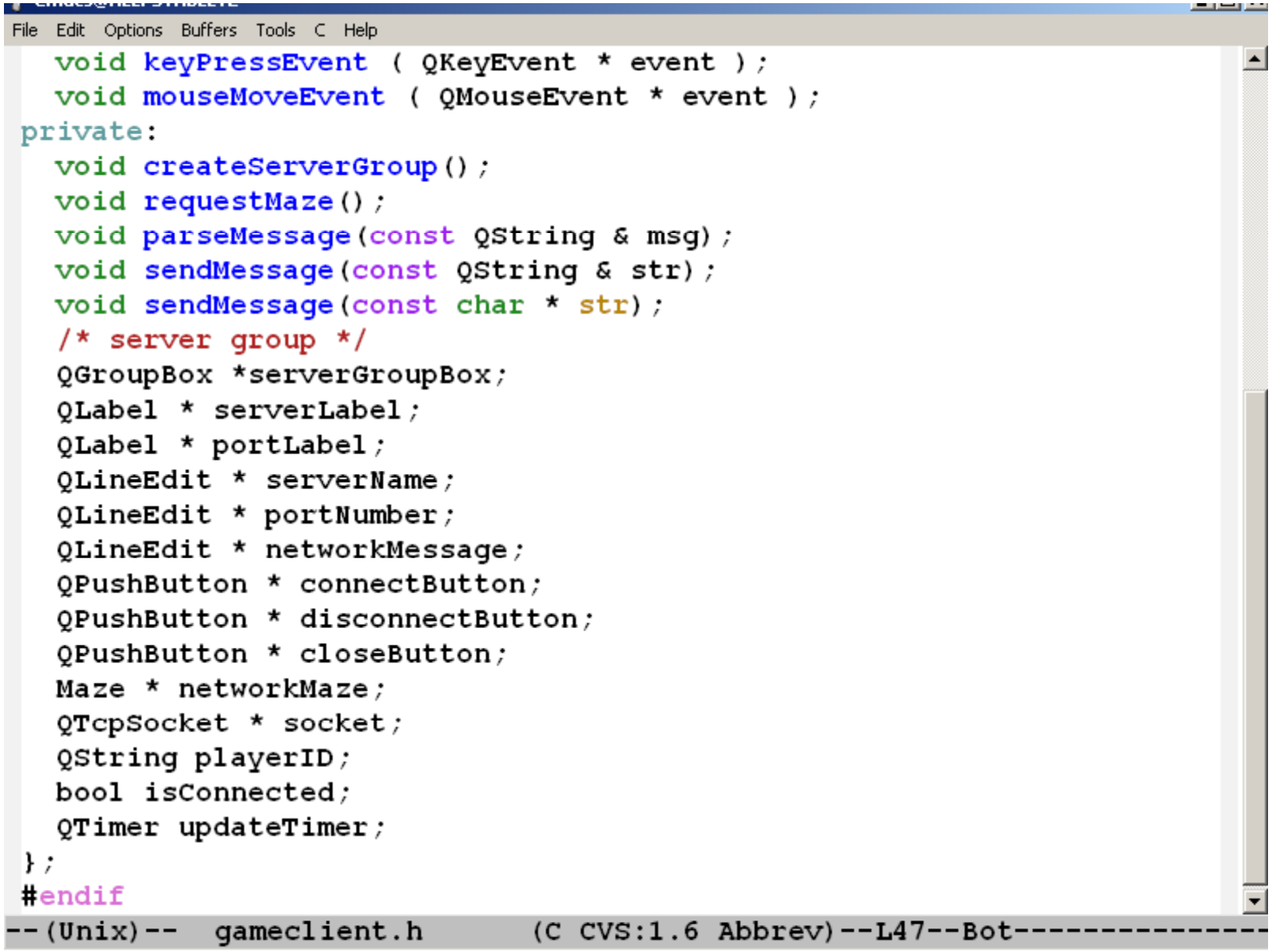
# C++ Qt Implementation

# Client


File Edit Options Buffers Tools C++ Help
#include <QApplication>
#include "gameclient.h"
int main(int argc, char *argv[])
{
QApplication app(argc, argv);
GameClient gclient;
gclient.show();
return app.exec();
}
--(Unix)-- main.cpp (C++ CVS:1.1.1.1 Abbrev)--L6--All----------


Maze Client
Server Information
Server:
128.46.101.133
Port:
Connect
Disconnect
Close


File Edit Options Buffers Tools C Help
class Maze;
class GameClient : public QDialog
{
Q_OBJECT
public:
GameClient();
virtual ~ GameClient();
private slots:
void connectServer();
void disconnectServer();
void reportConnected();
void reportHostFound();
void getMessage();
void connectionClosed();
void closeSocket();
void movePlayer();
protected:
void keyPressEvent ( QKeyEvent * event );
void mouseMoveEvent ( QMouseEvent * event );
private:
void createServerGroup();
void requestMaze();
void parseMessage(const QString & msg);
void sendMessage(const QString & str);
void sendMessage(const char * str);
--(Unix)-- gameclient.h (C CVS:1.6 Abbrev)--L7--10%----------


File Edit Options Buffers Tools C Help
void keyPressEvent ( QKeyEvent * event );
void mouseMoveEvent ( QMouseEvent * event );
private:
void createServerGroup();
void requestMaze();
void parseMessage(const QString & msg);
void sendMessage(const QString & str);
void sendMessage(const char * str);
/* server group */
QGroupBox *serverGroupBox;
QLabel * serverLabel;
QLabel * portLabel;
QLineEdit * serverName;
QLineEdit * portNumber;
QLineEdit * networkMessage;
QPushButton * connectButton;
QPushButton * disconnectButton;
QPushButton * closeButton;
Maze * networkMaze;
QTcpSocket * socket;
QString playerID;
bool isConnected;
QTimer updateTimer;
};
#endif
--(Unix)-- gameclient.h (C CVS:1.6 Abbrev)--L47--Bot

File Edit Options Buffers Tools C++ Help

```
#include <QtGui>
#include <stdlib.h>
#include <string.h>
#include <iostream>
#include <sys/time.h>
#include "gameclient.h"
#include "maze.h"
using namespace std;
GameClient::GameClient()
{
  setWindowTitle(tr("Maze Client"));
  isConnected = false;
  socket = NULL;
  createServerGroup();
  QVBoxLayout *mainLayout = new QVBoxLayout;
  networkMaze = new Maze;
  mainLayout->addWidget(networkMaze);
  mainLayout->addWidget(serverGroupBox);
  setLayout(mainLayout);
  resize(480, 600);
  setMouseTracking(true);
  updateTimer.setInterval(300); // milisecond
  updateTimer.setSingleShot(false); // repeat the timer
  connect(& updateTimer, SIGNAL(timeout()), this,
          SLOT(movePlayer()));
```

--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L10--Top----------

File Edit Options Buffers Tools C++ Help

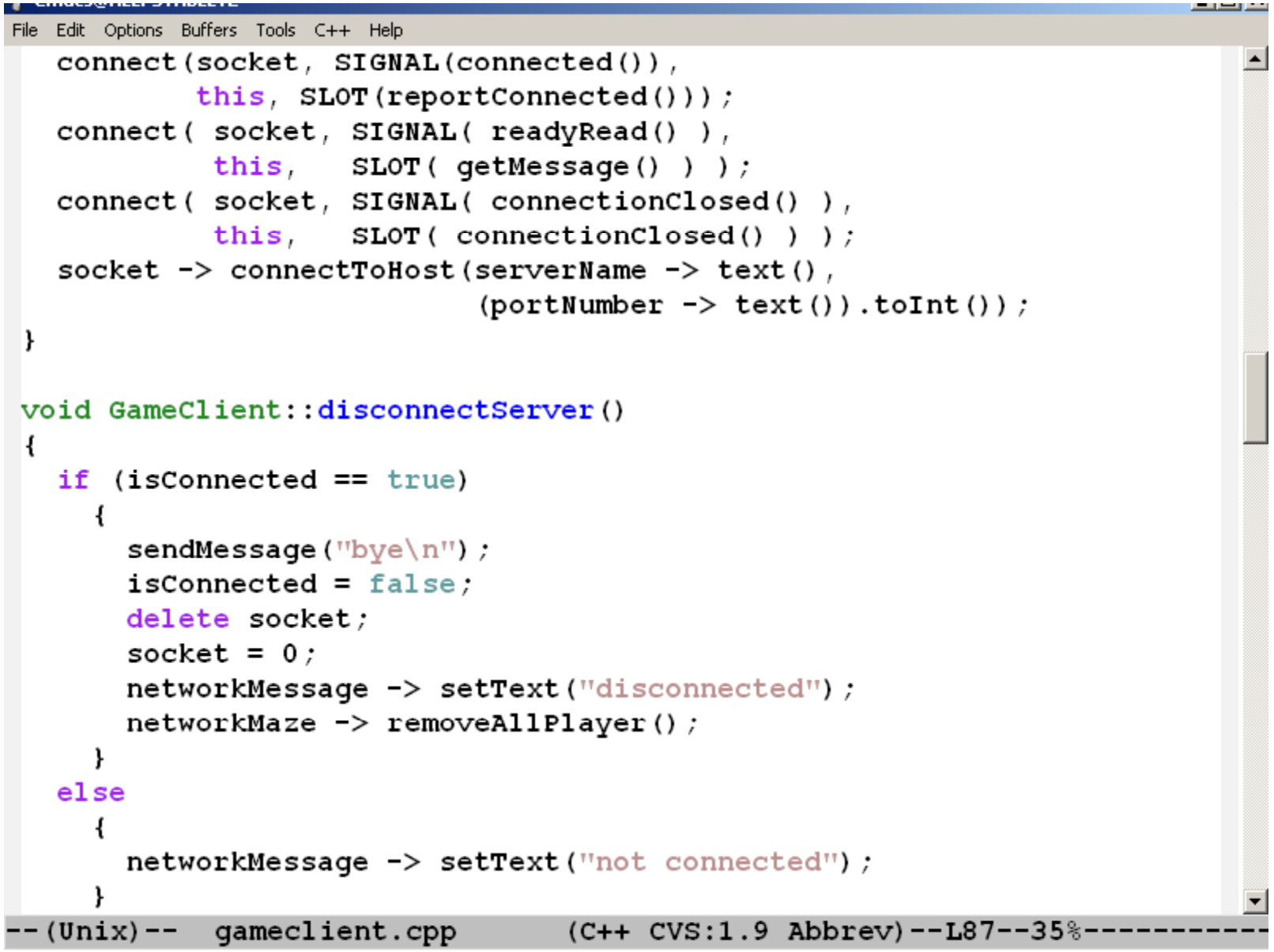
```
GameClient:: ~GameClient()
{
  delete serverGroupBox;
  delete networkMaze;
}

void GameClient::createServerGroup()
{
  serverGroupBox = new QGroupBox(tr("Server Information"));
  QGridLayout *layout = new QGridLayout;

  serverLabel = new QLabel("Server: ");
  serverName = new QLineEdit("128.46.101.133");
  layout -> addWidget(serverLabel, 0, 0);
  layout -> addWidget(serverName, 0, 1, 1, 2);

  portLabel = new QLabel("Port: ");
  portNumber = new QLineEdit();
  layout -> addWidget(portLabel, 1, 0);
  layout -> addWidget(portNumber, 1, 1, 1, 2);

  connectButton = new QPushButton("Connect");
  connect(connectButton, SIGNAL(clicked()),
          this, SLOT(connectServer()));
  layout -> addWidget(connectButton, 2, 0);
```

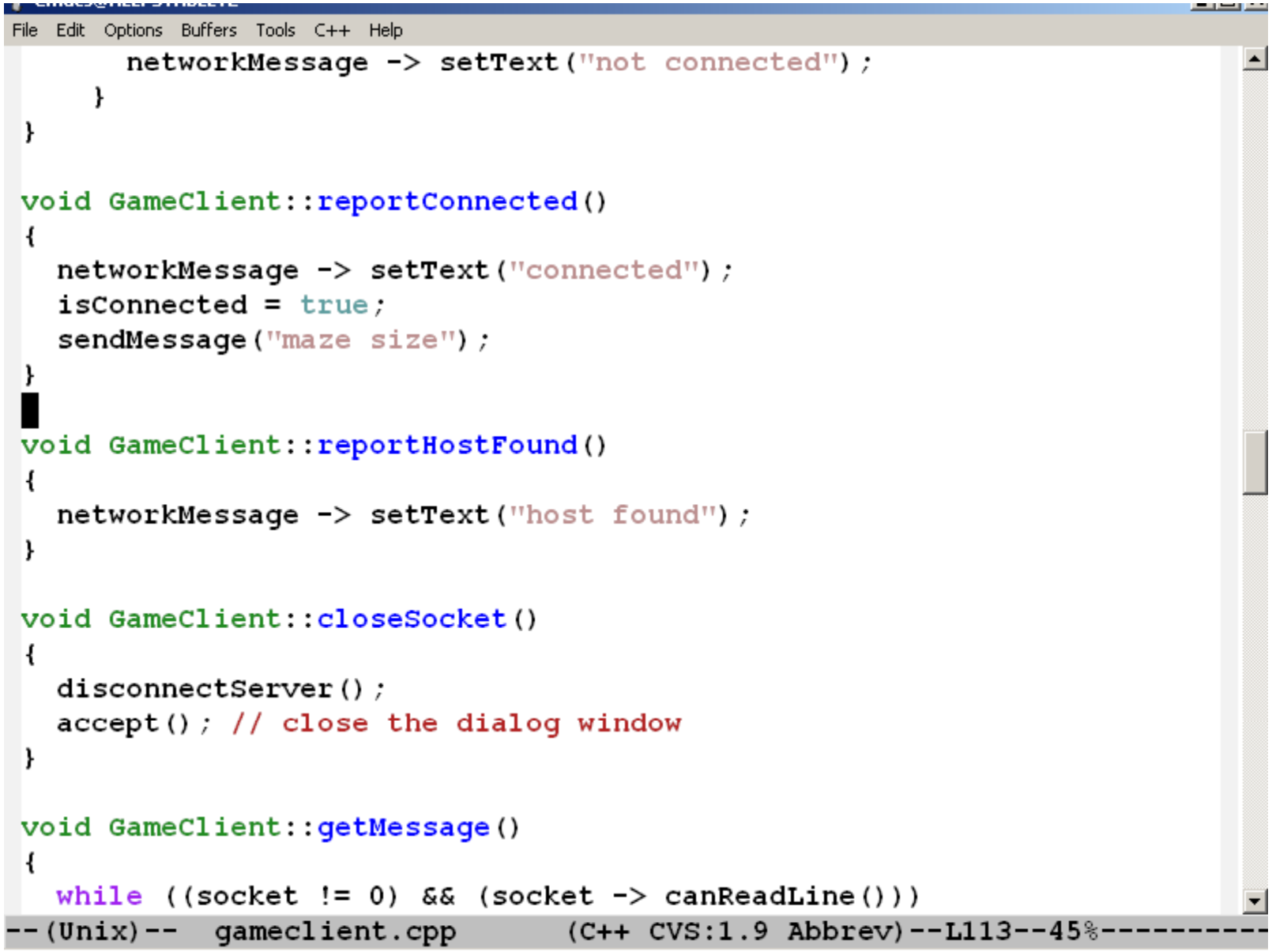
--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L33--11%----------

File Edit Options Buffers Tools C++ Help

```
  disconnectButton = new QPushButton("Disconnect");
  connect(disconnectButton, SIGNAL(clicked()),
          this, SLOT(disconnectServer()));
  layout -> addWidget(disconnectButton, 2, 1);

  closeButton = new QPushButton("Close");
  connect(closeButton, SIGNAL(clicked()),
          this, SLOT(closeSocket()));
  layout -> addWidget(closeButton, 2, 2);

  networkMessage = new QLineEdit;
  layout -> addWidget(networkMessage, 3, 0, 1, 3);

  serverGroupBox -> setLayout(layout);
}

void GameClient::connectServer()
{
  QString hostport(serverName -> text());
  hostport.append(":");
  hostport.append(portNumber -> text());
  QString textContent(hostport);
  networkMessage -> setText(hostport);
  socket = new QTcpSocket();
  connect( socket, SIGNAL( hostFound() ),
```

--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L69--23%----------

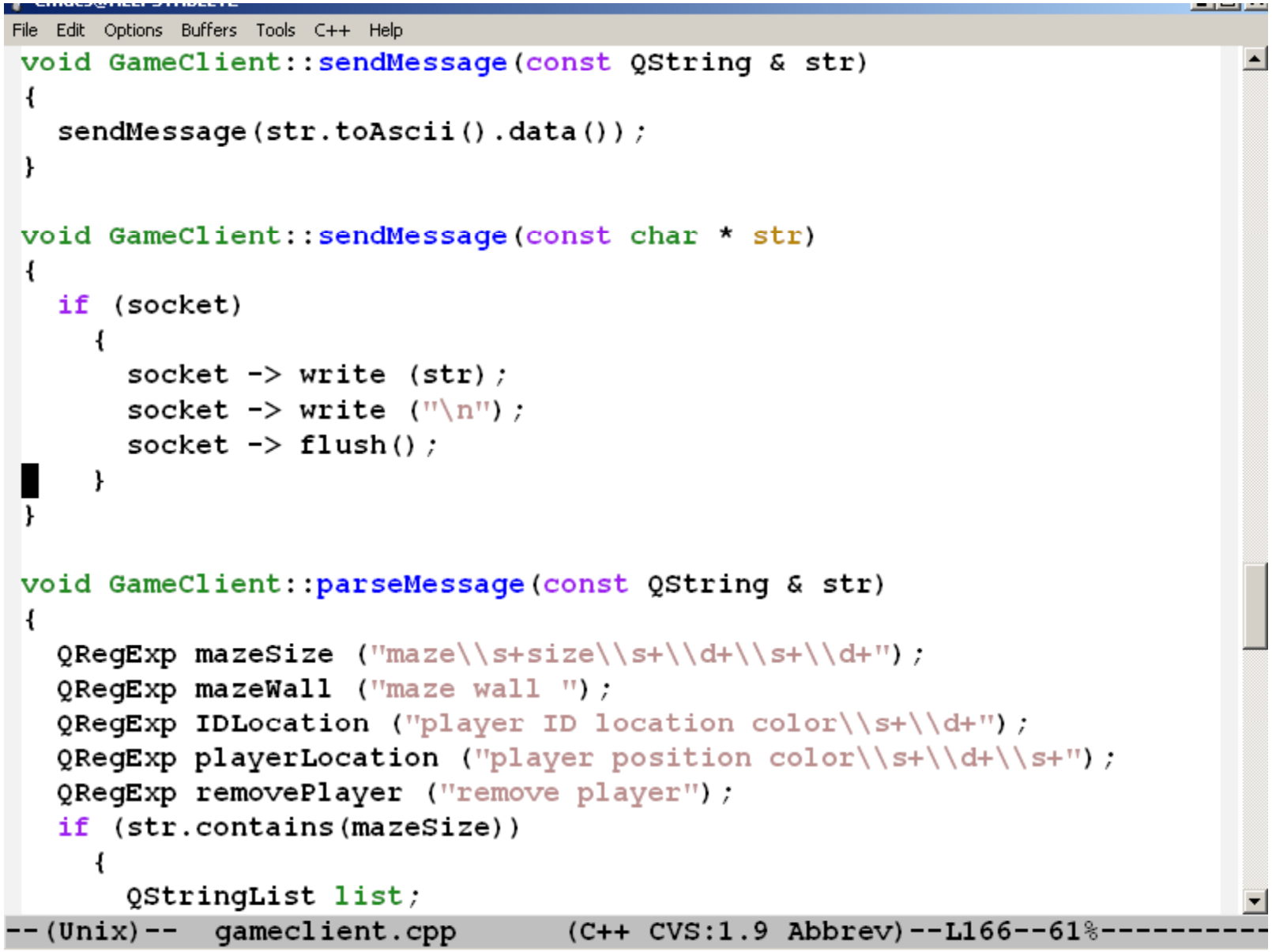

File Edit Options Buffers Tools C++ Help
connect(socket, SIGNAL(connected()),
this, SLOT(reportConnected()));
connect( socket, SIGNAL( readyRead() ),
this, SLOT( getMessage() ) );
connect( socket, SIGNAL( connectionClosed() ),
this, SLOT( connectionClosed() ) );
socket -> connectToHost(serverName -> text(),
(portNumber -> text()).toInt());
}
void GameClient::disconnectServer()
{
if (isConnected == true)
{
sendMessage("bye\n");
isConnected = false;
delete socket;
socket = 0;
networkMessage -> setText("disconnected");
networkMaze -> removeAllPlayer();
}
else
{
networkMessage -> setText("not connected");
}
--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L87--35%----------

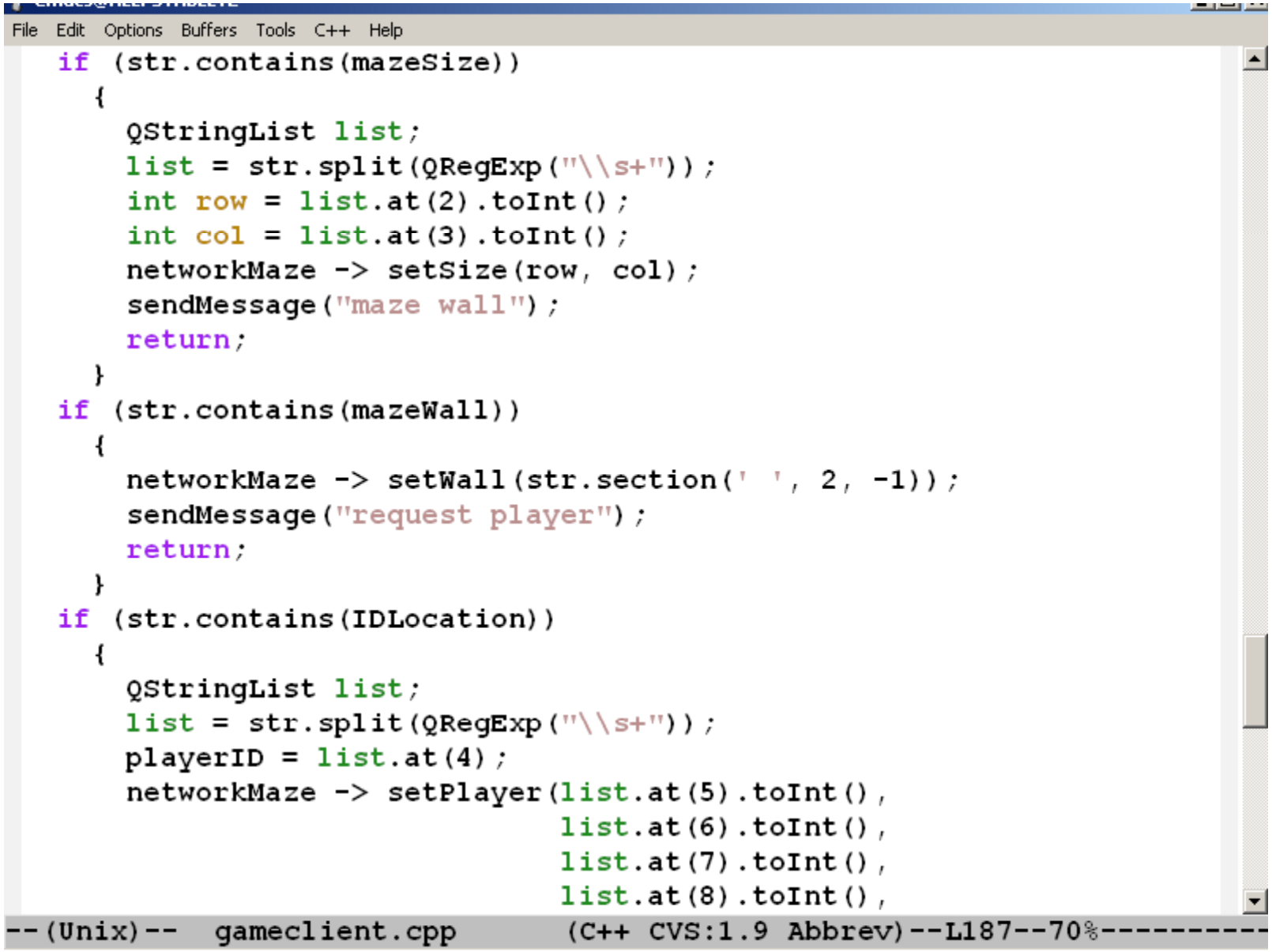

File Edit Options Buffers Tools C++ Help
networkMessage -> setText("not connected");
}
}
void GameClient::reportConnected()
{
networkMessage -> setText("connected");
isConnected = true;
sendMessage("maze size");
}
void GameClient::reportHostFound()
{
networkMessage -> setText("host found");
}
void GameClient::closeSocket()
{
disconnectServer();
accept(); // close the dialog window
}
void GameClient::getMessage()
{
while ((socket != 0) && (socket -> canReadLine()))
--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L113--45%----------

