[illegible] – 80

[illegible] 90

INHERIT SPACE

Joseph C. Giarratano, Ph.D.

University of Houston Clear Lake

JSC Mail Code BT2

August 8, 1996

Kenneth C. Jenks
Information Technology Office
Information Systems Division
Business and Information Systems Directorate

Joseph C. Giarratano

Joseph C. Giarratano

Kenneth C. Jenks

Kenneth C. Jenks

INHERIT SPACE

Final Report
NASA/ASEE Summer Faculty Program-1996
Johnson Space Center

Prepared by:	Joseph C. Giarratano, Ph.D.
Academic Rank:	Associate Professor of Computer Science
University and Department:	University of Houston Clear Lake Computer Science Dept. 2700 Bay Area Blvd. Houston, TX 77058

NASA/JSC

Directorate:	Business and Information Systems
Division:	Information Systems
Branch:	Information Technology
JSC Colleague:	Kenneth C. Jenks
Date Submitted:	August 8, 1996
Contract Number:	NAG 9-867

ABSTRACT

The objective of the proposed research was to begin development of a unique educational tool targeted at educating and inspiring young people 12-16 years old about NASA and the Space Program. Since these young people are the future engineers, scientists and space pioneers, the nurturing of their enthusiasm and interest is of critical importance to the Nation.

This summer the basic infrastructure of the tool was developed in the context of an educational game paradigm. The game paradigm has achieved remarkable success in maintaining the interest of young people in a self-paced, student-directed learning environment. This type of environment encourages student exploration and curiosity which are exactly the traits that future space pioneers need to develop to prepare for the unexpected.

The Inherit Space Educational Tool is an open-ended learning environment consisting of a finite-state machine classic adventure game paradigm. As the young person explores this world, different obstacles must be overcome. Rewards will be offered such as using the Flight Simulator to fly around and explore Titan. This simulator was modeled on conventional Earth flight simulators but has been considerably enhanced to add texture mapping of Titan's atmosphere utilizing the latest information from the NASA Galileo Space Probe. Additional scenery was added to provide color VGA graphics of a futuristic research station on Titan as well as an interesting story to keep the youngster's attention.

This summer the game infrastructure has been developed as well as the Titan Flight Simulator. A number of other enhancements are planned.

INTRODUCTION

Significance of Project
In addition to its primary mission of Space Research, NASA has a number of other goals such as Technology Transfer and educating the public about Space. The education of our Nation's youth is particularly important fewer young people are going into science and engineering, yet these disciplines are the foundations of the Space Program. Besides education, there is a critical need to *inspire* the public about the Space Program to keep the dream alive of expanding beyond the boundaries of Earth.

To carry out this mission of educating and inspiring youth about Space and NASA, the strategy chosen was to develop a game called *Inherit Space* targeted at 12-16 year olds. The game paradigm was selected to make the educational component about living in Space interesting while also being fun to play.

The game design is based on popular adventure games in which the user has a goal and must overcome a number of obstacles. The user must explore new environments and challenges in Inherit Space, thus mirroring the challenges faced in Space Exploration. The great advantage of an adventure game is that it allows the user an open-ended exploration of the environment rather than the linear design of a comic book. The disadvantage of an adventure game is that it requires a great deal of programming and creativity on the part of the designers.

Statement of Project
The initial tasks undertaken this summer were to
- Create the infrastructure of a multimedia adventure game involving graphics, video, and sound.
- Implement a basic exploration scenario involving multiple rooms and obstacles
- Implement a flight simulator for Titan, a moon of Saturn.

In order to make it convenient for the target audience of 12-16 year olds to use the game, a graphical user interface (GUI) was designed and implemented to provide seamless transitions between the adventure game and application programs such as the Flight Simulator. The GUI is designed for Windows and Mouse support, and a joystick for operation of the Flight Simulator. Graphic images of the different rooms are shown the user and a musical score is played which changes depending on the room the user is in.

