1. Modify the display of an airplane on the top row of the grid. If you look at an airplane on that row you cannot tell what the letter is. Fix it.

In class ATCUI\_impl.java:

//Change text\_gap to -10 so that the plane ID shows up at the top of the GUI

**protected** **int** text\_height, text\_gap = -10;

1. When an airplane that is ready to take off from an airport is listed in the information column on the right it doesn’t indicate which airport the airplane is at. Add another column that will indicate which airport the plane is at. Be careful – there are already 3 columns in use. You’ll have to work this one in.

In the Class ATCUI\_impl.java, method getPlaneInfoText():

/\*

\* Check to see if the airplane is waiting at an airport

\* then iterate through the airport array to determine which airport

\* the plane is located at. Once a match is found add the airport to the string of

\* information displayed during the game.

\*/

**if**(p.waiting\_flag == **true**){

Airfield[] a = atc\_obj.data.airfields;

**for** (**int** i = 0; i < a.length; i++){

**if**(p.pos.equals(a[i].pos)){

rs += "@" + a[i].getName() + " ";

**break**;

}

}

}

1. Change all dialog windows to display an “airplane” icon instead of a “coffee cup” icon. Pick out your own airplane icon image that works well.

I changed this to a gif of a plane crashing for fun, I hope that’s ok.

In Class ATCUI\_impl.java:

**public** **void** gameOver( String gameOverMessage )

{

/\*Message box to inform user that the game has ended and what happened

\* to end the game and display an airplane.

\*/

**final** ImageIcon icon = **new** ImageIcon("images/planecrash.gif");

JOptionPane.*showMessageDialog*(**null**, gameOverMessage,

"GAMEOVER", JOptionPane.*INFORMATION\_MESSAGE*,icon);

1. Currently airplanes crash if they are within 1000 feet of each other. Make them crash if they are only at the same altitude as the other plane.

In Class ATCData.java and method isDead():

/\*

\* Changed "<= 1" to "== 0" of the altitude difference parameter

\* of the following if statement so that plane's will only crash

\* when flying at the same altitude.

\*/

**if**( Math.*abs*( planes[id].alt - planes[id2].alt ) == 0 &&

Math.*abs*( planes[id].pos.x - planes[id2].pos.x ) <= 1 &&

Math.*abs*( planes[id].pos.y - planes[id2].pos.y ) <= 1 )

**throw** **new** ATCGameOverException(

"Planes " + planes[id].getIdChar() + " and "

+ planes[id2].getIdChar() + " crashed." );

1. Add a pause button on the screen that will pause the game until you come back.

In the Class ATCUI\_inpl.java: class RadarPane:

JButton pauseButton;

/\*

\* add pause button to the display

\*/

pauseButton = **new** JButton( "Pause" );

pauseButton.setActionCommand("Pause");

pauseButton.addActionListener( **this** );

pauseButton.setFocusable( **false** );

In the Class ATCData.java:

//Attribute to handle pausing and unpausing the game

**protected** **boolean** paused = **false**;

In method tick():

**if**(paused != **true**){/\*Only tick while game is not paused\*/

In Class ATCInputhandler\_impl.java, Method processActionCommand():

/\* When the pause button is clicked, call the pause() method in ATCData to stop the timer and display a dialog box. When the user clicks ok the box will close, then

\* call the unpause() method in ATCData to resume the timer/game.

\*/

**else** **if**("Pause".equals(command)){

atc\_obj.data.paused = **true**;

**final** ImageIcon icon = **neW**ImageIcon("images/planeturn.gif");

JOptionPane.*showMessageDialog*(**null**, "Game Paused\n\nOk to Continue",

"GAME PAUSED", JOptionPane.*INFORMATION\_MESSAGE*,icon);

atc\_obj.data.paused = **false**;

}

1. Currently the highest altitude is 9000 feet. Raise that to 11000 feet. All airplanes come into the grid at 7000 feet and they must exit the grid at 11000 feet.

In Class ATCInputhandler\_impl.java method parse():

**case** 3: //altitude

**int** new\_alt;

**switch**(c){

**case** 't': //use letter t for ten

new\_alt = 10;

**break**;

**case** 'e'://use letter e for eleven

new\_alt = 11;

**break**;

**default**:

**try** { new\_alt = Integer.*parseInt*(Character.*toString*(c)); }

**catch**(Exception e) { **return** **false**; }

}

//change max altitude to 11000

**if**( new\_alt < 0 || new\_alt > 11 )

**return** **false**;

full\_cmd\_str += Integer.*toString*(new\_alt) + "000 feet ";

**protected** Exit() { **super**(); alt = 11; }

**public** Exit( Exit ao )

{

**super**( (StaticObj)ao );

alt = 7;

exit\_alt = 11;//change exit altitude to 11

}

**public** Exit( Position p, Direction d )

{

**super**( p, d );

alt = 7;

exit\_alt = 11;//change exit altitude to 11

exit\_dir = **null**; // We don't care which direction you are exiting, yet.

}