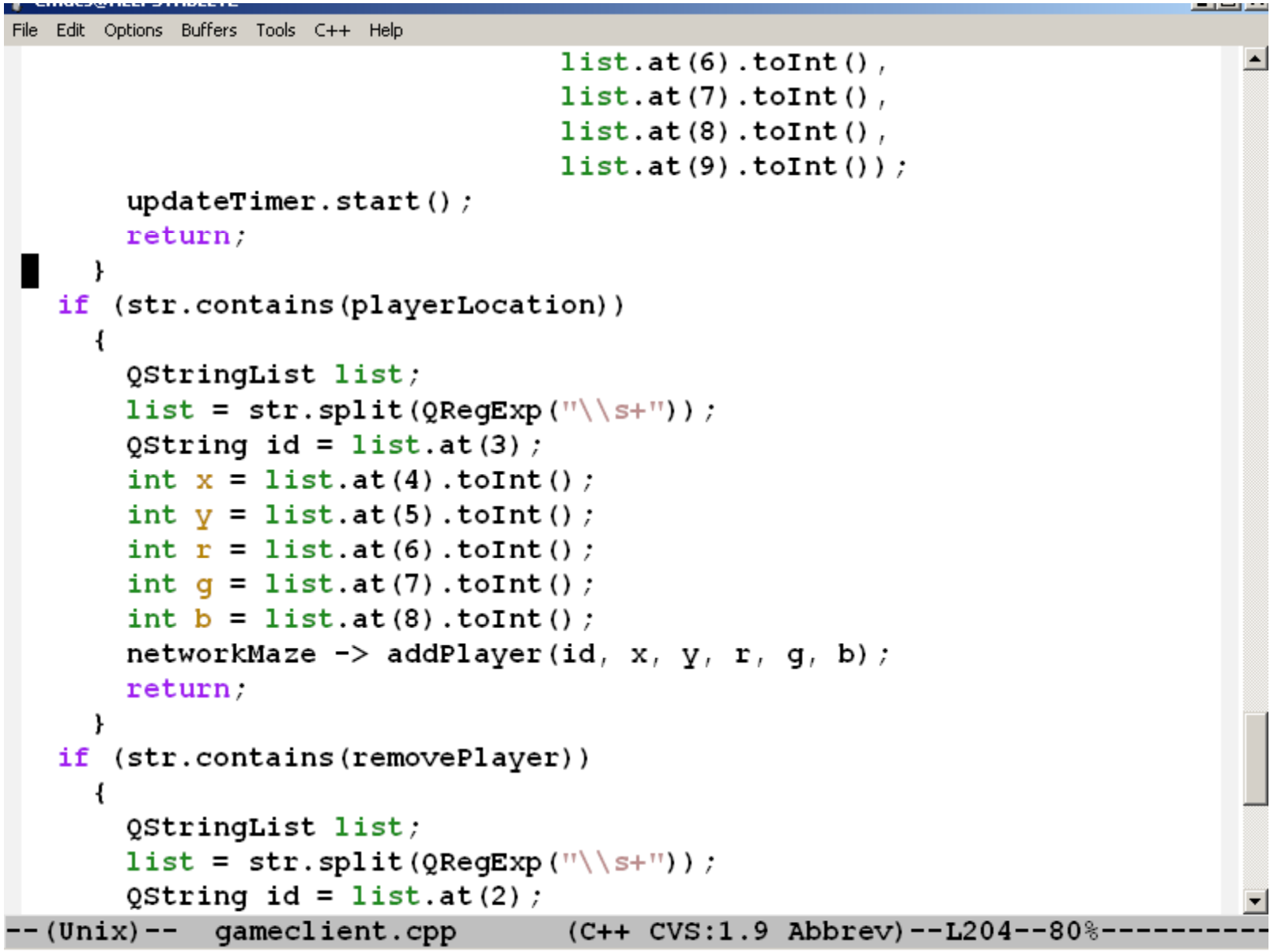
File Edit Options Buffers Tools C++ Help

```
void GameClient::getMessage()
{
  while ((socket != 0) && (socket -> canReadLine()))
    {
      QString qstr = socket -> readLine();
      cout << "=received= " << qstr.toAscii().data();
      networkMessage -> setText(qstr);
      parseMessage(qstr);
    }
}

void GameClient::connectionClosed()
{
  networkMessage -> setText("connection closed\n");
}

void GameClient::mouseMoveEvent ( QMouseEvent * event )
{
  QString mouseText = "mouse move " +
    QString::number(event -> x()) + " " +
    QString::number(event -> y());
  networkMessage -> setText(mouseText);
}

void GameClient::keyPressEvent ( QKeyEvent * event )
```

--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L135--51%----------


File Edit Options Buffers Tools C++ Help
void GameClient::sendMessage(const QString & str)
{
sendMessage(str.toAscii().data());
}
void GameClient::sendMessage(const char * str)
{
if (socket)
{
socket -> write (str);
socket -> write ("\n");
socket -> flush();
}
}
void GameClient::parseMessage(const QString & str)
{
QRegExp mazeSize ("maze\\s+size\\s+\\d+\\s+\\d+");
QRegExp mazeWall ("maze wall ");
QRegExp IDLocation ("player ID location color\\s+\\d+");
QRegExp playerLocation ("player position color\\s+\\d+\\s+");
QRegExp removePlayer ("remove player");
if (str.contains(mazeSize))
{
QStringList list;
--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L166--61%----------


File Edit Options Buffers Tools C++ Help
if (str.contains(mazeSize))
{
QStringList list;
list = str.split(QRegExp("\\s+"));
int row = list.at(2).toInt();
int col = list.at(3).toInt();
networkMaze -> setSize(row, col);
sendMessage("maze wall");
return;
}
if (str.contains(mazeWall))
{
networkMaze -> setWall(str.section(' ', 2, -1));
sendMessage("request player");
return;
}
if (str.contains(IDLocation))
{
QStringList list;
list = str.split(QRegExp("\\s+"));
playerID = list.at(4);
networkMaze -> setPlayer(list.at(5).toInt(),
list.at(6).toInt(),
list.at(7).toInt(),
list.at(8).toInt(),
--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L187--70%----------


File Edit Options Buffers Tools C++ Help
list.at(6).toInt(),
list.at(7).toInt(),
list.at(8).toInt(),
list.at(9).toInt());
updateTimer.start();
return;
}
if (str.contains(playerLocation))
{
QStringList list;
list = str.split(QRegExp("\\s+"));
QString id = list.at(3);
int x = list.at(4).toInt();
int y = list.at(5).toInt();
int r = list.at(6).toInt();
int g = list.at(7).toInt();
int b = list.at(8).toInt();
networkMaze -> addPlayer(id, x, y, r, g, b);
return;
}
if (str.contains(removePlayer))
{
QStringList list;
list = str.split(QRegExp("\\s+"));
QString id = list.at(2);
--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L204--80%----------

File Edit Options Buffers Tools C++ Help

```
  if (str.contains(removePlayer))
    {
      QStringList list;
      list = str.split(QRegExp("\\s+"));
      QString id = list.at(2);
      networkMaze -> removePlayer(id);
      return;
    }
}

void GameClient::movePlayer()
{
  // subsitute handling keyboard or mouse movement so
  // that the players autonomously move
  int x, y;
  if ((networkMaze -> tryMove(rand() % 5, x, y)) == true)
    {
      // actually moved, inform the server
      // cout << "move to " << x << " " << y << endl;
      QString msg("move player ");
      msg.append(playerID);
      msg.append(" ");
      msg.append(QString::number(x));
      msg.append(" ");
      msg.append(QString::number(y));
```

--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L231--88%----------

File Edit Options Buffers Tools C++ Help

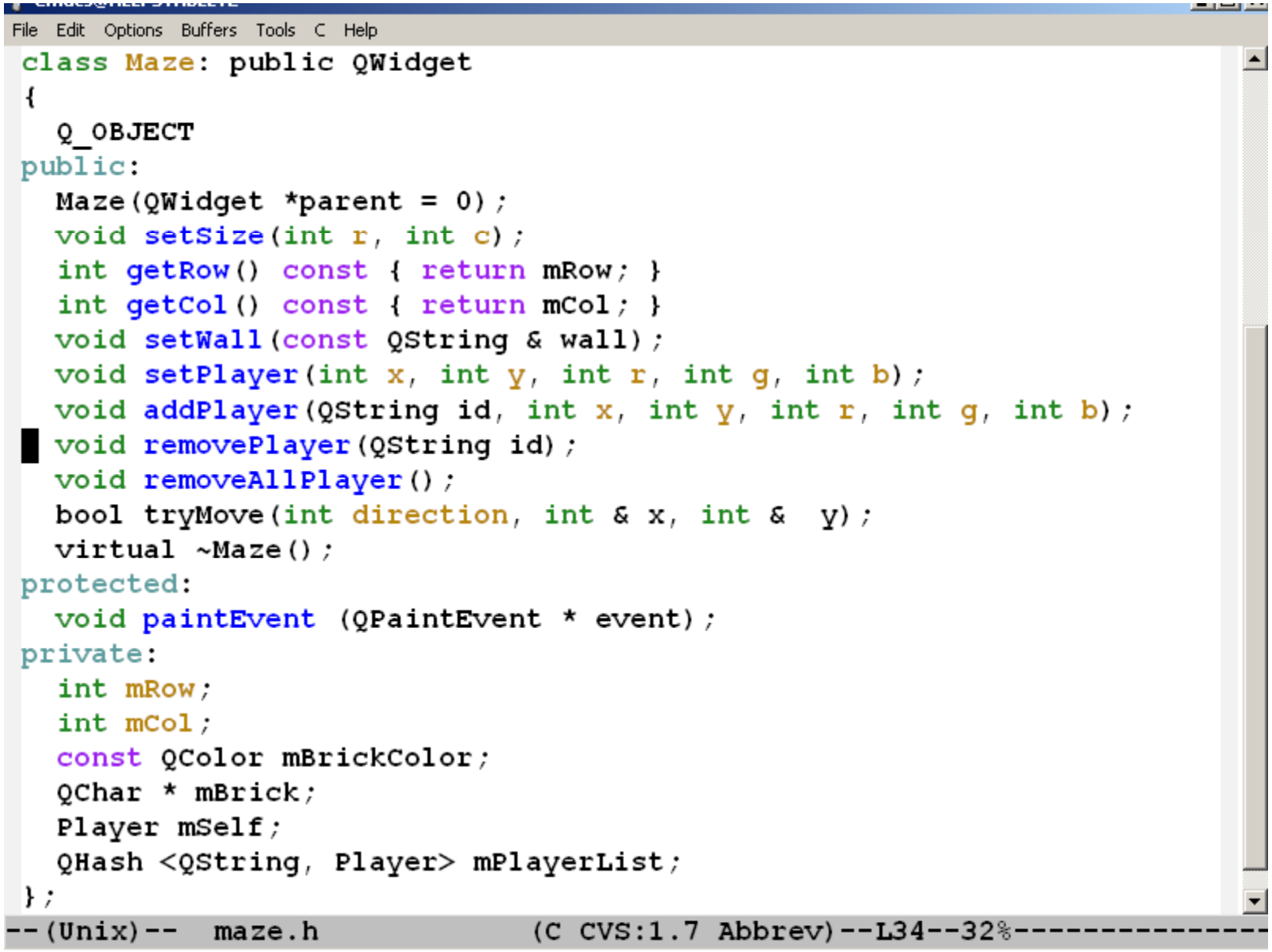
```
}

void GameClient::movePlayer()
{
  // subsitute handling keyboard or mouse movement so
  // that the players autonomously move
  int x, y;
  if ((networkMaze -> tryMove(rand() % 5, x, y)) == true)
    {
      // actually moved, inform the server
      // cout << "move to " << x << " " << y << endl;
      QString msg("move player ");
      msg.append(playerID);
      msg.append(" ");
      msg.append(QString::number(x));
      msg.append(" ");
      msg.append(QString::number(y));
      sendMessage(msg);
    }
  else
    {
      // attempt fails, nothing to do
    }
}
```

--(Unix)-- gameclient.cpp (C++ CVS:1.9 Abbrev)--L231--Bot----------

File Edit Options Buffers Tools C Help

```
#ifndef MAZE_H_
#define MAZE_H_
#include <QtGui>
class Player
{
public:
 Player(int x = 0, int y = 0,
        int r = 0, int g = 0, int b = 0)
   {
     pX = x;
     pY = y;
     pColor.setRgb(r, g, b);
     pDirection = 0;
   }
private:
 int pX;
 int pY;
 int pDirection;
 QColor pColor;
 friend class Maze;
};

class Maze: public QWidget
{
  Q_OBJECT
```

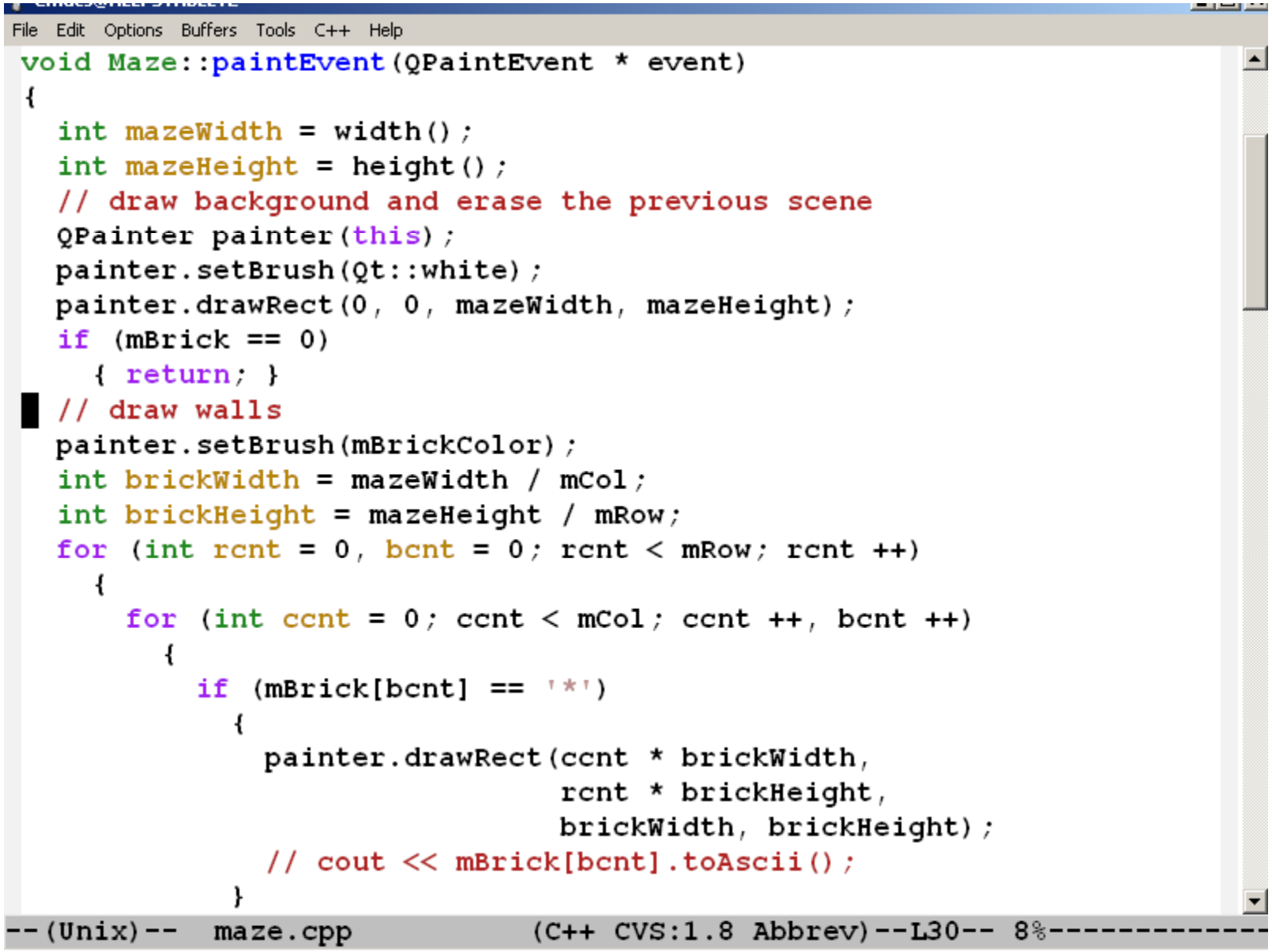
--(Unix)-- maze.h (C CVS:1.7 Abbrev)--L1--Top----------

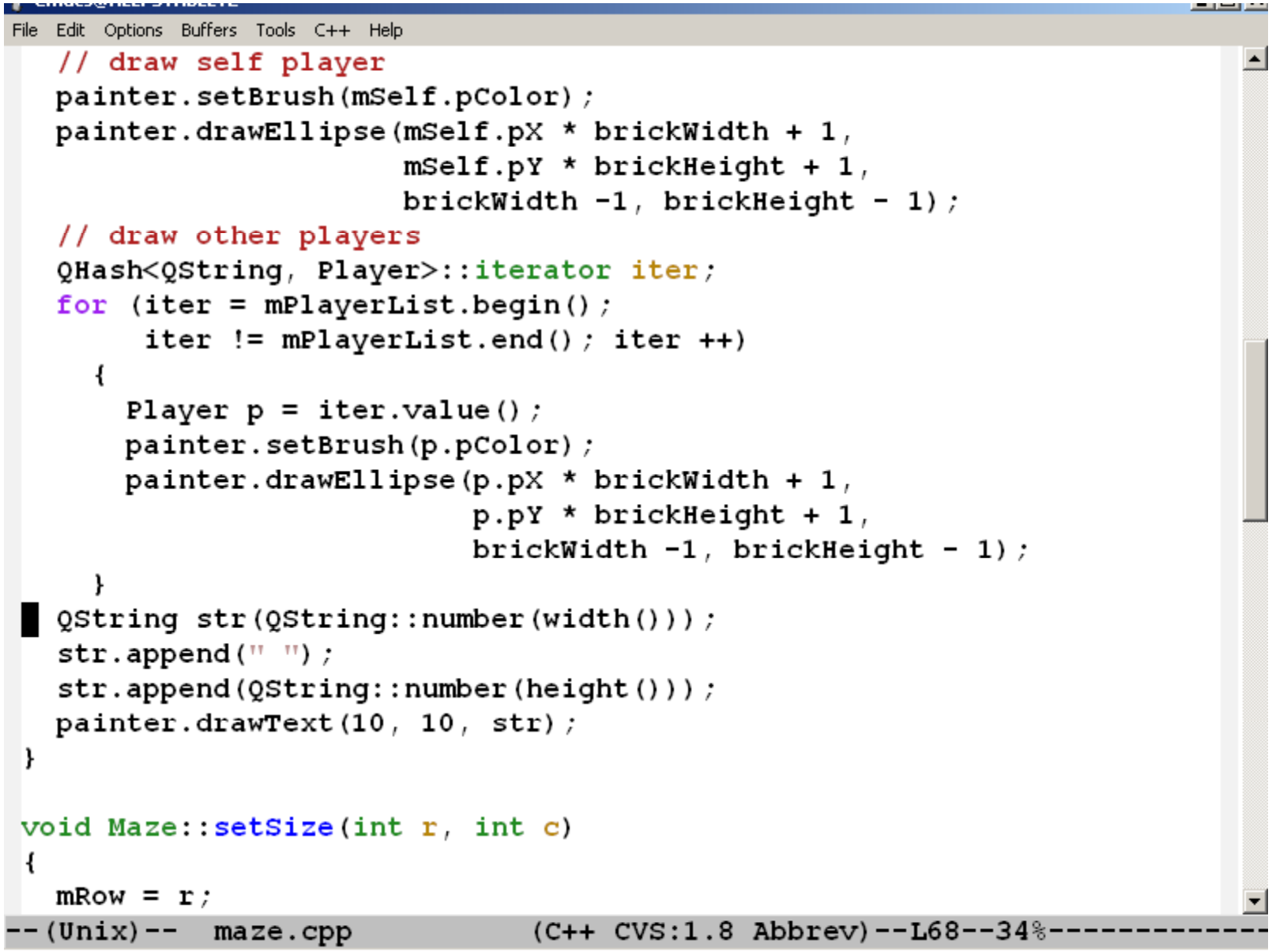

File Edit Options Buffers Tools C Help
class Maze: public QWidget
{
Q_OBJECT
public:
Maze(QWidget *parent = 0);
void setSize(int r, int c);
int getRow() const { return mRow; }
int getCol() const { return mCol; }
void setWall(const QString & wall);
void setPlayer(int x, int y, int r, int g, int b);
void addPlayer(QString id, int x, int y, int r, int g, int b);
void removePlayer(QString id);
void removeAllPlayer();
bool tryMove(int direction, int & x, int & y);
virtual ~Maze();
protected:
void paintEvent (QPaintEvent * event);
private:
int mRow;
int mCol;
const QColor mBrickColor;
QChar * mBrick;
Player mSelf;
QHash <QString, Player> mPlayerList;
};
--(Unix)-- maze.h (C CVS:1.7 Abbrev)--L34--32%----------