METHODOLOGY

In order to expedite the software development, commercial off the shelf software was used whenever possible (Michael Radke and Chris Lampton, *Build Your Own Flight Sim in C++*). However this was still not trivial as the original flight simulator is 6,000 lines of C++ code which required substantial modification for Titan. In particular the background scenery was modified from the simple blue sky of Earth to provide colors and texture mapping for realistic views of the Titanian atmosphere based on data from the NASA Galileo Space Probe. In addition, the scenery had to be altered to display buildings and other objects of the game plot.

Initially Java was tried as the development language. However since Java does not support video clips which are planned for the game, and is also too slow for real-time response, it was decided to switch to an alternative development tool called Delphi.

IMPLEMENTATION

Finite-State Machine Adventure Game

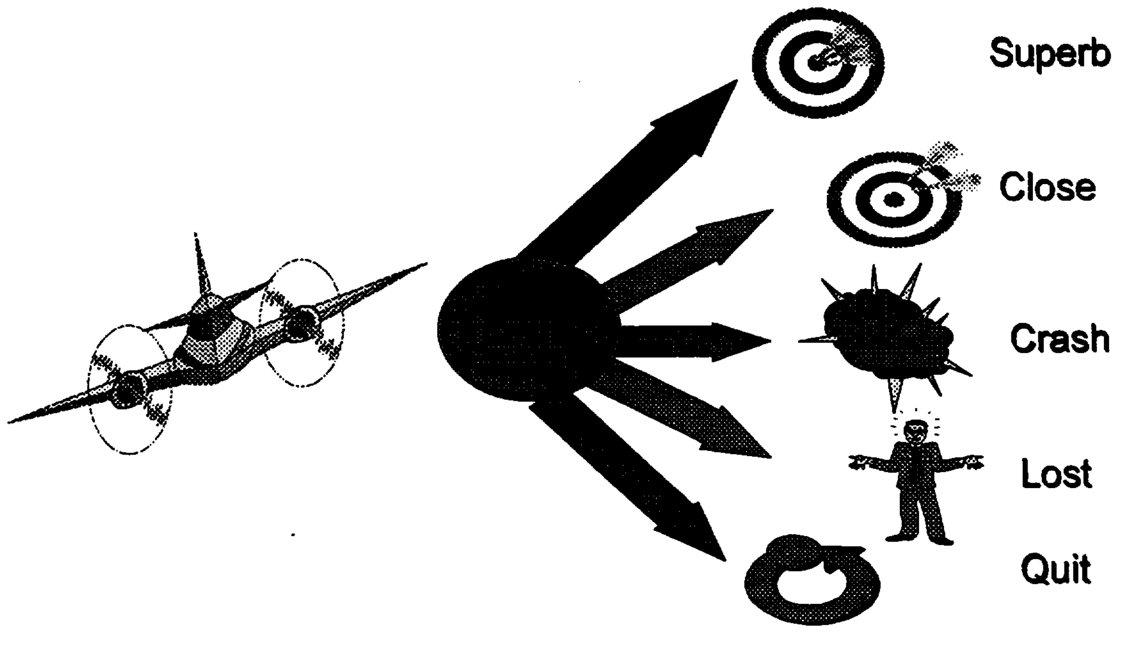
A finite-state machine was implemented using Delphi for the adventure game infrastructure. The finite-state machine is straightforward to implement easily extensible as further rooms are added to the game. It is also very easy to call other code such as the flight simulator depending on the state of the finite-state machine, i.e., what room the user is in.

Graphic images are shown to the user and mouse-sensitive buttons are displayed to allow the user to explore other rooms or take other actions. This feature is particularly convenient for the 12-16 year olds as it minimizes typing of input. It also keeps up the pace of the game rather than slowing down while the user types input.