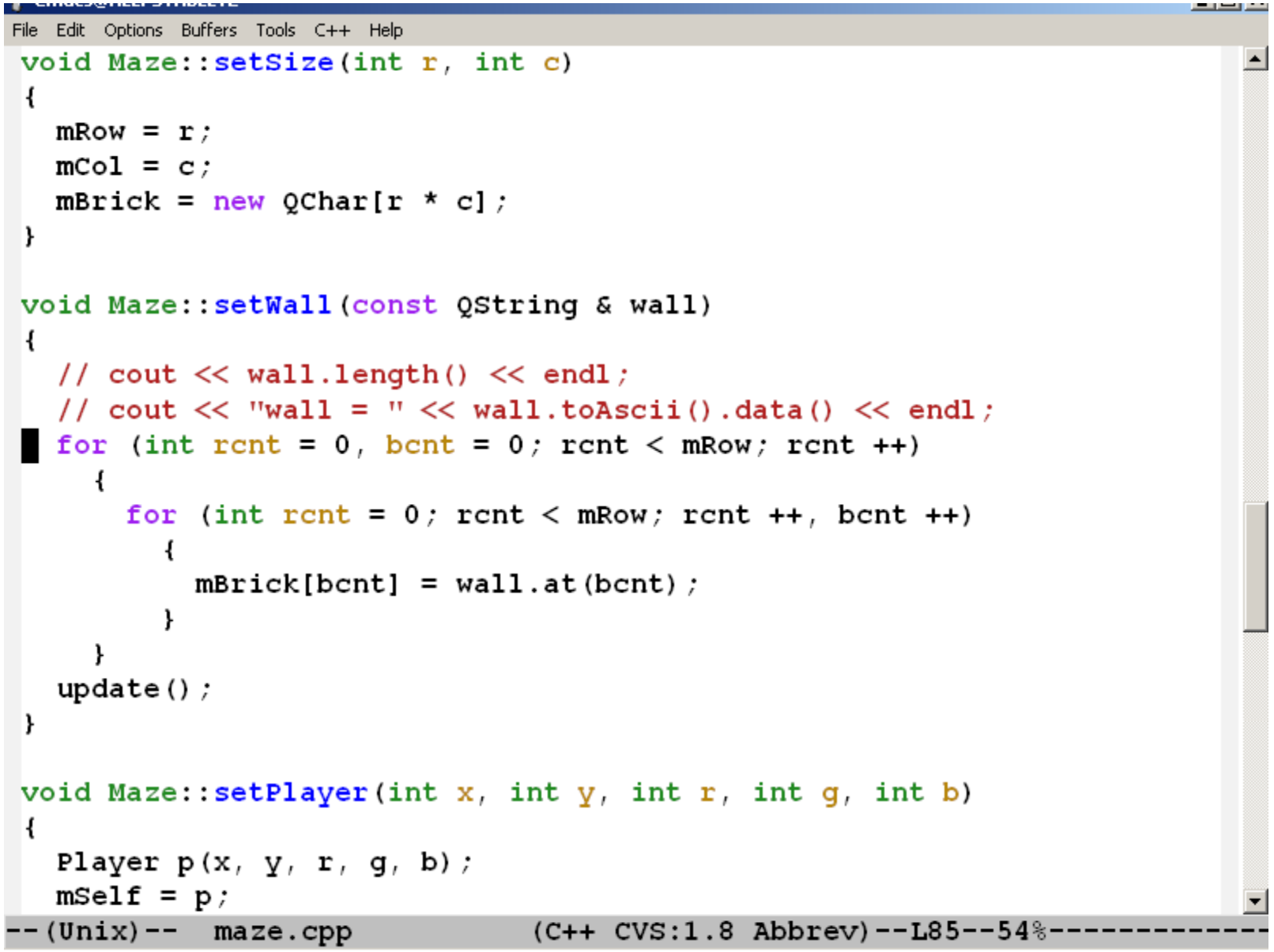
File Edit Options Buffers Tools C++ Help

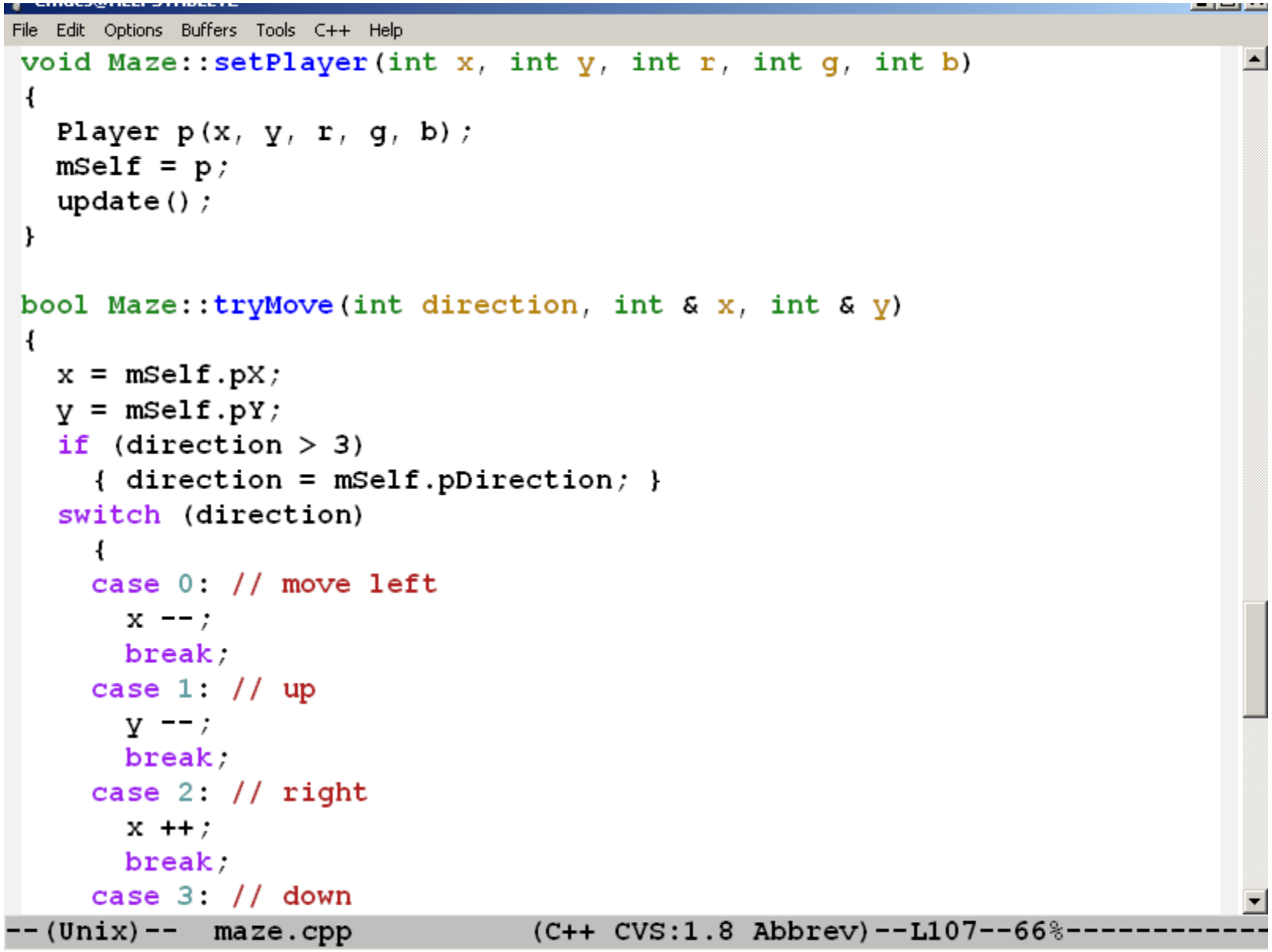
```
#include "maze.h"
#include <stdlib.h>
#include <math.h>
#include <iostream>
using namespace std;
Maze::Maze(QWidget *parent):
  QWidget(parent),
  mBrickColor(Qt::black)
{
  mBrick = 0;
  mRow = -1;
  mCol = -1;
}

Maze::~Maze() {
  if (mBrick)
    { delete mBrick; }
}

void Maze::paintEvent(QPaintEvent * event)
{
  int mazeWidth = width();
  int mazeHeight = height();
  // draw background and erase the previous scene
  QPainter painter(this);
```

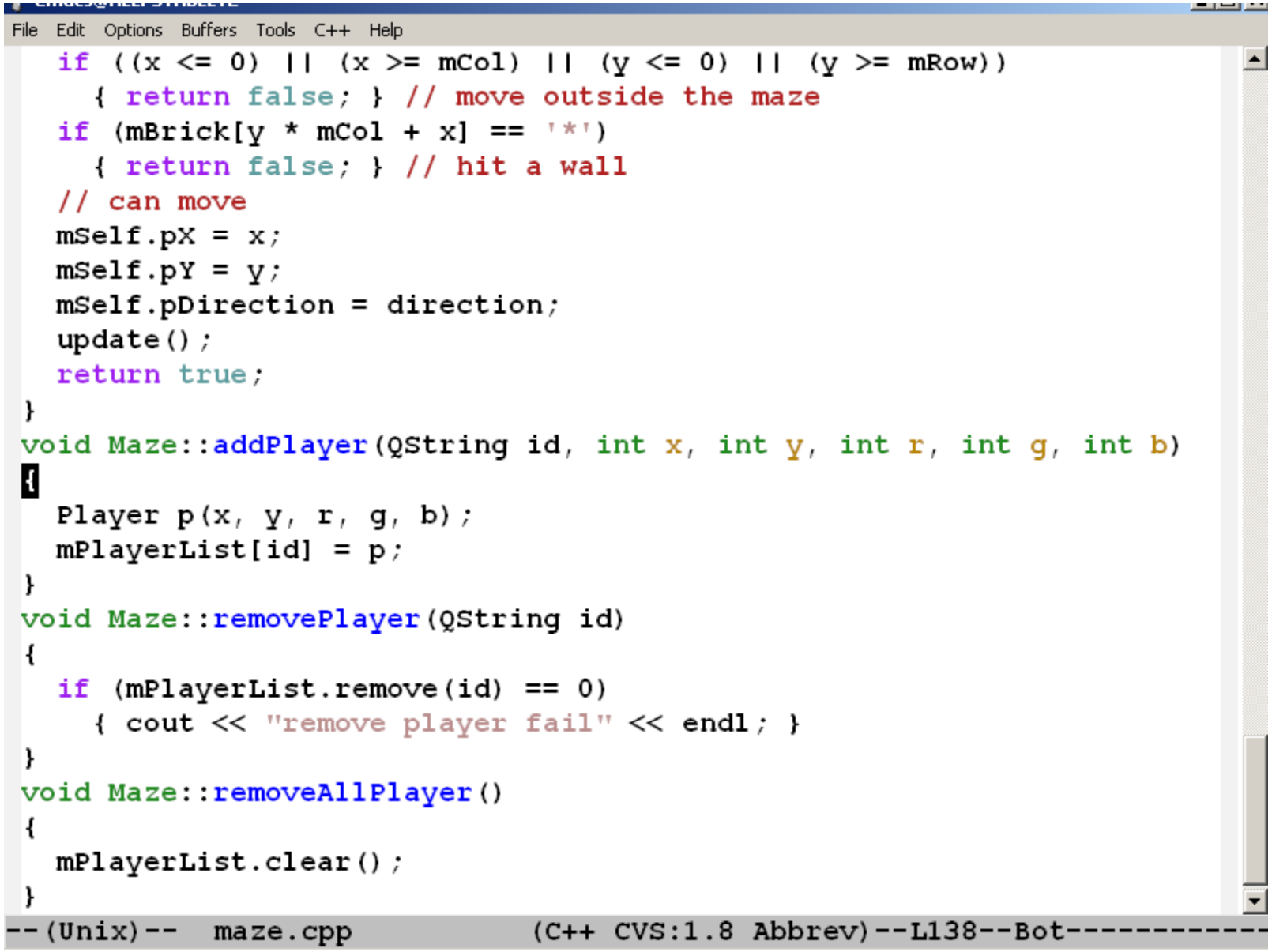
--(Unix)-- maze.cpp (C++ CVS:1.8 Abbrev)--L4--Top----------


File Edit Options Buffers Tools C++ Help
void Maze::paintEvent(QPaintEvent * event)
{
int mazeWidth = width();
int mazeHeight = height();
// draw background and erase the previous scene
QPainter painter(this);
painter.setBrush(Qt::white);
painter.drawRect(0, 0, mazeWidth, mazeHeight);
if (mBrick == 0)
{ return; }
// draw walls
painter.setBrush(mBrickColor);
int brickWidth = mazeWidth / mCol;
int brickHeight = mazeHeight / mRow;
for (int rcnt = 0, bcnt = 0; rcnt < mRow; rcnt ++)
{
for (int ccnt = 0; ccnt < mCol; ccnt ++, bcnt ++)
{
if (mBrick[bcnt] == '*')
{
painter.drawRect(ccnt * brickWidth,
rcnt * brickHeight,
brickWidth, brickHeight);
// cout << mBrick[bcnt].toAscii();
}
--(Unix)-- maze.cpp (C++ CVS:1.8 Abbrev)--L30-- 8%----------

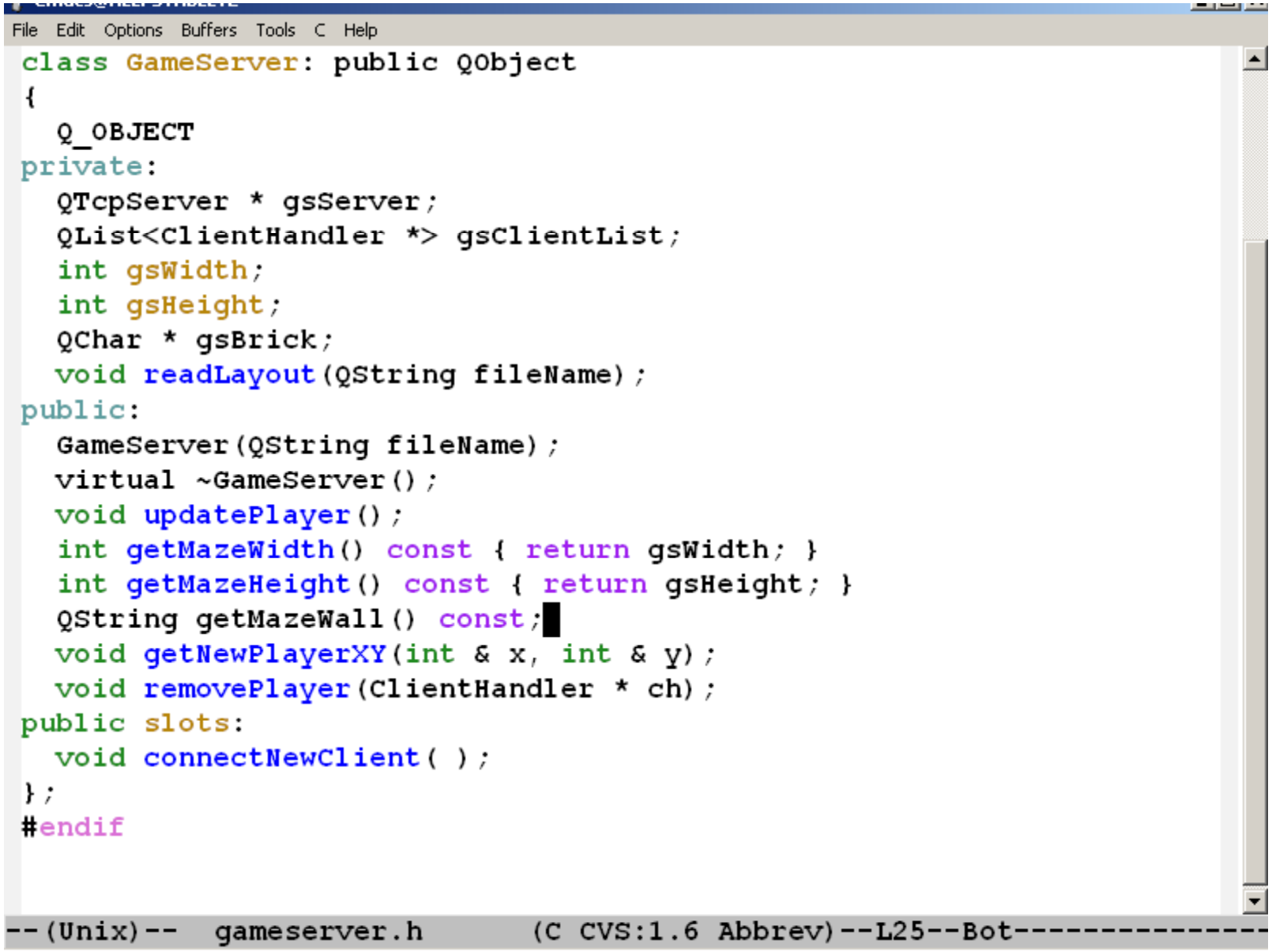

File Edit Options Buffers Tools C++ Help
// draw self player
painter.setBrush(mSelf.pColor);
painter.drawEllipse(mSelf.pX * brickWidth + 1,
mSelf.pY * brickHeight + 1,
brickWidth -1, brickHeight - 1);
// draw other players
QHash<QString, Player>::iterator iter;
for (iter = mPlayerList.begin();
iter != mPlayerList.end(); iter ++)
{
Player p = iter.value();
painter.setBrush(p.pColor);
painter.drawEllipse(p.pX * brickWidth + 1,
p.pY * brickHeight + 1,
brickWidth -1, brickHeight - 1);
}
QString str(QString::number(width()));
str.append(" ");
str.append(QString::number(height()));
painter.drawText(10, 10, str);
}
void Maze::setSize(int r, int c)
{
mRow = r;
--(Unix)-- maze.cpp (C++ CVS:1.8 Abbrev)--L68--34%----------


File Edit Options Buffers Tools C++ Help
void Maze::setSize(int r, int c)
{
mRow = r;
mCol = c;
mBrick = new QChar[r * c];
}
void Maze::setWall(const QString & wall)
{
// cout << wall.length() << endl;
// cout << "wall = " << wall.toAscii().data() << endl;
for (int rcnt = 0, bcnt = 0; rcnt < mRow; rcnt ++)
{
for (int rcnt = 0; rcnt < mRow; rcnt ++, bcnt ++)
{
mBrick[bcnt] = wall.at(bcnt);
}
}
update();
}
void Maze::setPlayer(int x, int y, int r, int g, int b)
{
Player p(x, y, r, g, b);
mSelf = p;
--(Unix)-- maze.cpp (C++ CVS:1.8 Abbrev)--L85--54%----------


File Edit Options Buffers Tools C++ Help
void Maze::setPlayer(int x, int y, int r, int g, int b)
{
Player p(x, y, r, g, b);
mSelf = p;
update();
}
bool Maze::tryMove(int direction, int & x, int & y)
{
x = mSelf.pX;
y = mSelf.pY;
if (direction > 3)
{ direction = mSelf.pDirection; }
switch (direction)
{
case 0: // move left
x --;
break;
case 1: // up
y --;
break;
case 2: // right
x ++;
break;
case 3: // down
--(Unix)-- maze.cpp (C++ CVS:1.8 Abbrev)--L107--66%----------


File Edit Options Buffers Tools C++ Help
if ((x <= 0) || (x >= mCol) || (y <= 0) || (y >= mRow))
{ return false; } // move outside the maze
if (mBrick[y * mCol + x] == '*')
{ return false; } // hit a wall
// can move
mSelf.pX = x;
mSelf.pY = y;
mSelf.pDirection = direction;
update();
return true;
}
void Maze::addPlayer(QString id, int x, int y, int r, int g, int b)
{
Player p(x, y, r, g, b);
mPlayerList[id] = p;
}
void Maze::removePlayer(QString id)
{
if (mPlayerList.remove(id) == 0)
{ cout << "remove player fail" << endl; }
}
void Maze::removeAllPlayer()
{
mPlayerList.clear();
}
--(Unix)-- maze.cpp (C++ CVS:1.8 Abbrev)--L138--Bot----------

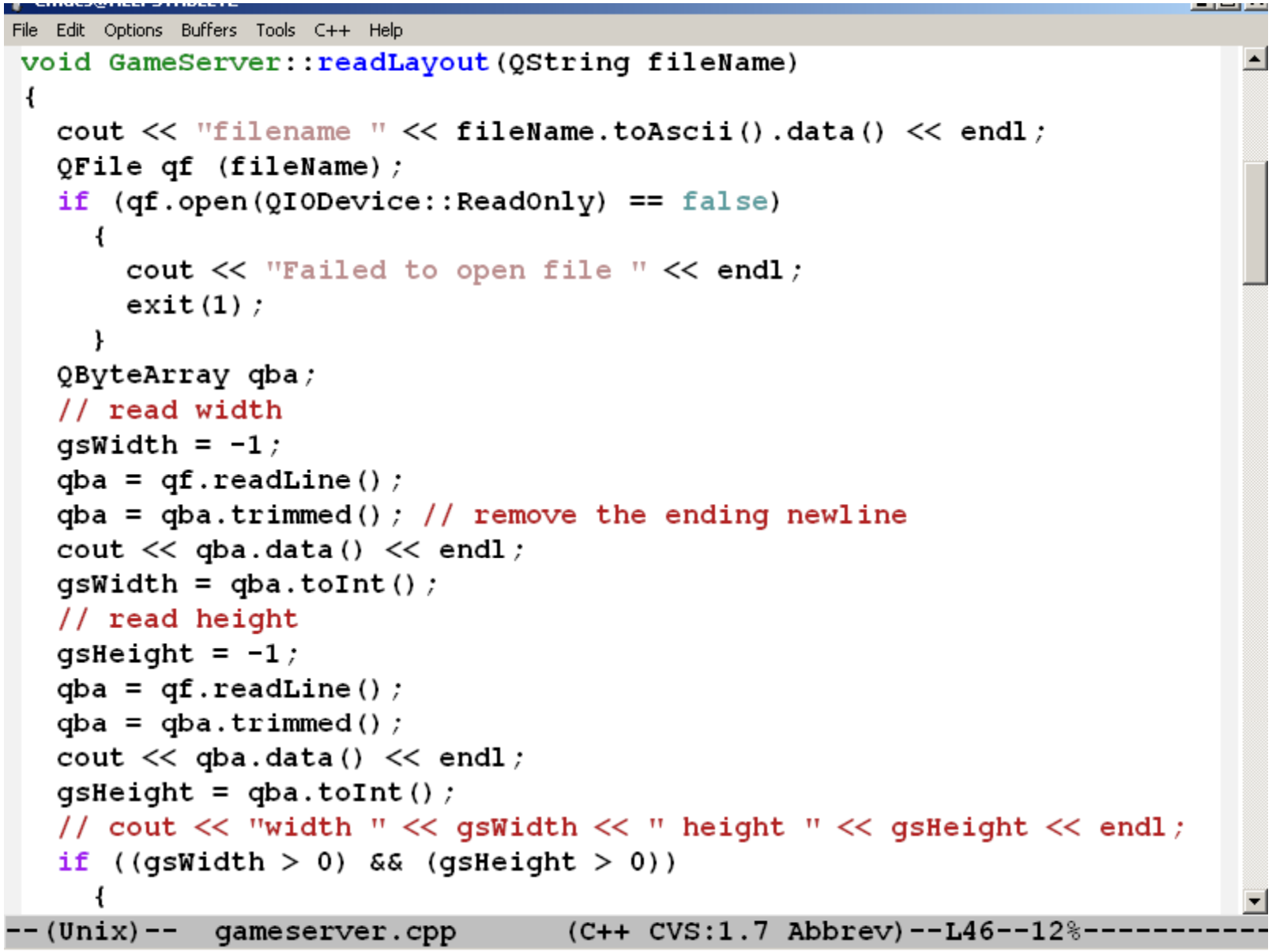
# Server

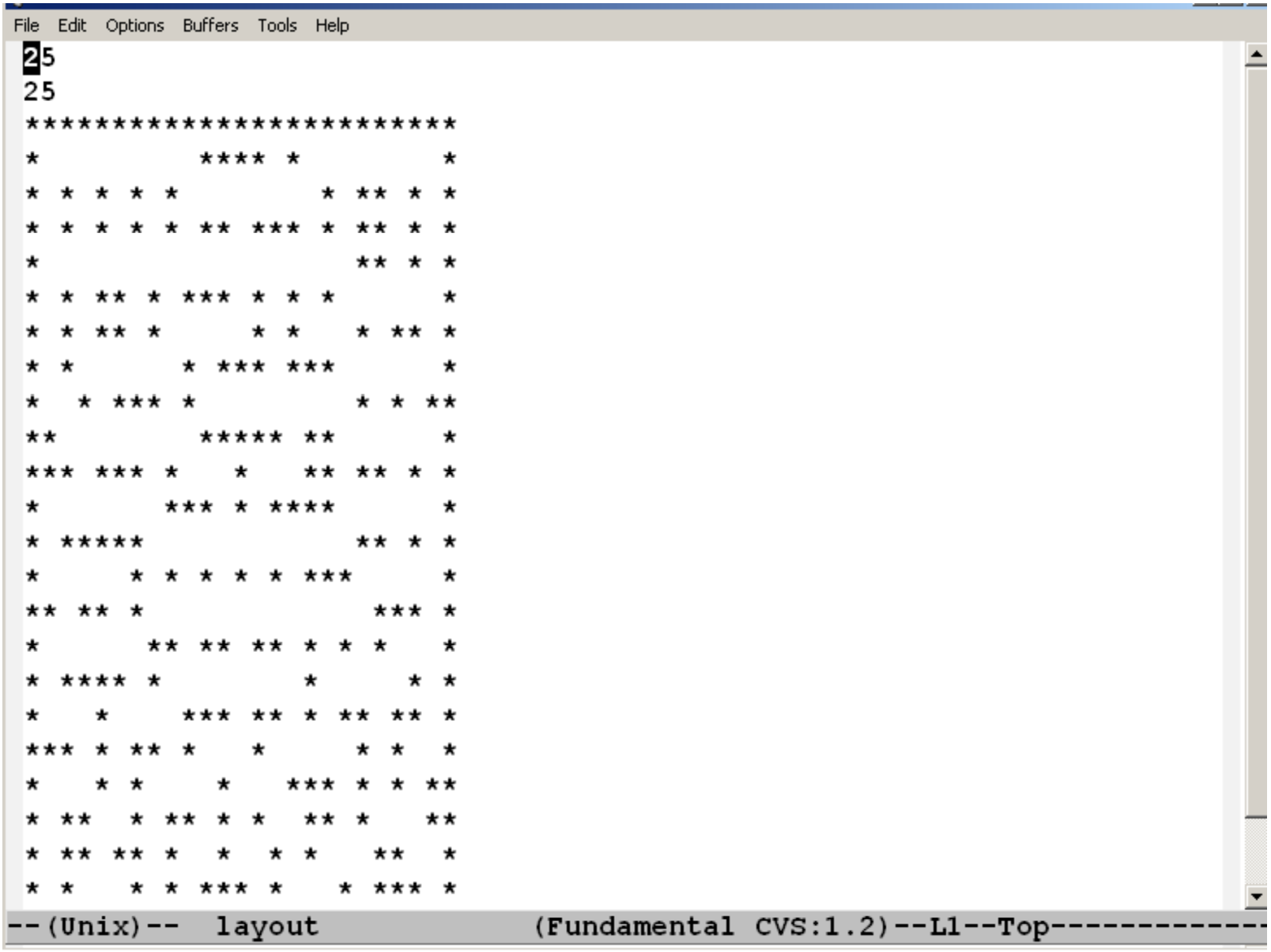

File Edit Options Buffers Tools C Help
class GameServer: public QObject
{
Q_OBJECT
private:
QTcpServer * gsServer;
QList<ClientHandler *> gsClientList;
int gsWidth;
int gsHeight;
QChar * gsBrick;
void readLayout(QString fileName);
public:
GameServer(QString fileName);
virtual ~GameServer();
void updatePlayer();
int getMazeWidth() const { return gsWidth; }
int getMazeHeight() const { return gsHeight; }
QString getMazeWall() const;
void getNewPlayerXY(int & x, int & y);
void removePlayer(ClientHandler * ch);
public slots:
void connectNewClient( );
};
#endif
--(Unix)-- gameserver.h (C CVS:1.6 Abbrev)--L25--Bot--

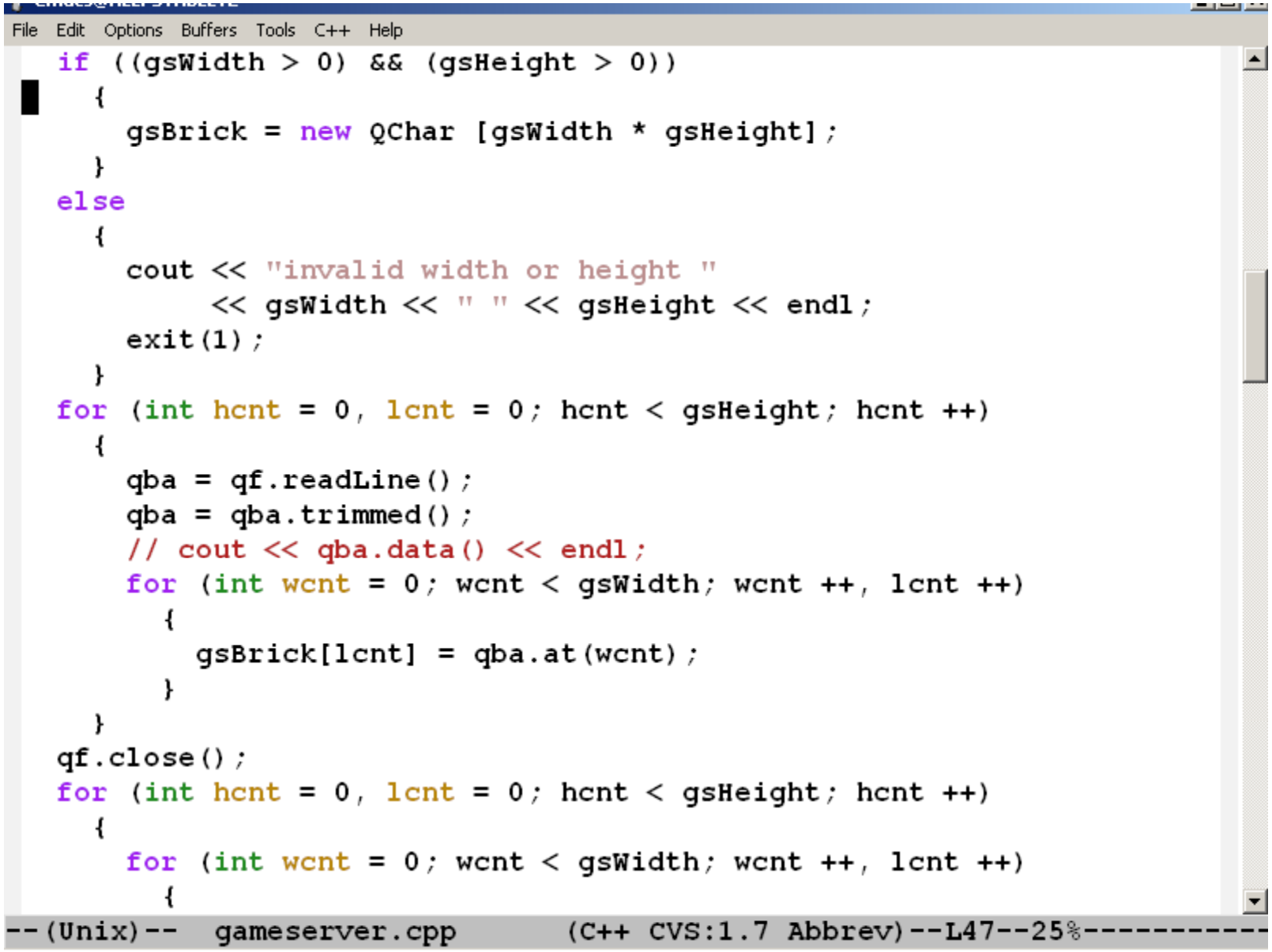
File Edit Options Buffers Tools C++ Help

```
#include "clienthandler.h"
#include "gameserver.h"
#include <iostream>
#include <stdlib.h>
using namespace std;

GameServer::GameServer(QString fileName)
{
  gsServer = new QTcpServer();
  if (! gsServer->listen())
    {
      cout << "Failed to register the server port" << endl;
      exit(1);
    }
  readLayout(fileName);
  cout << "Server port " << gsServer->serverPort() << endl;
  connect(gsServer, SIGNAL(newConnection()),
          this, SLOT(connectNewClient()));
  ClientHandler::chServer = this;
  srand(time(NULL));
}

void GameServer::readLayout(QString fileName)
{
  cout << "filename " << fileName.toAscii().data() << endl;
```

--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L6--Top----------


File Edit Options Buffers Tools C++ Help
void GameServer::readLayout(QString fileName)
{
cout << "filename " << fileName.toAscii().data() << endl;
QFile qf (fileName);
if (qf.open(QIODevice::ReadOnly) == false)
{
cout << "Failed to open file " << endl;
exit(1);
}
QByteArray qba;
// read width
gsWidth = -1;
qba = qf.readLine();
qba = qba.trimmed(); // remove the ending newline
cout << qba.data() << endl;
gsWidth = qba.toInt();
// read height
gsHeight = -1;
qba = qf.readLine();
qba = qba.trimmed();
cout << qba.data() << endl;
gsHeight = qba.toInt();
// cout << "width " << gsWidth << " height " << gsHeight << endl;
if ((gsWidth > 0) && (gsHeight > 0))
{
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L46--12%----------

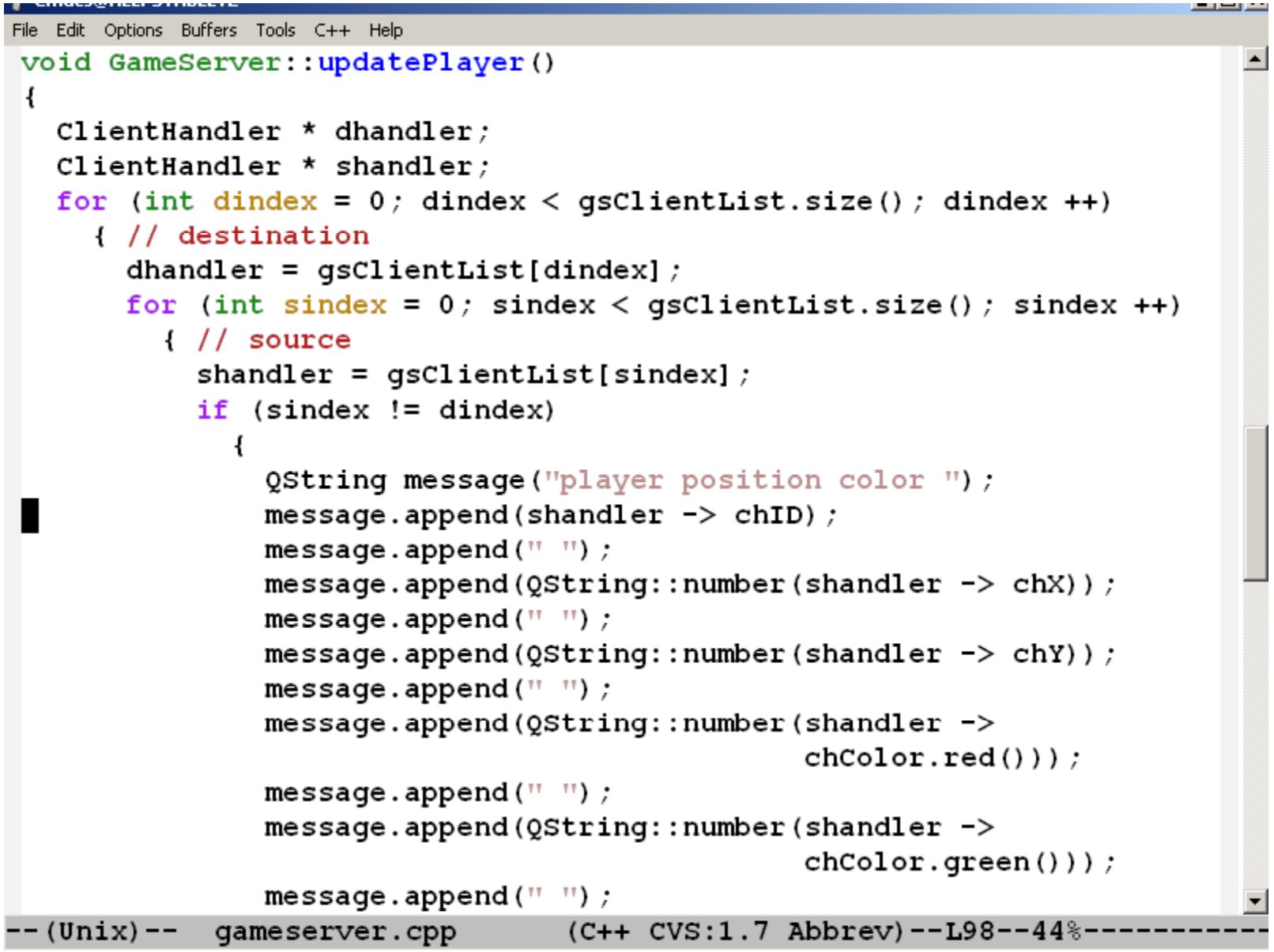

File Edit Options Buffers Tools Help
25
25
--(Unix)-- layout (Fundamental CVS:1.2)--L1--Top----------

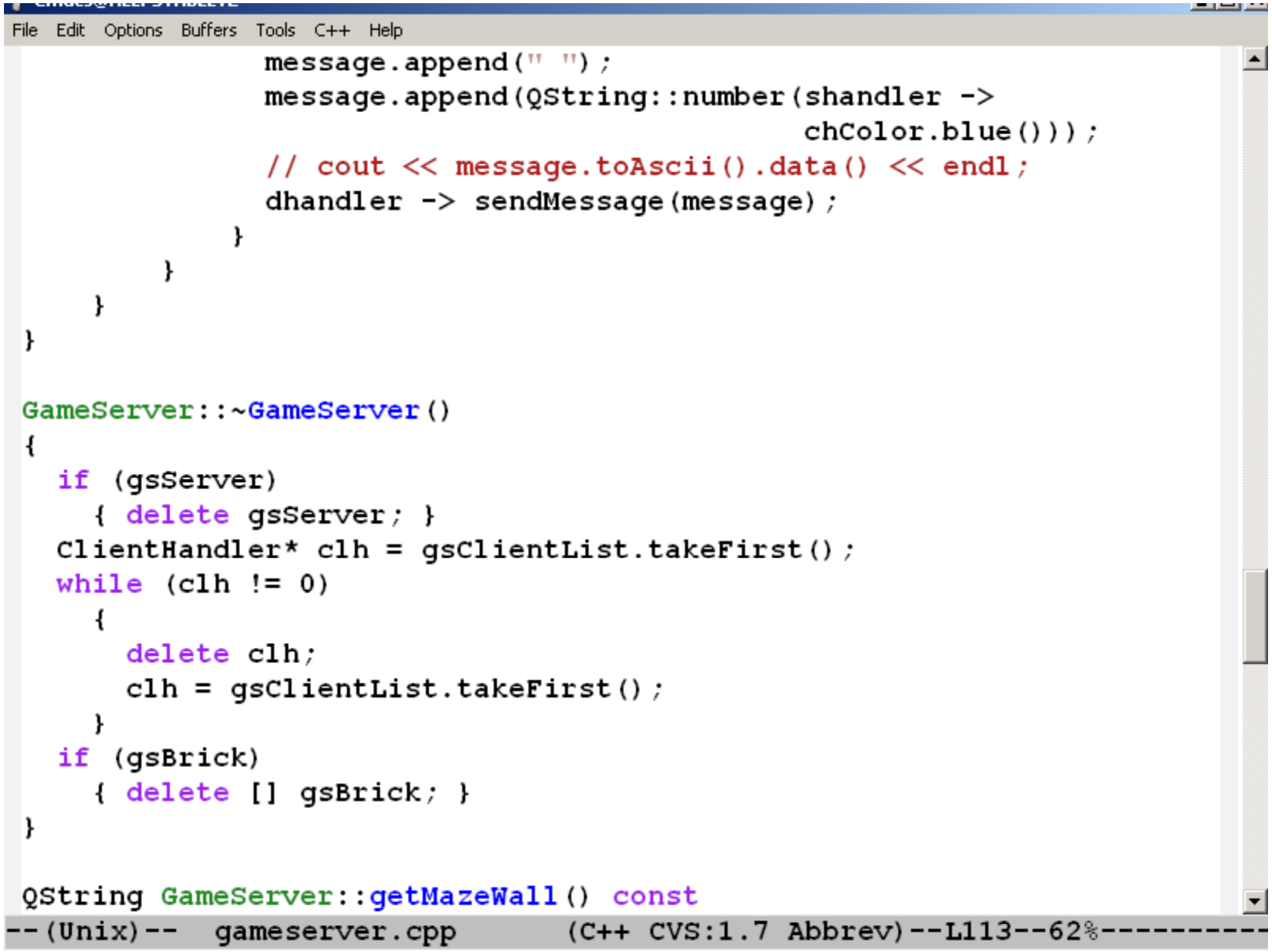

File Edit Options Buffers Tools C++ Help
if ((gsWidth > 0) && (gsHeight > 0))
{
gsBrick = new QChar [gsWidth * gsHeight];
}
else
{
cout << "invalid width or height "
<< gsWidth << " " << gsHeight << endl;
exit(1);
}
for (int hcnt = 0, lcnt = 0; hcnt < gsHeight; hcnt ++)
{
qba = qf.readLine();
qba = qba.trimmed();
// cout << qba.data() << endl;
for (int wcnt = 0; wcnt < gsWidth; wcnt ++, lcnt ++)
{
gsBrick[lcnt] = qba.at(wcnt);
}
}
qf.close();
for (int hcnt = 0, lcnt = 0; hcnt < gsHeight; hcnt ++)
{
for (int wcnt = 0; wcnt < gsWidth; wcnt ++, lcnt ++)
{
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L47--25%----------

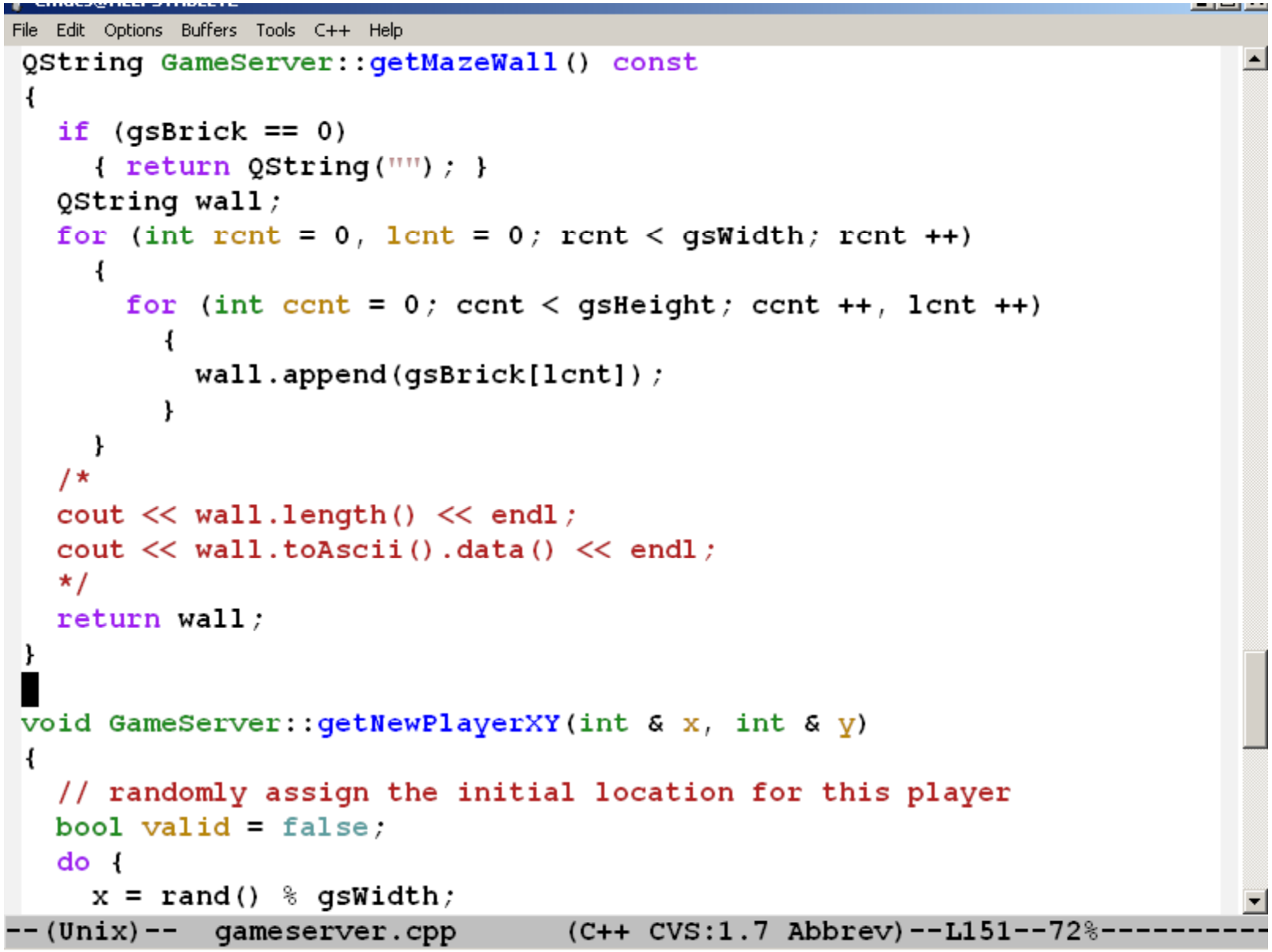
File Edit Options Buffers Tools C++ Help

```
void GameServer::connectNewClient()
{
  QTcpSocket* socket = gsServer->nextPendingConnection();
  ClientHandler* clh = new ClientHandler( socket );
  gsClientList.push_back(clh);
  cout << "A new client connected " << endl;
}

void GameServer::updatePlayer()
{
  ClientHandler * dhandler;
  ClientHandler * shandler;
  for (int dindex = 0; dindex < gsClientList.size(); dindex ++)
    { // destination
      dhandler = gsClientList[dindex];
      for (int sindex = 0; sindex < gsClientList.size(); sindex ++)
        { // source
          shandler = gsClientList[sindex];
          if (sindex != dindex)
            {
              QString message("player position color ");
              message.append(shandler -> chID);
              message.append(" ");
              message.append(QString::number(shandler -> chX));
              message.append(" ");
```