Flight Simulator

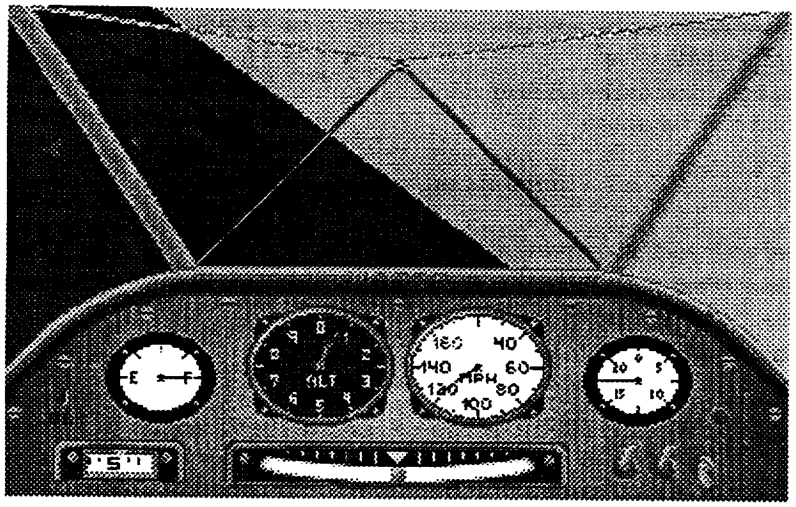
The Flight Simulator is game designed to be called from the main adventure game. This game within a game is called a *gamelet*. The Flight Simulator is designed as a fun educational component. It teaches the user what conditions on Titan are like through flying over the surface and various instrument readouts. There is a good deal of skill required by the user for successful flying. Even if the user has experience with Earth simulators, the Titan Flight Simulator is substantially different and challenging.
The following modifications to the original flight simulator were implemented.

- Initialize flight in the air to match the game plot
- Improve flying controls for Titan
- Customize scenery for Titan
- Provide interface with adventure game GUI

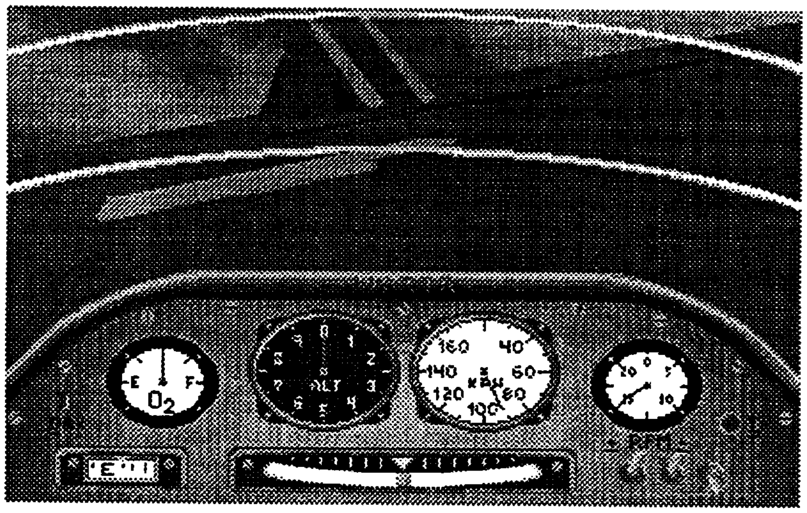


A number of 3-D interactive game programming techniques and algorithms were required to properly implement the Flight Simulator:

- Object representation
- Keyboard, mouse, and joystick input
- Flight modeling
- VGA color system, texture and color mapping
- Viewing transformations
- Integrate with the main program
- Flight Initialization in mid-air required proper initialization of the State Vector
- Improved Flight Controls since the original code provided poor flight control
- Sensitivity Variable to provide the same flying characteristics on different machines
- Real-time Timer



Before Modification



After Modification

3D VIEW PROCESSING

In addition, a considerable amount of effort was needed to provide realistic 3D modeling of the graphics showing Titan that involved view processing. The following factors were affected.

- Multiple coordinate system transformations
- Polygon sorting
- Hidden surface removal
- Polygon clipping
- Pixel color assignment
- VGA Master Palette
- The horizon
- Dynamic sky and ground
- Different approach needed for 3-D texturing of sky and ground
- Texturing not supported by existing object classes and algorithms
- Bitmaps and the master color palette required modification
- Improving the Scenery
- Scenery Objects - World File
- Objects
- Polygons
  - Vertices
  - Color
- Location
- World Coordinates
- Improving the Scenery

PROBLEMS ENCOUNTERED

Some of the major problems encountered were as follows.