--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L83--40%----------


File Edit Options Buffers Tools C++ Help
void GameServer::updatePlayer()
{
ClientHandler * dhandler;
ClientHandler * shandler;
for (int dindex = 0; dindex < gsClientList.size(); dindex ++)
{ // destination
dhandler = gsClientList[dindex];
for (int sindex = 0; sindex < gsClientList.size(); sindex ++)
{ // source
shandler = gsClientList[sindex];
if (sindex != dindex)
{
QString message("player position color ");
message.append(shandler -> chID);
message.append(" ");
message.append(QString::number(shandler -> chX));
message.append(" ");
message.append(QString::number(shandler -> chY));
message.append(" ");
message.append(QString::number(shandler ->
chColor.red()));
message.append(" ");
message.append(QString::number(shandler ->
chColor.green()));
message.append(" ");
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L98--44%----------

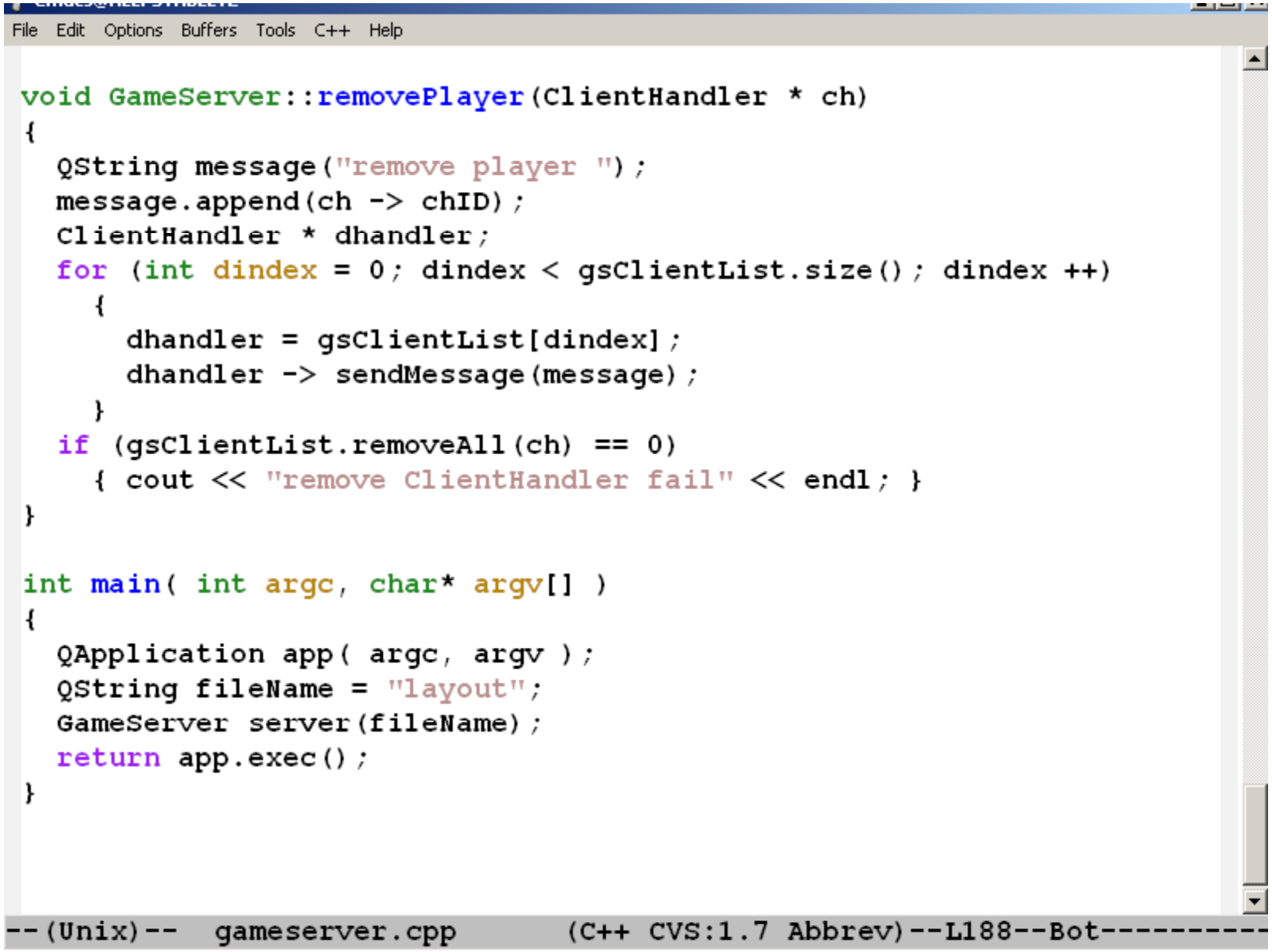

File Edit Options Buffers Tools C++ Help
message.append(" ");
message.append(QString::number(shandler ->
chColor.blue()));
// cout << message.toAscii().data() << endl;
dhandler -> sendMessage(message);
}
}
}
}
GameServer::~GameServer()
{
if (gsServer)
{ delete gsServer; }
ClientHandler* clh = gsClientList.takeFirst();
while (clh != 0)
{
delete clh;
clh = gsClientList.takeFirst();
}
if (gsBrick)
{ delete [] gsBrick; }
}
QString GameServer::getMazeWall() const
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L113--62%----------


File Edit Options Buffers Tools C++ Help
QString GameServer::getMazeWall() const
{
if (gsBrick == 0)
{ return QString(""); }
QString wall;
for (int rcnt = 0, lcnt = 0; rcnt < gsWidth; rcnt ++)
{
for (int ccnt = 0; ccnt < gsHeight; ccnt ++, lcnt ++)
{
wall.append(gsBrick[lcnt]);
}
}
/*
cout << wall.length() << endl;
cout << wall.toAscii().data() << endl;
*/
return wall;
}
void GameServer::getNewPlayerXY(int & x, int & y)
{
// randomly assign the initial location for this player
bool valid = false;
do {
x = rand() % gsWidth;
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L151--72%----------

File Edit Options Buffers Tools C++ Help

```
void GameServer::getNewPlayerXY(int & x, int & y)
{
  // randomly assign the initial location for this player
  bool valid = false;
  do {
    x = rand() % gsWidth;
    y = rand() % gsHeight;
    if (gsBrick[y * gsWidth + x] != '*')
      { valid = true; }
  }
  while (valid == false);
  // should check whether the location is far away from all existing
  // players. for simplicity, it does not check here
}

void GameServer::removePlayer(ClientHandler * ch)
{
  QString message("remove player ");
  message.append(ch -> chID);
  ClientHandler * dhandler;
  for (int dindex = 0; dindex < gsClientList.size(); dindex ++)
    {
      dhandler = gsClientList[dindex];
      dhandler -> sendMessage(message);
    }
```

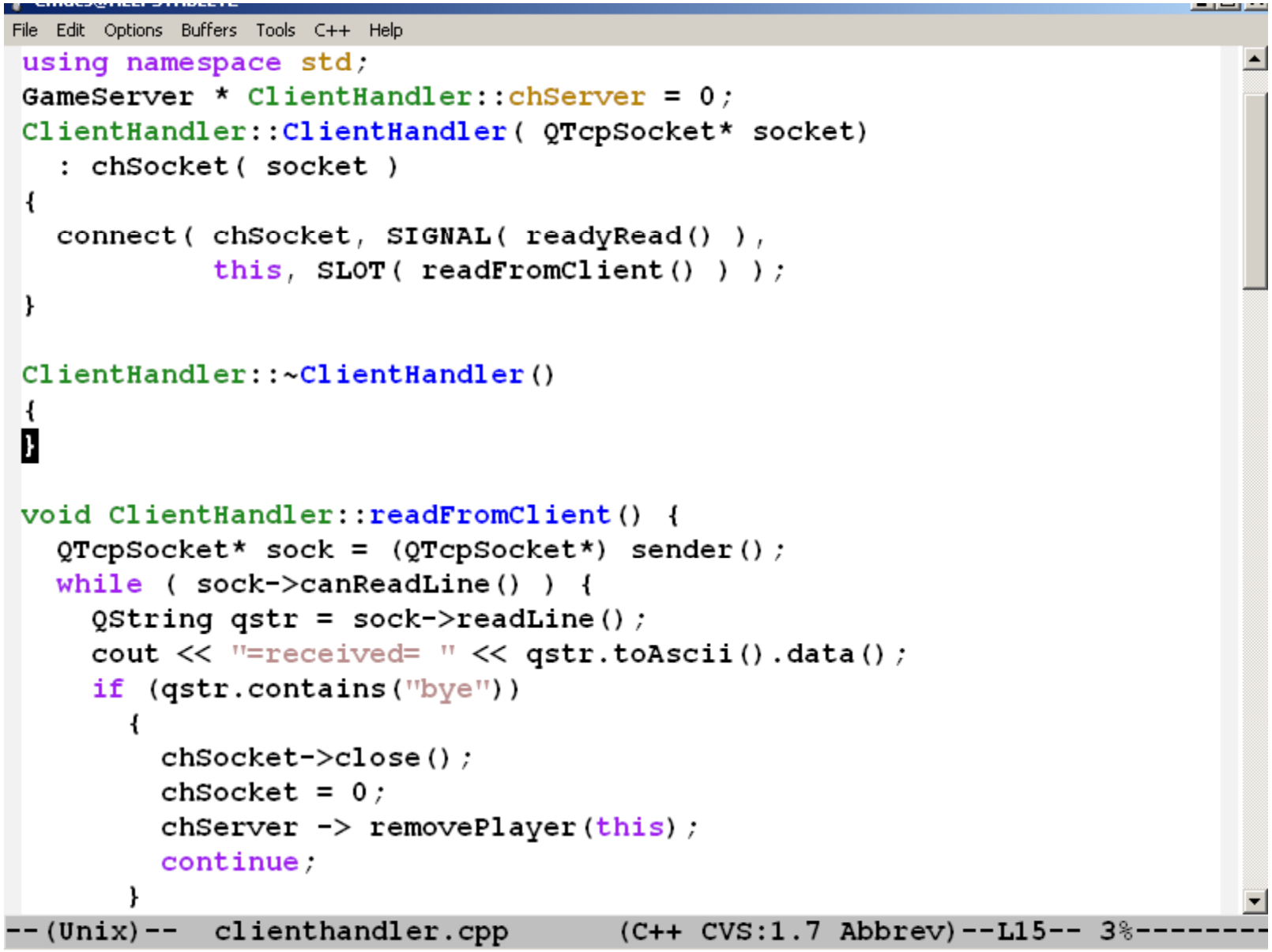
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L166--80%----------

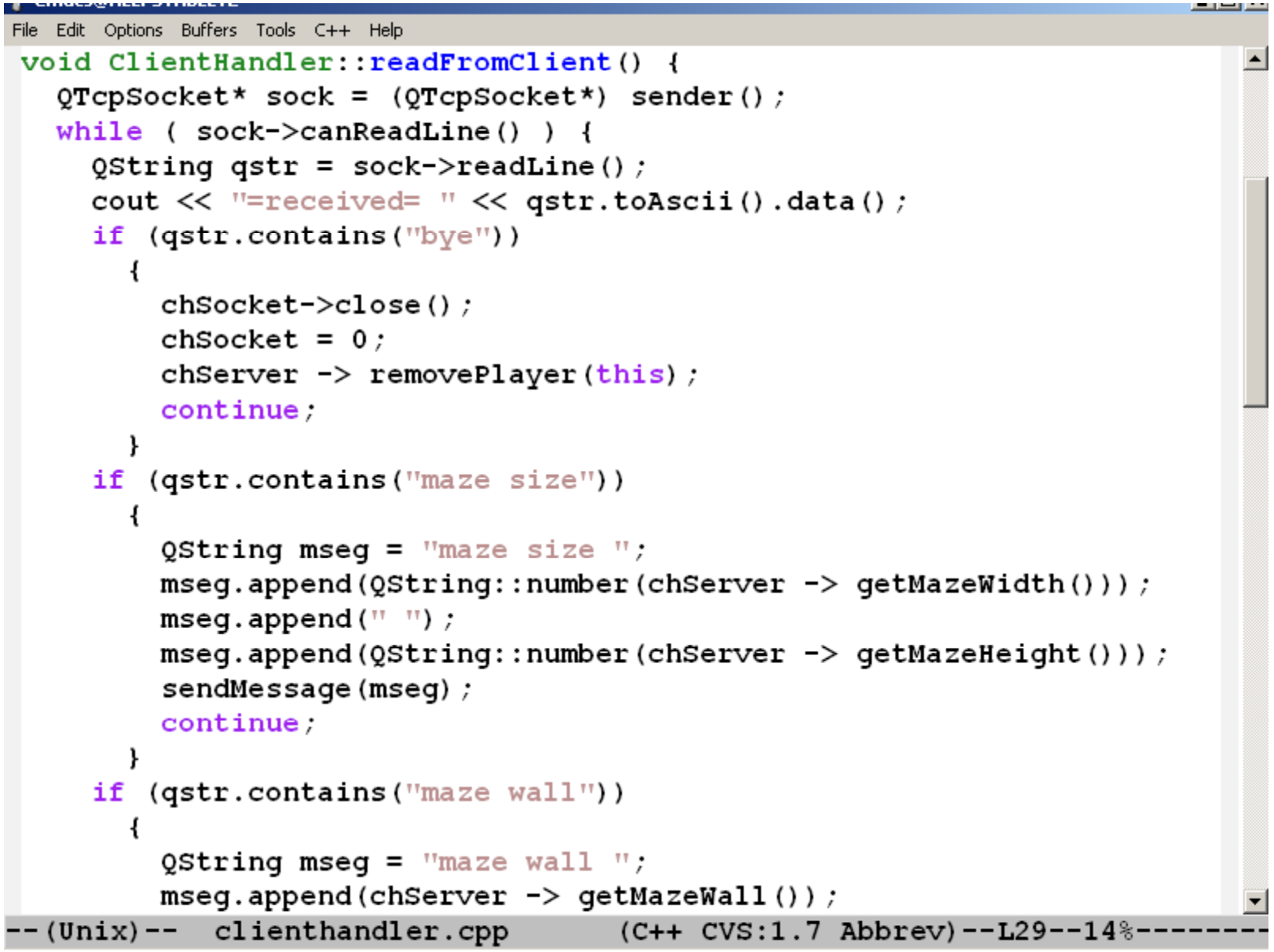

File Edit Options Buffers Tools C++ Help
void GameServer::removePlayer(ClientHandler * ch)
{
QString message("remove player ");
message.append(ch -> chID);
ClientHandler * dhandler;
for (int dindex = 0; dindex < gsClientList.size(); dindex ++)
{
dhandler = gsClientList[dindex];
dhandler -> sendMessage(message);
}
if (gsClientList.removeAll(ch) == 0)
{ cout << "remove ClientHandler fail" << endl; }
}
int main( int argc, char* argv[] )
{
QApplication app( argc, argv );
QString fileName = "layout";
GameServer server(fileName);
return app.exec();
}
--(Unix)-- gameserver.cpp (C++ CVS:1.7 Abbrev)--L188--Bot----------

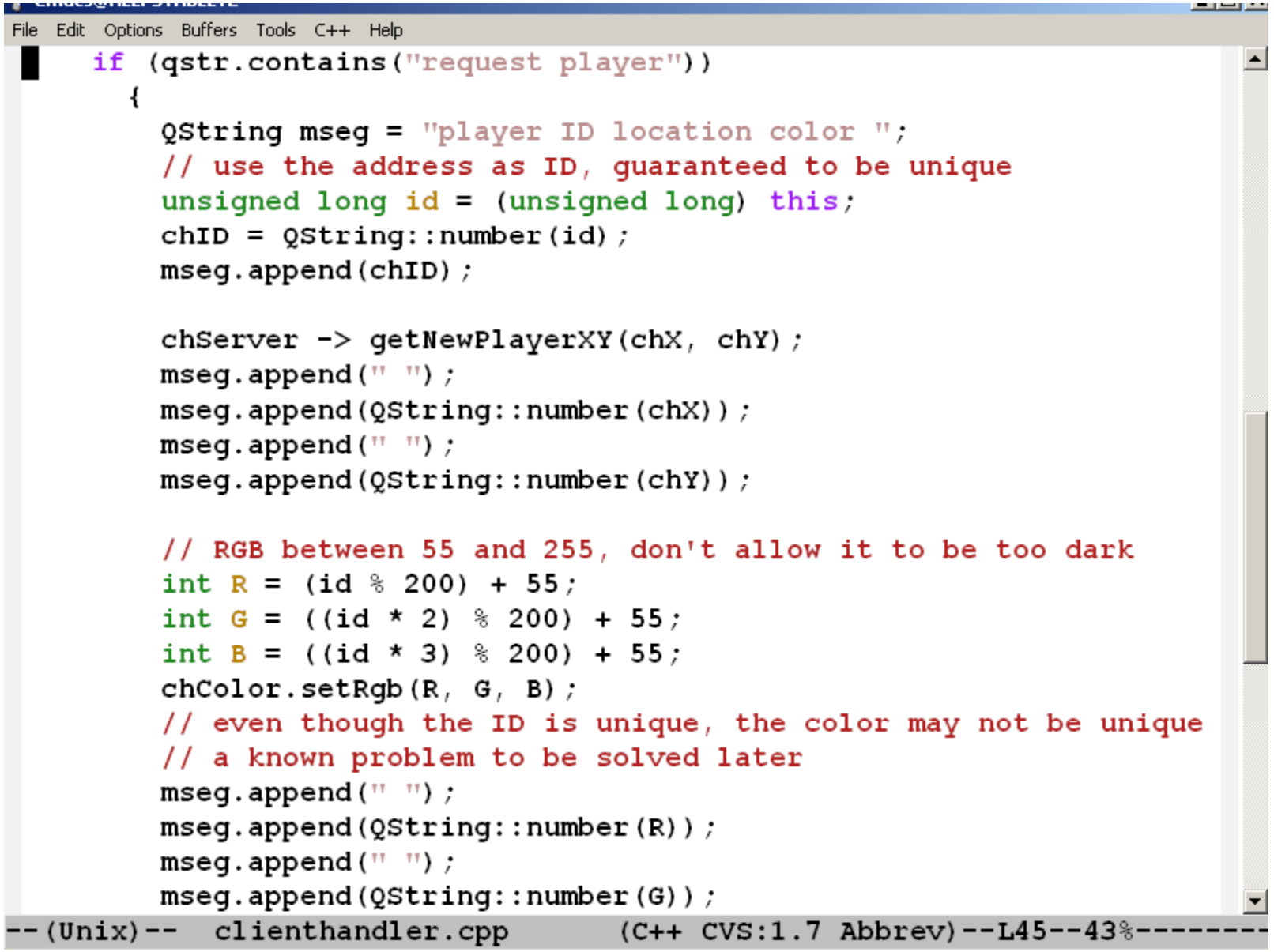
File Edit Options Buffers Tools C Help

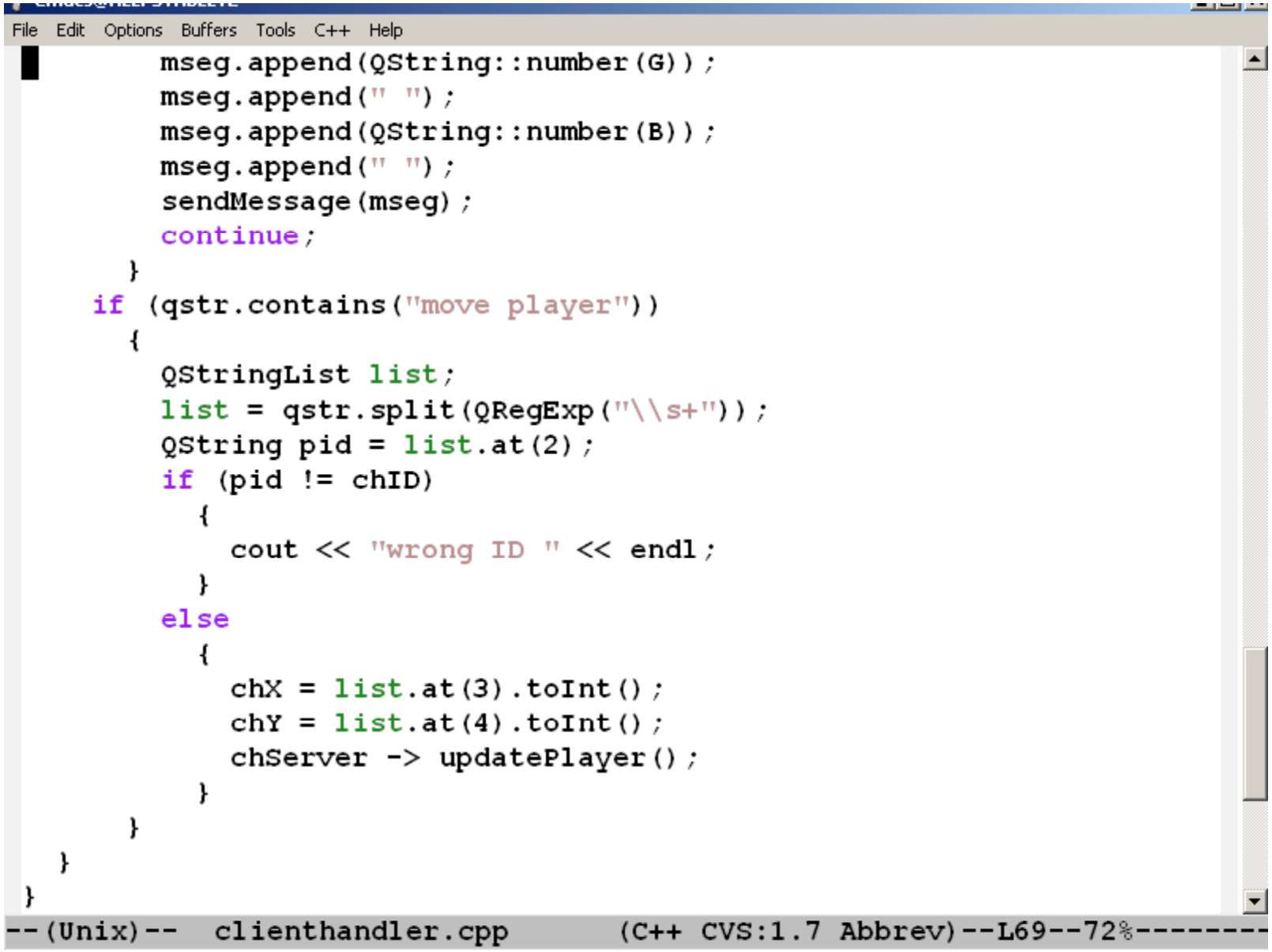
```
#include <QApplication>
#include <vector>
class GameServer;
class ClientHandler : public QObject
{
  Q_OBJECT
private:
  QTcpSocket* chSocket;
  // QTextStream* chTextStream;
  int chX;
  int chY;
  QColor chColor;
  QString chID;
  void sendMessage(const QString str);
  void sendMessage(const char * str);
public:
  ClientHandler( QTcpSocket* sock);
  virtual ~ClientHandler();
  void getXYColor(int & x, int & y, QColor & c)
    { x = chX; y = chY; c = chColor; }
  static GameServer * chServer;
private slots:
  void readFromClient();
  friend class GameServer;
};
```

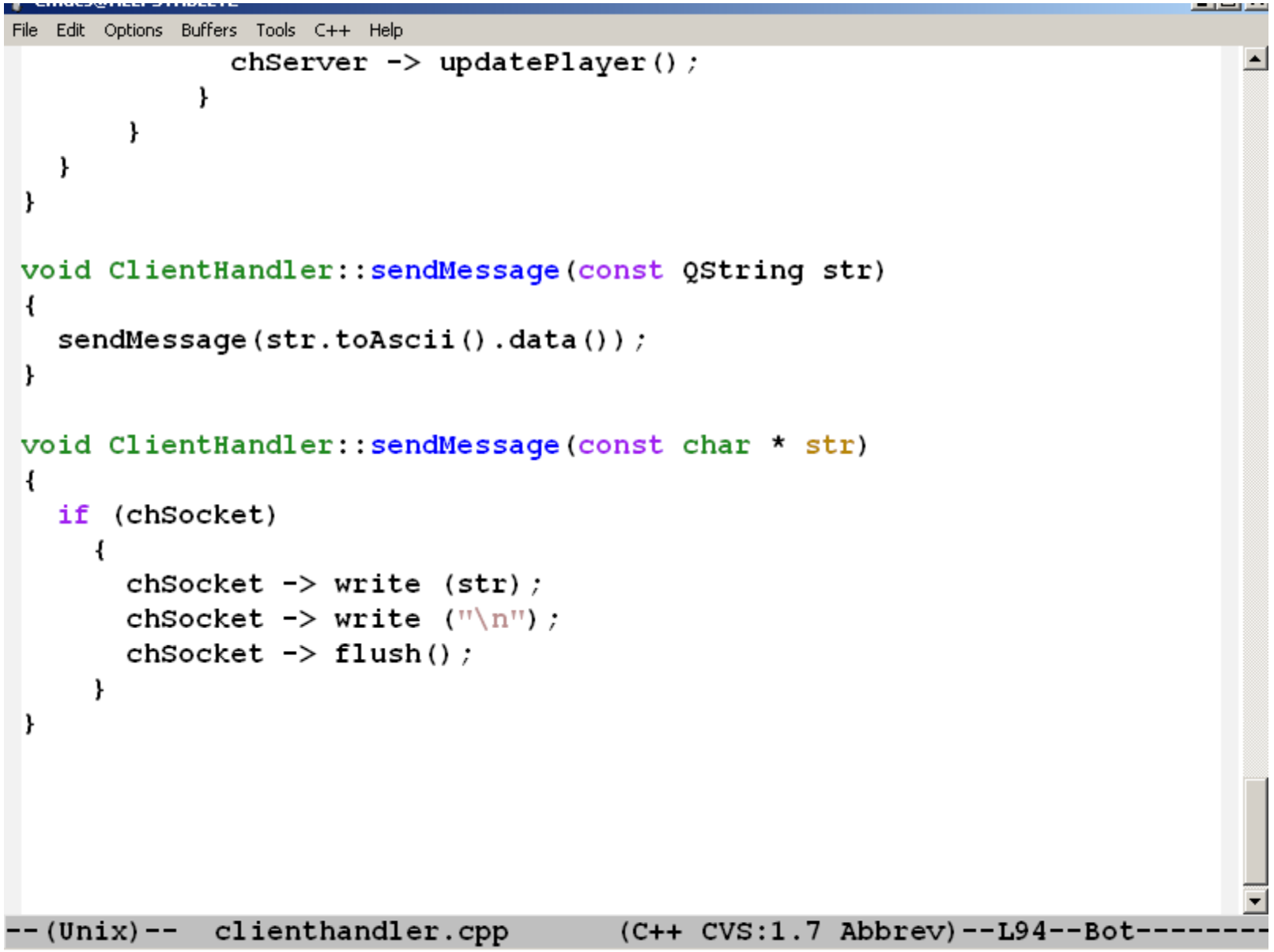
--(Unix)-- clienthandler.h (C CVS:1.5 Abbrev)--L6--13%----------


File Edit Options Buffers Tools C++ Help
using namespace std;
GameServer * ClientHandler::chServer = 0;
ClientHandler::ClientHandler( QTcpSocket* socket)
: chSocket( socket )
{
connect( chSocket, SIGNAL( readyRead() ),
this, SLOT( readFromClient() ) );
}
ClientHandler::~ClientHandler()
{
}
void ClientHandler::readFromClient() {
QTcpSocket* sock = (QTcpSocket*) sender();
while ( sock->canReadLine() ) {
QString qstr = sock->readLine();
cout << "=received= " << qstr.toAscii().data();
if (qstr.contains("bye"))
{
chSocket->close();
chSocket = 0;
chServer -> removePlayer(this);
continue;
}
--(Unix)-- clienthandler.cpp (C++ CVS:1.7 Abbrev)--L15-- 3%--------


File Edit Options Buffers Tools C++ Help
void ClientHandler::readFromClient() {
QTcpSocket* sock = (QTcpSocket*) sender();
while ( sock->canReadLine() ) {
QString qstr = sock->readLine();
cout << "=received= " << qstr.toAscii().data();
if (qstr.contains("bye"))
{
chSocket->close();
chSocket = 0;
chServer -> removePlayer(this);
continue;
}
if (qstr.contains("maze size"))
{
QString mseg = "maze size ";
mseg.append(QString::number(chServer -> getMazeWidth()));
mseg.append(" ");
mseg.append(QString::number(chServer -> getMazeHeight()));
sendMessage(mseg);
continue;
}
if (qstr.contains("maze wall"))
{
QString mseg = "maze wall ";
mseg.append(chServer -> getMazeWall());
--(Unix)-- clienthandler.cpp (C++ CVS:1.7 Abbrev)--L29--14%--------


File Edit Options Buffers Tools C++ Help
if (qstr.contains("request player"))
{
QString mseg = "player ID location color ";
// use the address as ID, guaranteed to be unique
unsigned long id = (unsigned long) this;
chID = QString::number(id);
mseg.append(chID);
chServer -> getNewPlayerXY(chX, chY);
mseg.append(" ");
mseg.append(QString::number(chX));
mseg.append(" ");
mseg.append(QString::number(chY));
// RGB between 55 and 255, don't allow it to be too dark
int R = (id % 200) + 55;
int G = ((id * 2) % 200) + 55;
int B = ((id * 3) % 200) + 55;
chColor.setRgb(R, G, B);
// even though the ID is unique, the color may not be unique
// a known problem to be solved later
mseg.append(" ");
mseg.append(QString::number(R));
mseg.append(" ");
mseg.append(QString::number(G));
--(Unix)-- clienthandler.cpp (C++ CVS:1.7 Abbrev)--L45--43%--------


File Edit Options Buffers Tools C++ Help
mseg.append(QString::number(G));
mseg.append(" ");
mseg.append(QString::number(B));
mseg.append(" ");
sendMessage(mseg);
continue;
}
if (qstr.contains("move player"))
{
QStringList list;
list = qstr.split(QRegExp("\\s+"));
QString pid = list.at(2);
if (pid != chID)
{
cout << "wrong ID " << endl;
}
else
{
chX = list.at(3).toInt();
chY = list.at(4).toInt();
chServer -> updatePlayer();
}
}
}
}
--(Unix)-- clienthandler.cpp (C++ CVS:1.7 Abbrev)--L69--72%--------


File Edit Options Buffers Tools C++ Help
chServer -> updatePlayer();
}
}
}
}
void ClientHandler::sendMessage(const QString str)
{
sendMessage(str.toAscii().data());
}
void ClientHandler::sendMessage(const char * str)
{
if (chSocket)
{
chSocket -> write (str);
chSocket -> write ("\n");
chSocket -> flush();
}
}
--(Unix)-- clienthandler.cpp (C++ CVS:1.7 Abbrev)--L94--Bot--------

# Improvements

- respond to keyboard and mouse
- place treasures and traps
- detect collisions of player
- assign different roles to the players
- use images for players, add sound
- load multiple layouts

- handle multiple games, allow players to register
- logging and debugging
- prevent denial-of-service attacks

# ECE 462
# Object-Oriented Programming using C++ and Java

# Java Remote Method Invocation

Yung-Hsiang Lu
yunglu@purdue.edu

# Rethink Function Call

## Caller

```
...
pi = computePi(40);
...
```

## Callee

```
LongFloat computePi(int numDigits)
{
    ....
    ....
}
```

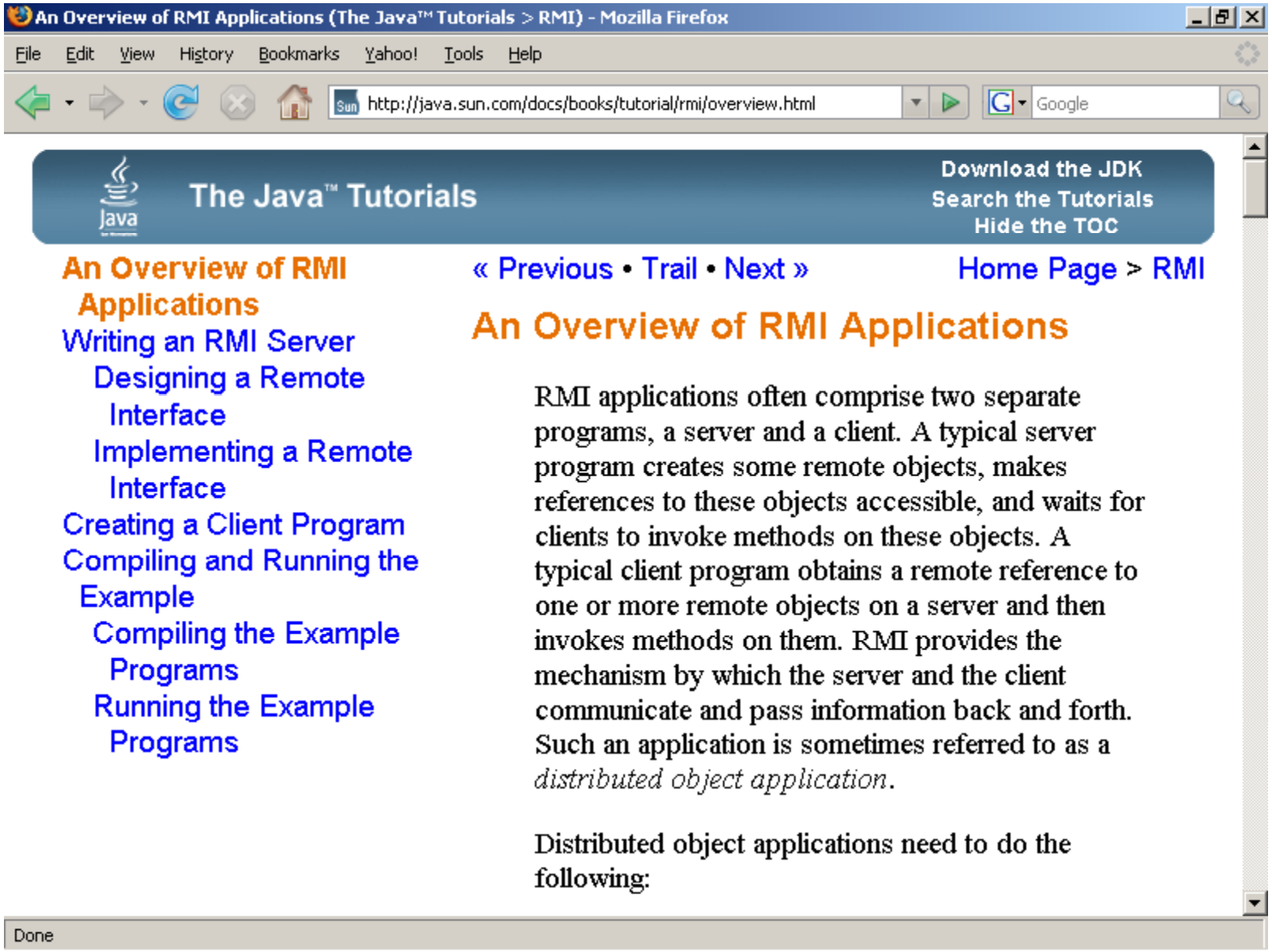
The caller does not care how computePi obtain the result.

# How to Implement the Callee?

- compute PI
- use a lookup table
- ask another machine to compute

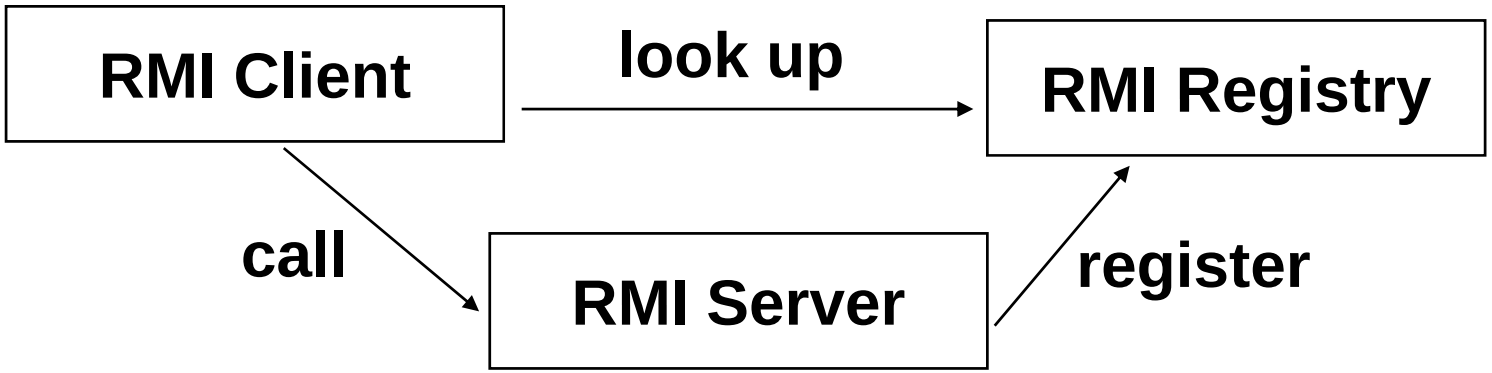
```
LongFloat computePi(int numDigits)
{
    connect to a server
    send the request to the server
    wait for the response
    return the result to the caller
}
```

# Remote Method Invocation


An Overview of RMI Applications (The Java™ Tutorials > RMI) - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/docs/books/tutorial/rmi/overview.html
Google
The Java™ Tutorials
Download the JDK
Search the Tutorials
Hide the TOC
An Overview of RMI Applications
Writing an RMI Server
Designing a Remote Interface
Implementing a Remote Interface
Creating a Client Program
Compiling and Running the Example
Compiling the Example Programs
Running the Example Programs
« Previous • Trail • Next »
Home Page > RMI
An Overview of RMI Applications
RMI applications often comprise two separate programs, a server and a client. A typical server program creates some remote objects, makes references to these objects accessible, and waits for clients to invoke methods on these objects. A typical client program obtains a remote reference to one or more remote objects on a server and then invokes methods on them. RMI provides the mechanism by which the server and the client communicate and pass information back and forth. Such an application is sometimes referred to as a distributed object application.
Distributed object applications need to do the following:
Done

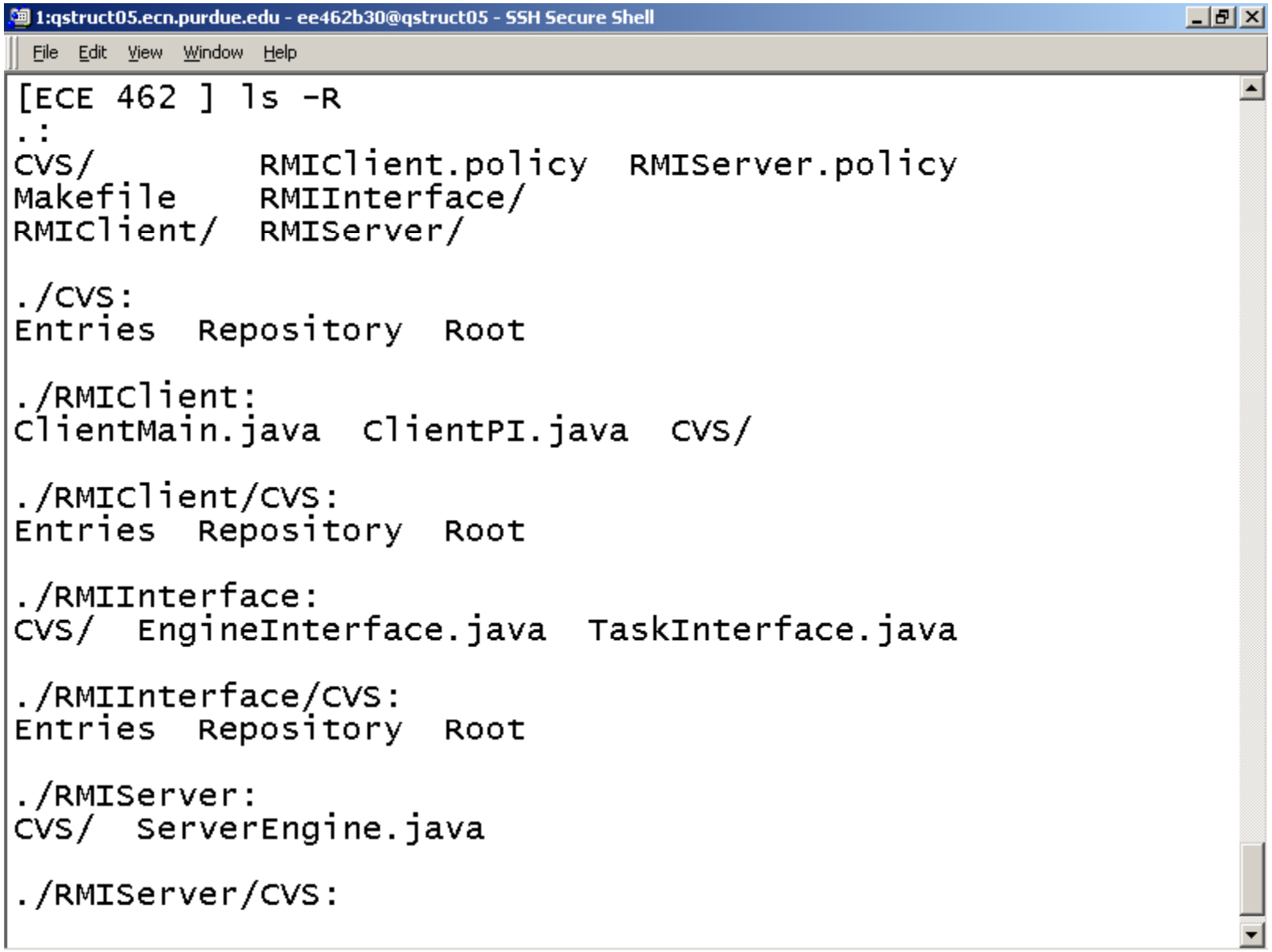
# RMI Architecture

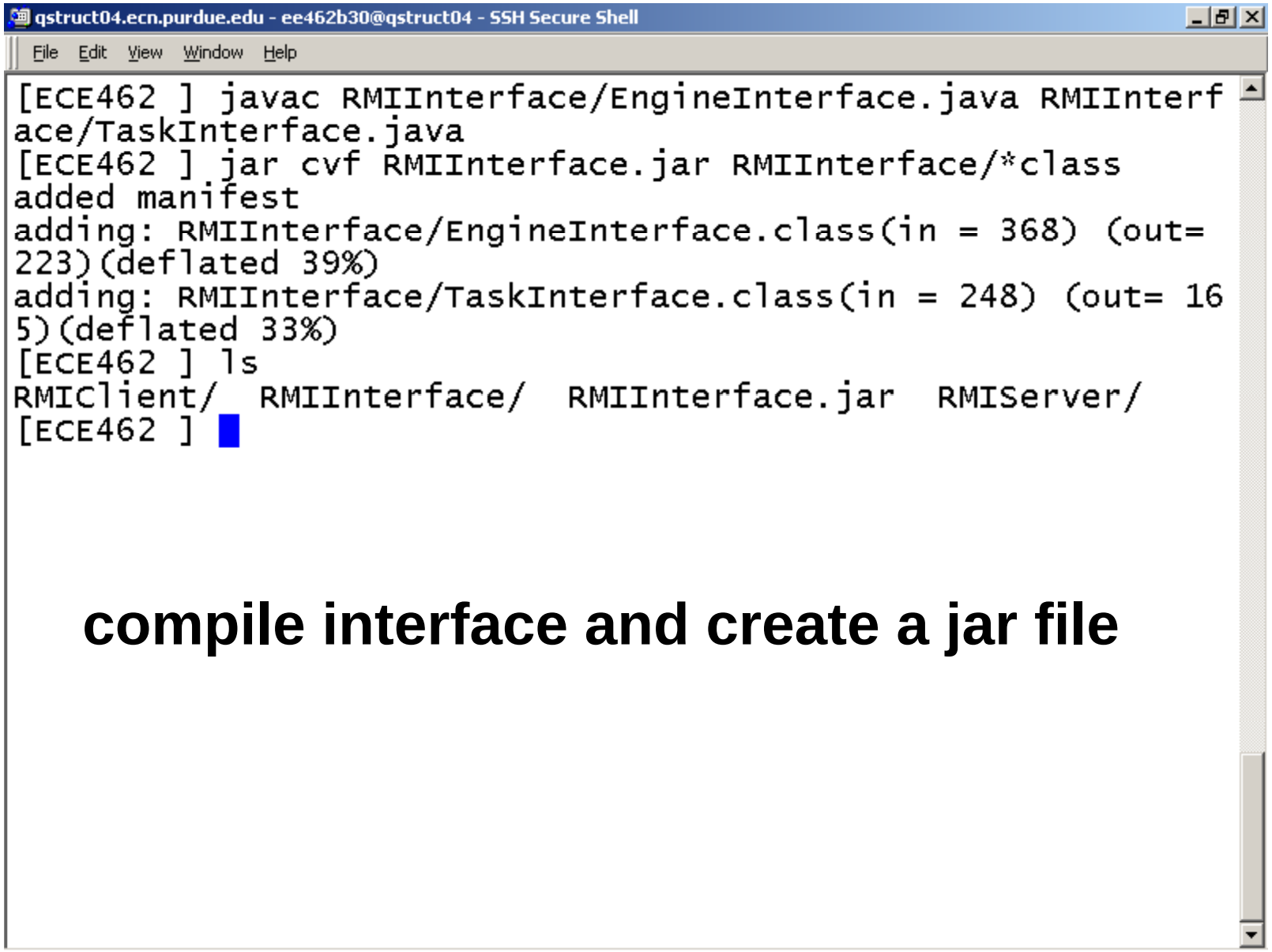
- client-server model
- transmit objects to remote Java virtual machine



# Compute Engine Example

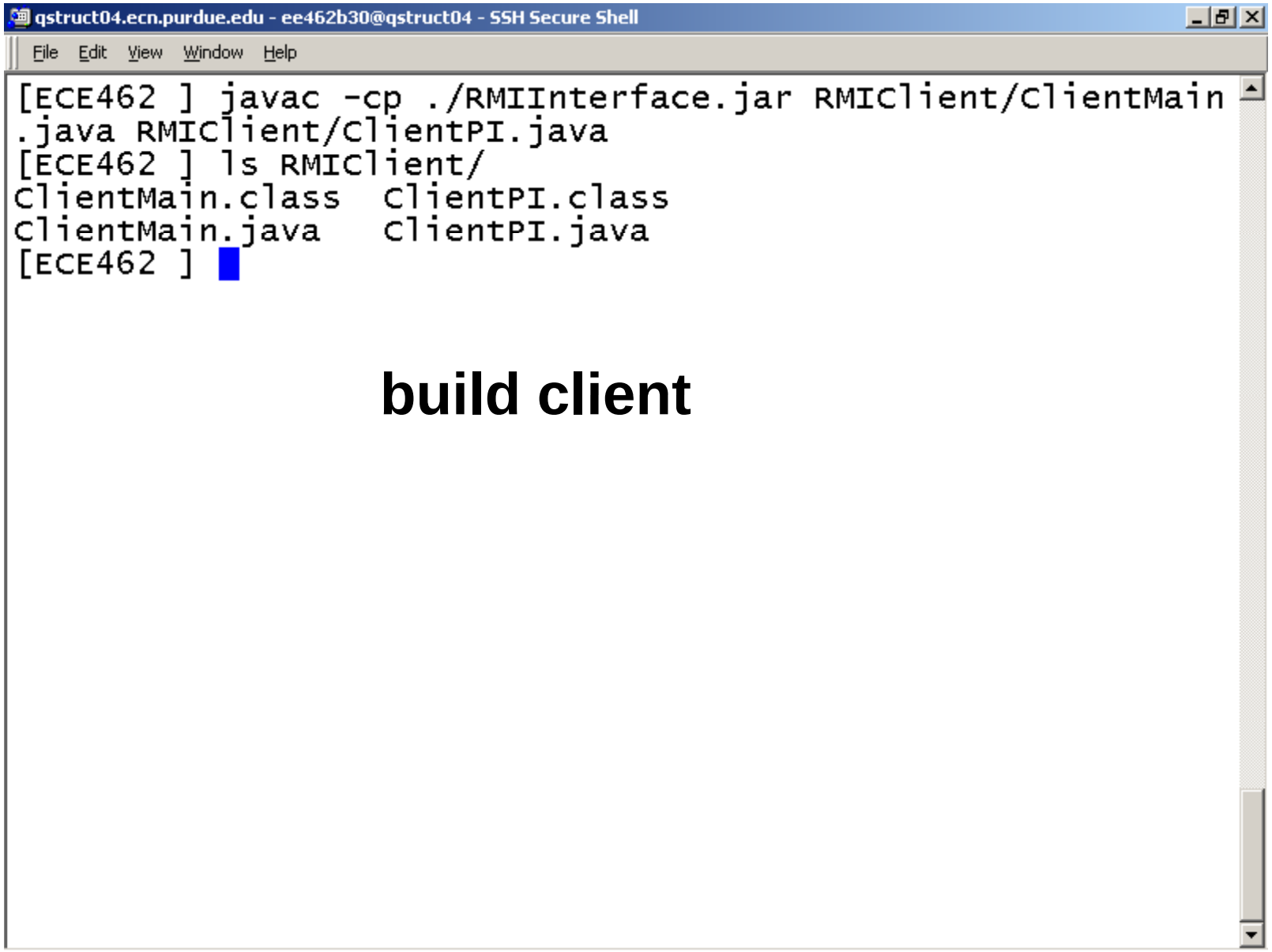
# Demonstration


1:qstruct05.ecn.purdue.edu - ee462b30@qstruct05 - SSH Secure Shell
File Edit View Window Help
[ECE 462 ] ls -R
.:
CVS/ RMIClient.policy RMIServer.policy
Makefile RMIInterface/
RMIClient/ RMIServer/
./CVS:
Entries Repository Root
./RMIClient:
ClientMain.java ClientPI.java CVS/
./RMIClient/CVS:
Entries Repository Root
./RMIInterface:
CVS/ EngineInterface.java TaskInterface.java
./RMIInterface/CVS:
Entries Repository Root
./RMIServer:
CVS/ ServerEngine.java
./RMIServer/CVS:

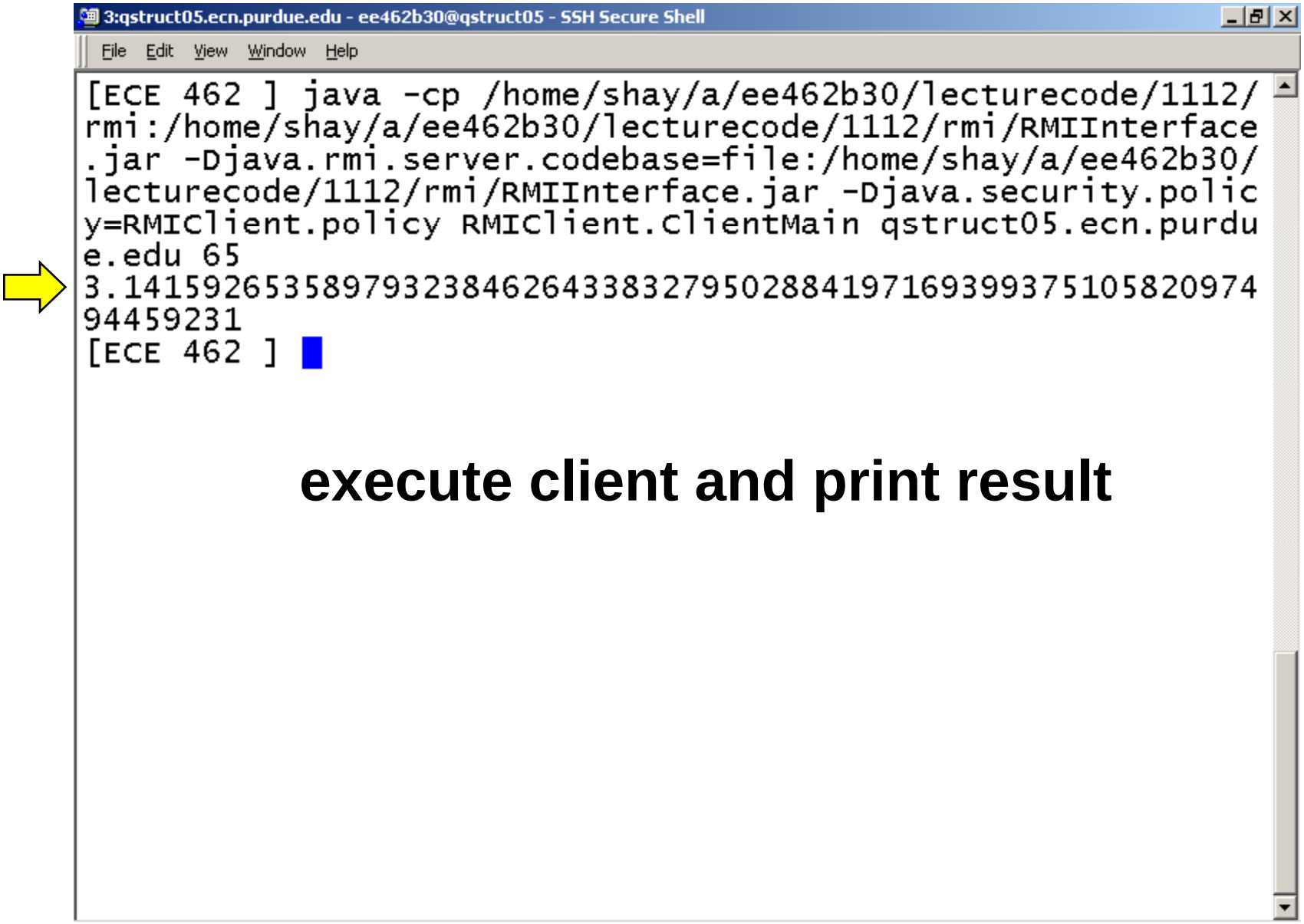


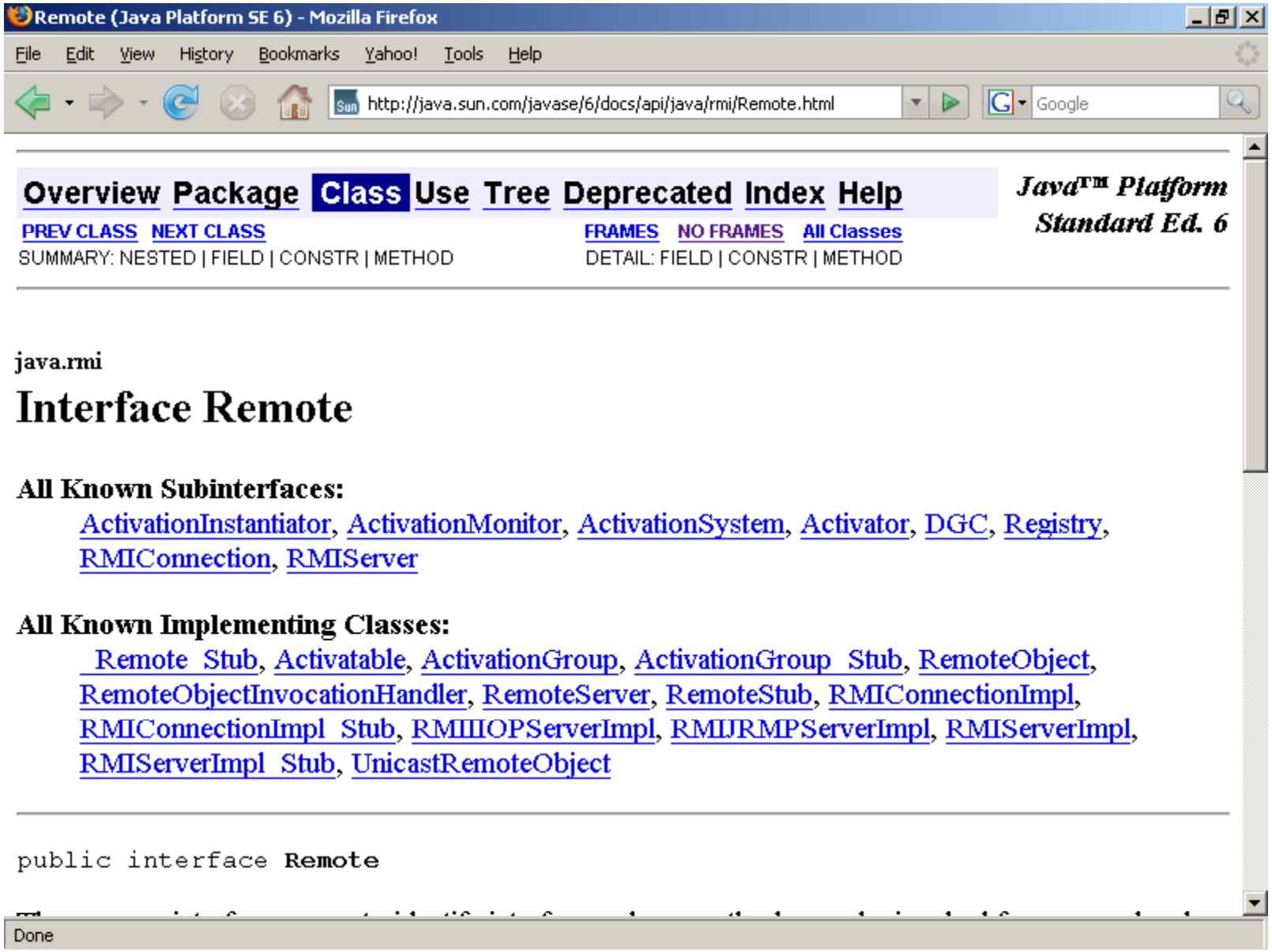

**compile interface and create a jar file**


**build server**


qstruct04.ecn.purdue.edu - ee462b30@qstruct04 - SSH Secure Shell
File Edit View Window Help
[ECE462 ] javac -cp ./RMIInterface.jar RMIClient/ClientMain
.java RMIClient/ClientPI.java
[ECE462 ] ls RMIClient/
ClientMain.class ClientPI.class
ClientMain.java ClientPI.java
[ECE462 ]
build client


**start server**

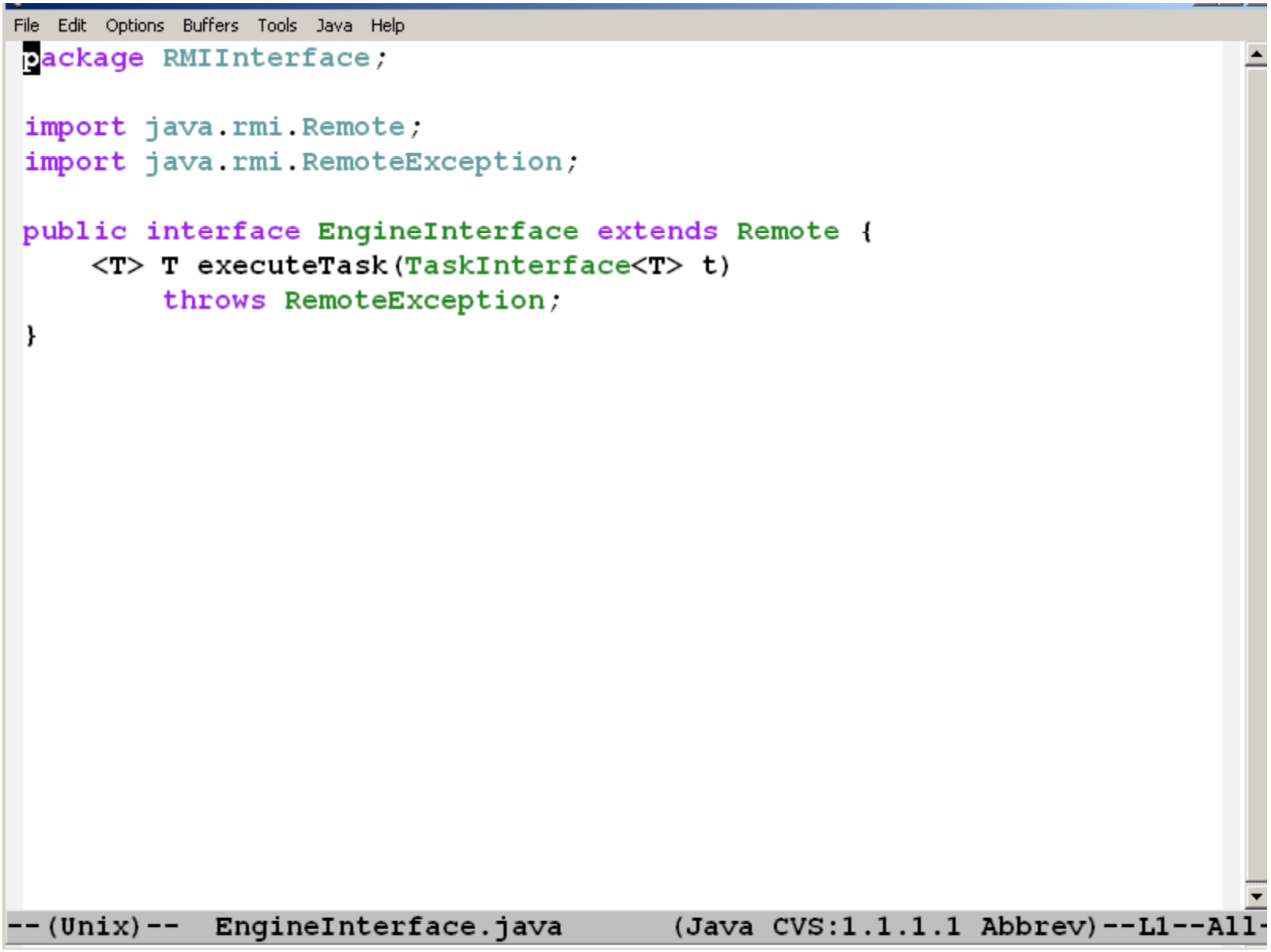

3:qstruct05.ecn.purdue.edu - ee462b30@qstruct05 - SSH Secure Shell
File Edit View Window Help
[ECE 462 ] java -cp /home/shay/a/ee462b30/lecturecode/1112/rmi:/home/shay/a/ee462b30/lecturecode/1112/rmi/RMIInterface.jar -Djava.rmi.server.codebase=file:/home/shay/a/ee462b30/lecturecode/1112/rmi/RMIInterface.jar -Djava.security.policy=RMIClient.policy RMIClient.ClientMain qstruct05.ecn.purdue.edu 65
3.14159265358979323846264338327950288419716939937510582097494459231
[ECE 462 ]
execute client and print result

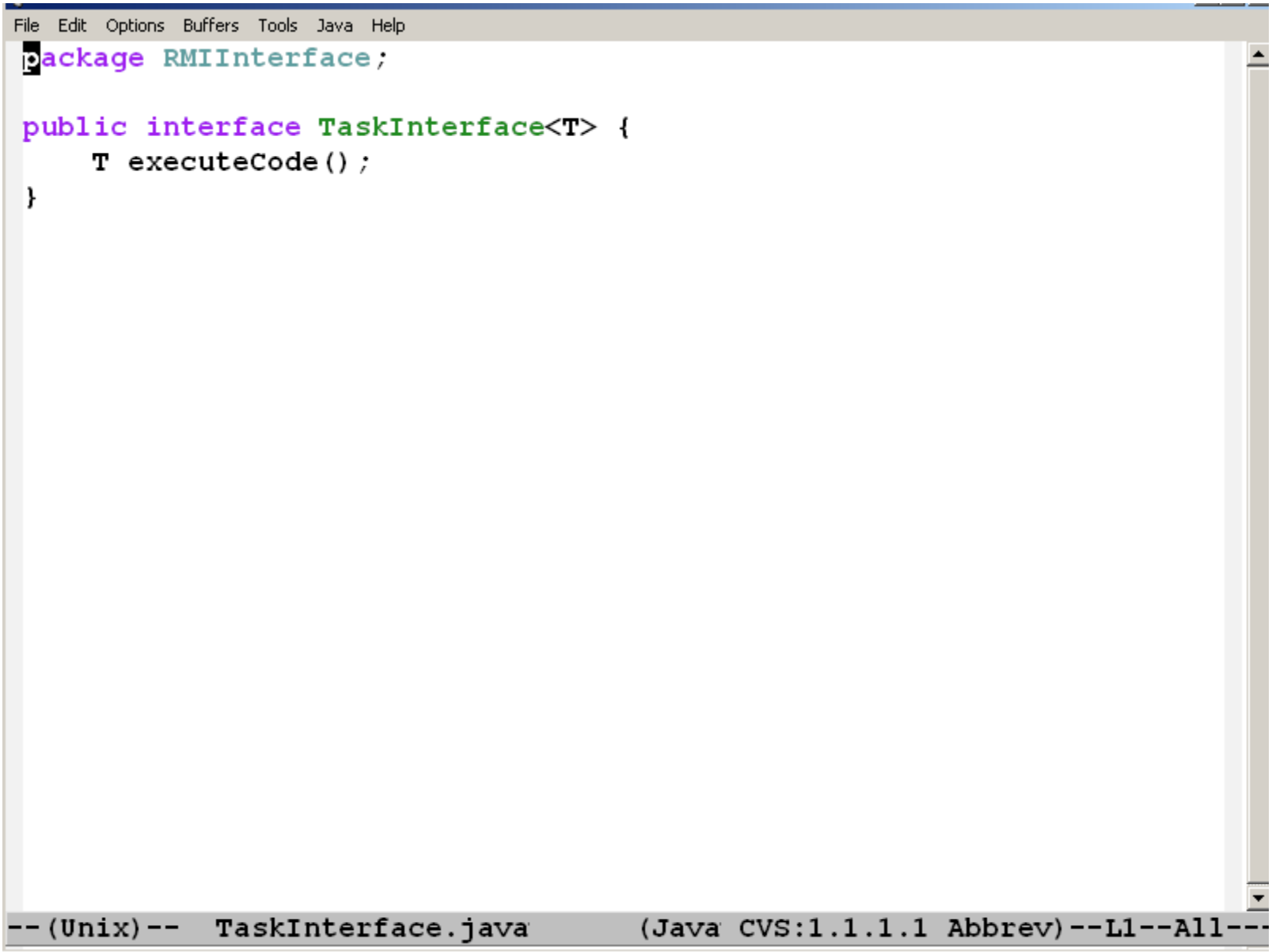

Remote (Java Platform SE 6) - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/javase/6/docs/api/java/rmi/Remote.html
Google
Overview Package Class Use Tree Deprecated Index Help
Java™ Platform Standard Ed. 6
PREV CLASS NEXT CLASS
FRAMES NO FRAMES All Classes
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD
java.rmi
Interface Remote
All Known Subinterfaces:
ActivationInstantiator, ActivationMonitor, ActivationSystem, Activator, DGC, Registry, RMIConnection, RMIServer
All Known Implementing Classes:
Remote_Stub, Activatable, ActivationGroup, ActivationGroup_Stub, RemoteObject, RemoteObjectInvocationHandler, RemoteServer, RemoteStub, RMIConnectionImpl, RMIConnectionImpl_Stub, RMIIIOPServerImpl, RMIJRMPServerImpl, RMIServerImpl, RMIServerImpl_Stub, UnicastRemoteObject
public interface Remote
Done

# Source Code

# Name Mapping

| | Sun's Tutorial | This Example |
|---|---|---|
| package | compute | RMIInterface |
| package | client | RMIClient |
| package | engine | RMIServer |
| interface | Compute | EngineInterface |
| interface | Task | TaskInterface |
| class | ComputeEngine implements Compute | **Server**Engine implements EngineInterface |
| class | ComputePi | **Client**Main |
| class | Pi implements Task <BigDecimal>, Serializable | **Client**PI |

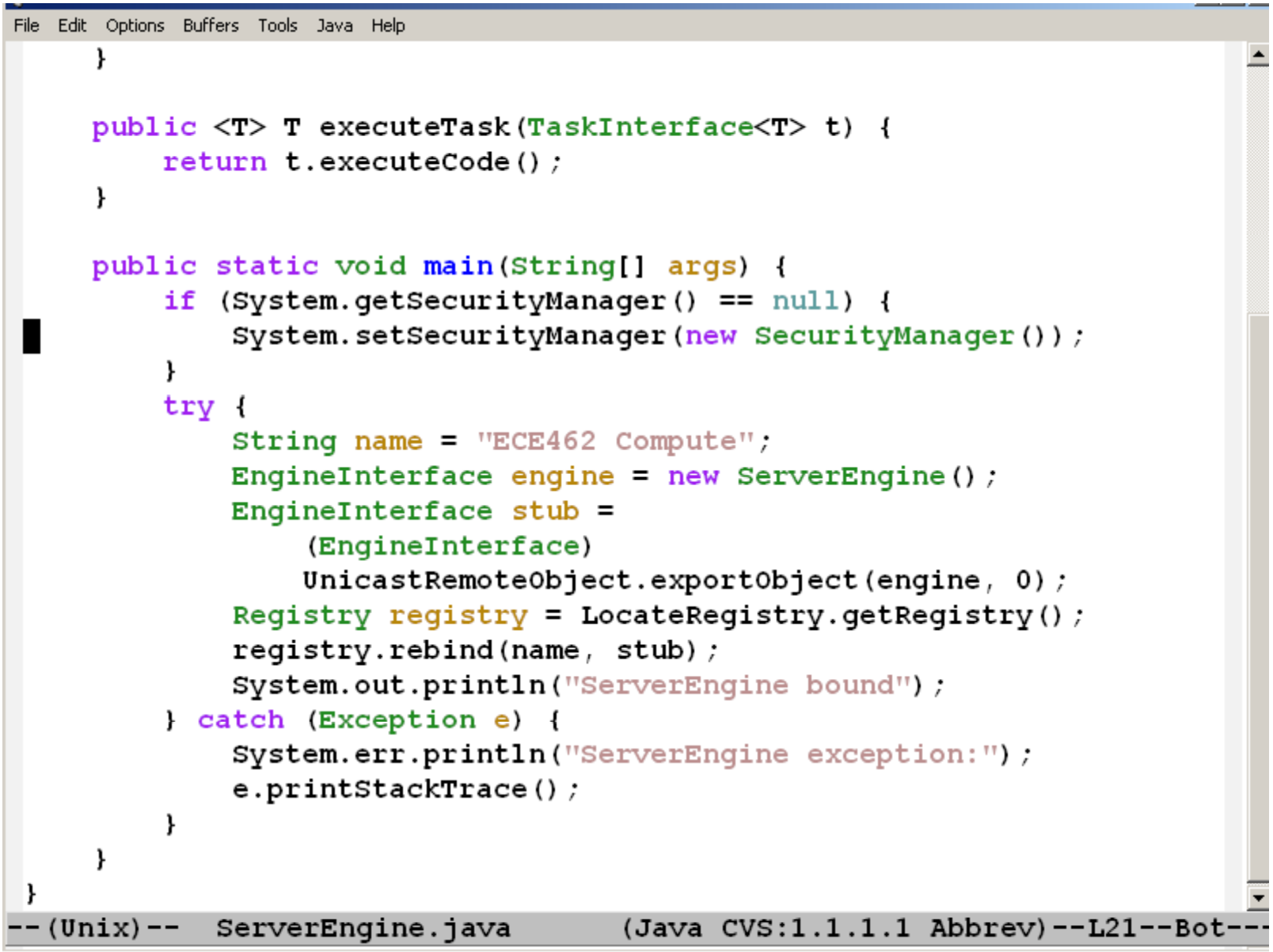

File Edit Options Buffers Tools Java Help
package RMIInterface;
import java.rmi.Remote;
import java.rmi.RemoteException;
public interface EngineInterface extends Remote {
<T> T executeTask(TaskInterface<T> t)
throws RemoteException;
}
--(Unix)-- EngineInterface.java (Java CVS:1.1.1.1 Abbrev)--L1--All-


File Edit Options Buffers Tools Java Help
package RMIInterface;
public interface TaskInterface<T> {
T executeCode();
}
--(Unix)-- TaskInterface.java (Java CVS:1.1.1.1 Abbrev)--L1--All---

File Edit Options Buffers Tools Java Help

```
package RMIServer;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
import RMIInterface.EngineInterface;
import RMIInterface.TaskInterface;

public class ServerEngine implements EngineInterface {

    public ServerEngine() {
        super();
    }

    public <T> T executeTask(TaskInterface<T> t) {
        return t.executeCode();
    }

    public static void main(String[] args) {
        if (System.getSecurityManager() == null) {
            System.setSecurityManager(new SecurityManager());
        }
        try {
            String name = "ECE462 Compute";
            EngineInterface engine = new ServerEngine();
```

--(Unix)-- ServerEngine.java (Java CVS:1.1.1.1 Abbrev)--L1--Top----


File Edit Options Buffers Tools Java Help
}
public <T> T executeTask(TaskInterface<T> t) {
return t.executeCode();
}
public static void main(String[] args) {
if (System.getSecurityManager() == null) {
System.setSecurityManager(new SecurityManager());
}
try {
String name = "ECE462 Compute";
EngineInterface engine = new ServerEngine();
EngineInterface stub =
(EngineInterface)
UnicastRemoteObject.exportObject(engine, 0);
Registry registry = LocateRegistry.getRegistry();
registry.rebind(name, stub);
System.out.println("ServerEngine bound");
} catch (Exception e) {
System.err.println("ServerEngine exception:");
e.printStackTrace();
}
}
}
--(Unix)-- ServerEngine.java (Java CVS:1.1.1.1 Abbrev)--L21--Bot---

File Edit Options Buffers Tools Java Help

```
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.math.BigDecimal;
import java.net.InetAddress;
import RMIInterface.EngineInterface;

public class ClientMain {
    public static void main(String args[]) {
        if (System.getSecurityManager() == null) {
            System.setSecurityManager(new SecurityManager());
        }
        try {
            String name = "ECE462 Compute";
            Registry registry = LocateRegistry.getRegistry(args[0]);
            EngineInterface comp = (EngineInterface)
                registry.lookup(name);
            ClientPI task = new ClientPI(Integer.parseInt(args[1]));
            BigDecimal pi = comp.executeTask(task);
            System.out.println(pi);
        } catch (Exception e) {
            System.err.println("ClientMain exception:");
            e.printStackTrace();
        }
    }
}
```

--(Unix)--  ClientMain.java  (Java CVS:1.1.1.1 Abbrev)--L18--Bot-----

File Edit Options Buffers Tools Java Help

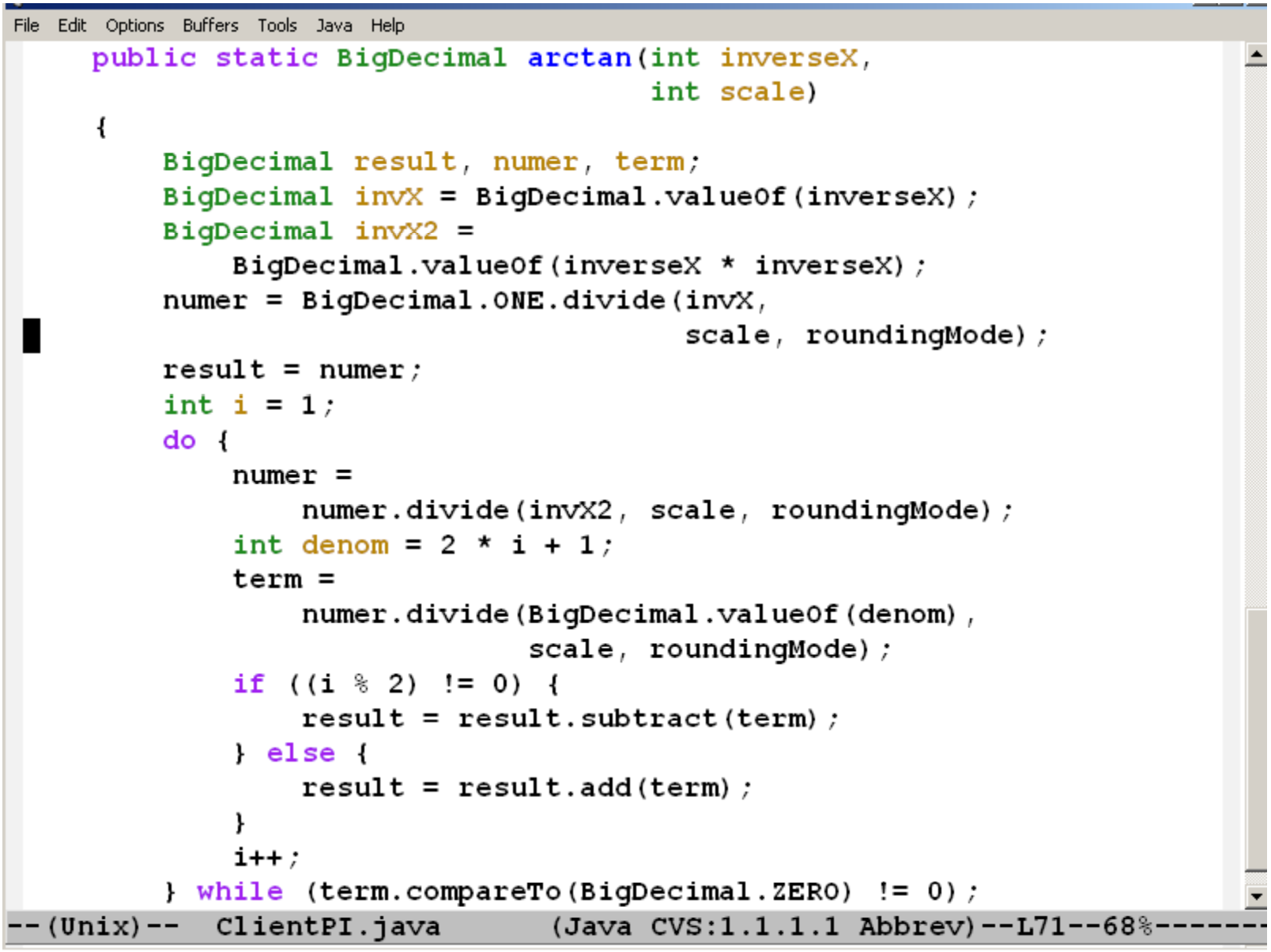
```
package RMIClient;

import RMIInterface.TaskInterface;
import java.io.Serializable;
import java.math.BigDecimal;
import java.net.InetAddress;
public class ClientPI implements TaskInterface<BigDecimal>,
                                 Serializable {

    private static final long serialVersionUID = 227L;
    /** constants used in pi computation */
    private static final BigDecimal FOUR =
        BigDecimal.valueOf(4);
    /** rounding mode to use during pi computation */
    private static final int roundingMode =
        BigDecimal.ROUND_HALF_EVEN;
    /** digits of precision after the decimal point */
    private final int digits;
    /**
     * Construct a task to calculate pi to the specified
     * precision.
     */
    public ClientPI(int d) {
        digits = d;
    }
```

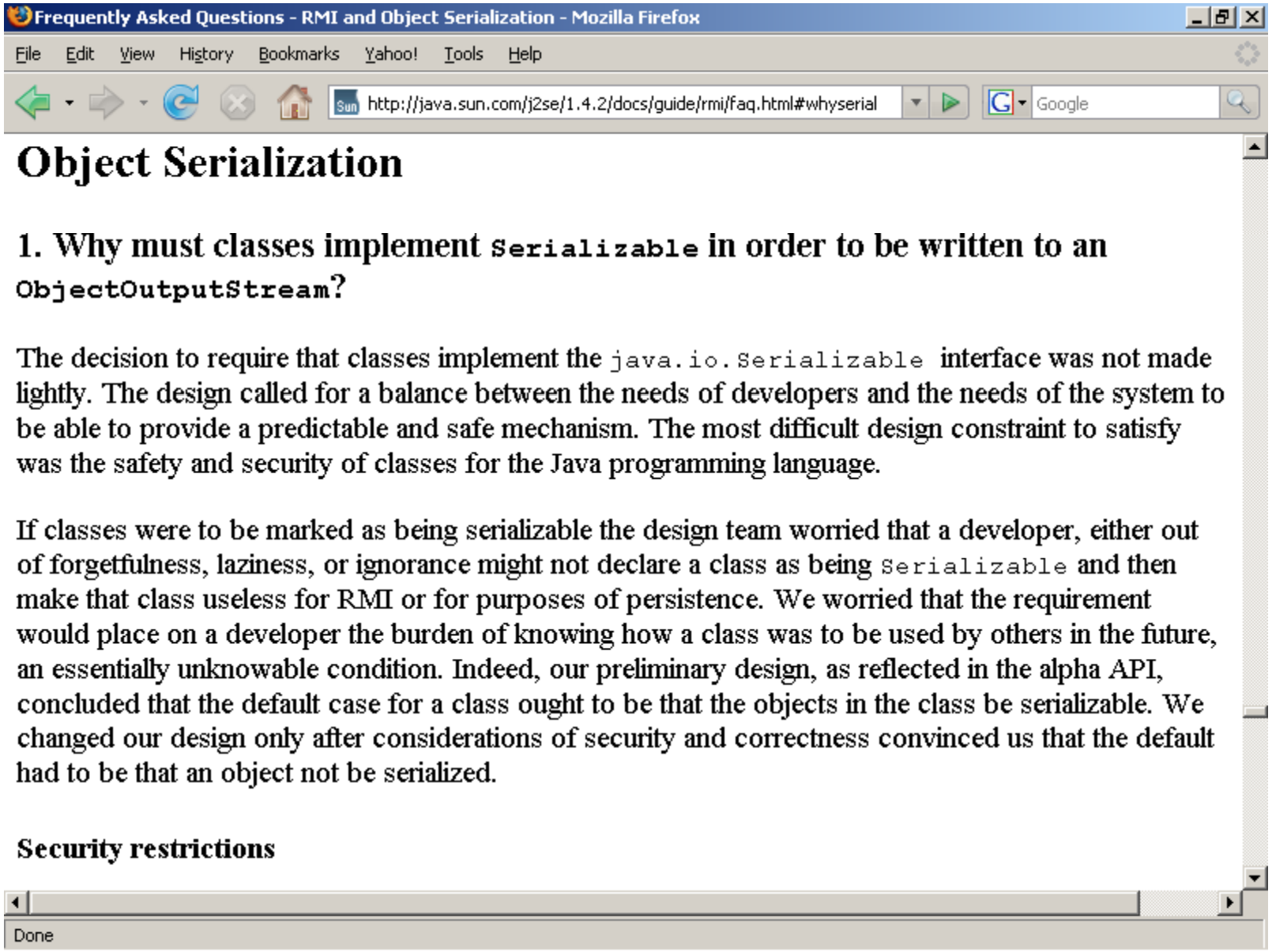
--(Unix)-- ClientPI.java (Java CVS:1.1.1.1 Abbrev)--L7--Top--------

File Edit Options Buffers Tools Java Help

```
    public BigDecimal executeCode() {
        return computePi(digits);
    }

    /**
     * Compute the value of pi to the specified number of
     * digits after the decimal point.  The value is
     * computed using Machin's formula:
     *
     *          pi/4 = 4*arctan(1/5) - arctan(1/239)
     *
     * and a power series expansion of arctan(x) to
     * sufficient precision.
     */
    public BigDecimal computePi(int digits) {
        int scale = digits + 5;
        BigDecimal arctan1_5 = arctan(5, scale);
        BigDecimal arctan1_239 = arctan(239, scale);
        BigDecimal pi = arctan1_5.multiply(FOUR).subtract(
                                    arctan1_239).multiply(FOUR);
        return pi.setScale(digits,
                           BigDecimal.ROUND_HALF_UP);
    }
    /**
     * Compute the value, in radians, of the arctangent of
```

--(Unix)--  ClientPI.java      (Java CVS:1.1.1.1 Abbrev)--L49--28%-------

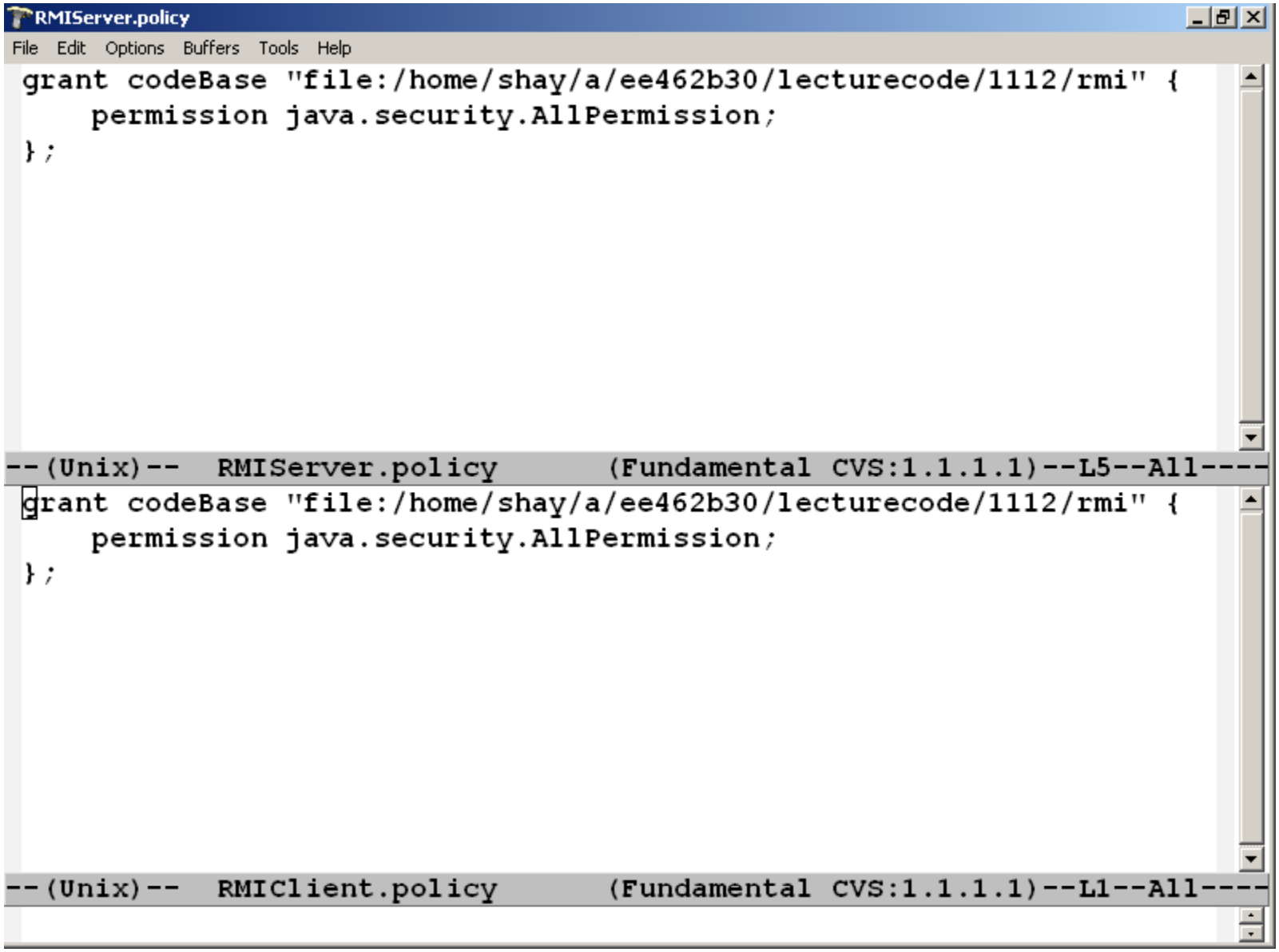

File Edit Options Buffers Tools Java Help
public static BigDecimal arctan(int inverseX,
int scale)
{
BigDecimal result, numer, term;
BigDecimal invX = BigDecimal.valueOf(inverseX);
BigDecimal invX2 =
BigDecimal.valueOf(inverseX * inverseX);
numer = BigDecimal.ONE.divide(invX,
scale, roundingMode);
result = numer;
int i = 1;
do {
numer =
numer.divide(invX2, scale, roundingMode);
int denom = 2 * i + 1;
term =
numer.divide(BigDecimal.valueOf(denom),
scale, roundingMode);
if ((i % 2) != 0) {
result = result.subtract(term);
} else {
result = result.add(term);
}
i++;
} while (term.compareTo(BigDecimal.ZERO) != 0);
--(Unix)-- ClientPI.java (Java CVS:1.1.1.1 Abbrev)--L71--68%-------


Frequently Asked Questions - RMI and Object Serialization - Mozilla Firefox
File Edit View History Bookmarks Yahoo! Tools Help
http://java.sun.com/j2se/1.4.2/docs/guide/rmi/faq.html#whyserial
Google
Object Serialization
1. Why must classes implement Serializable in order to be written to an ObjectOutputStream?
The decision to require that classes implement the java.io.Serializable interface was not made lightly. The design called for a balance between the needs of developers and the needs of the system to be able to provide a predictable and safe mechanism. The most difficult design constraint to satisfy was the safety and security of classes for the Java programming language.
If classes were to be marked as being serializable the design team worried that a developer, either out of forgetfulness, laziness, or ignorance might not declare a class as being Serializable and then make that class useless for RMI or for purposes of persistence. We worried that the requirement would place on a developer the burden of knowing how a class was to be used by others in the future, an essentially unknowable condition. Indeed, our preliminary design, as reflected in the alpha API, concluded that the default case for a class ought to be that the objects in the class be serializable. We changed our design only after considerations of security and correctness convinced us that the default had to be that an object not be serialized.
Security restrictions
Done

Makefile<2>

File Edit Options Buffers Tools Makefile Help

```
INT = RMIInterface
SER = RMIServer
CLI = RMIClient

build:
        javac $(INT)/EngineInterface.java $(INT)/TaskInterface.java
        jar cvf RMIInterface.jar $(INT)/*class
        javac -cp ./RMIInterface.jar $(SER)/ServerEngine.java
        javac -cp ./RMIInterface.jar $(CLI)/ClientMain.java $(CLI)/ClientPI.jav

clean:
        rm -f RMIInterface.jar
        rm -f $(INT)/*.class
        rm -f $(SER)/*.class
        rm -f $(CLI)/*.class

start:
        rmiregistry &
        java -cp /home/shay/a/ee462b30/lecturecode/1112/rmi:/home/shay/a/ee462b
$12/rmi/RMIInterface.jar -Djava.rmi.server.codebase=file:/home/shay/a/ee462b30/l
$mi/RMIInterface.jar -Djava.rmi.server.hostname=qstruct05.ecn.purdue.edu -Djava.
$MIServer.policy RMIServer.ServerEngine

test:
        java -cp /home/shay/a/ee462b30/lecturecode/1112/rmi:/home/shay/a/ee462b
$12/rmi/RMIInterface.jar -Djava.rmi.server.codebase=file:/home/shay/a/ee462b30/l
$mi/RMIInterface.jar -Djava.security.policy=RMIClient.policy RMIClient.ClientMai
$urdue.edu 65
```


RMIServer.policy
File Edit Options Buffers Tools Help
grant codeBase "file:/home/shay/a/ee462b30/lecturecode/1112/rmi" {
permission java.security.AllPermission;
};
--(Unix)-- RMIServer.policy (Fundamental CVS:1.1.1.1)--L5--All----
grant codeBase "file:/home/shay/a/ee462b30/lecturecode/1112/rmi" {
permission java.security.AllPermission;
};
--(Unix)-- RMIClient.policy (Fundamental CVS:1.1.1.1)--L1--All----