- Determining when an EXE was closed and control returned to launcher program
- Launching flight simulator from the Delphi program
- Delphi executables were too large to fit on a diskette
- Distinguish between DLL and EXE programs
- Change music between modules
- Maintain continuity between modules with same music
- Silence was difficult
- Creating obstacles for the user

CONCLUSIONS

The initial version of game was successfully completed this summer. A number of future improvements are planned. For the Flight Simulator the following enhancements will be done.

- Improve texture mapping for sky and ground.
- Add terrain using fractals.
- Modernize cockpit indicators.
- Improve flight model.
- Add collision detection.
- Add auto-pilot and help.

For the adventure game, a number of additional rooms and gamelets need to be added as the adventure continues. Another feature that will be added is a multimedia Timeline that will show NASA history and technology development from the past to the future involving the game. Multimedia video clips and other educational aids will be used to pique user interest and provide a resource for other games.

APPENDIX-FINITE STATE MACHINE SOFTWARE

```
unit Launch;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
  Forms, Dialogs, StdCtrls, ExtCtrls, IniFiles, MPlayer;

type
  TLauncherForm = class(TForm)
    Timer1: TTimer;
    MediaPlayer1: TMediaPlayer;
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
  private
    { Private declarations }
    Other: Word;
    procedure InitEnvParms;
  public
    { Public declarations }
    EnvParms: TStringList;
  end;

var
  LauncherForm: TLauncherForm;

implementation

{$R *.DFM}
procedure TLauncherForm.InitEnvParms;
var
  p: PChar;
  s: string;
begin
  p:=GetDosEnvironment;
  while p^ <> #0 do begin
    s:=StrPas(p);
    EnvParms.Add(s);
    p:=StrEnd(p)+1;
  end;
end;


procedure TLauncherForm.FormCreate(Sender: TObject);
var
  RoomsIni,StatusIni: TIniFile;
  RtnCode: Integer;
  B : Integer;
  NameOfFirstRoom, OldSong : String;
begin
  RoomsIni :=
TIniFile.Create('d:\delphi\source\launch\rooms.INI');
  NameOfFirstRoom := RoomsIni.ReadString('rooms',
'first','ERROR');
  OldSong := RoomsIni.ReadString('midi',
NameOfFirstRoom,'ERROR');
  RoomsIni.Free;

  StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
  StatusIni.WriteString('location', 'current', NameOfFirstRoom);
  StatusIni.WriteString('location', 'previous',
NameOfFirstRoom);
  StatusIni.WriteString('location', 'oldsong', OldSong);
  StatusIni.WriteBool('items', 'toothpaste', False);
  StatusIni.WriteBool('items', 'food', False);
  StatusIni.Free;

  MediaPlayer1.FileName := OldSong;
  MediaPlayer1.Open;
  MediaPlayer1.Play;

  EnvParms:=TStringList.Create;
  InitEnvParms;
end;

procedure TLauncherForm.FormDestroy(Sender: TObject);
begin
  EnvParms.Free;
end;

  procedure LivingroomForm;
    far; external 'DLiving';

  procedure KitchenForm;
    far; external 'DKitchen';

procedure TLauncherForm.Timer1Timer(Sender: TObject);
var
  StatusIni,RoomsIni, MusicIni : TIniFile;
  Oldsong, CurrentLocation, CurrentLocationExe, CurrentSong
: String;
  cmd: array[0..100] of Char;

begin

  if GetModuleUsage(Other) = 0 then begin
    Timer1.Enabled:=False;

    StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
    CurrentLocation := StatusIni.ReadString('location',
'current','ERROR');
    OldSong := StatusIni.ReadString('location',
'oldsong','ERROR');
    StatusIni.Free;

    if CurrentLocation = 'EXIT' then halt;

    MusicIni :=
TIniFile.Create('d:\delphi\source\launch\rooms.INI');
    CurrentSong := MusicIni.ReadString('midi',
CurrentLocation,'ERROR');
    MusicIni.Free;

    If OldSong <> CurrentSong then
     begin
      MediaPlayer1.Stop;
      MediaPlayer1.Close;
      IF CurrentSong <> 'ERROR' THEN BEGIN
        MediaPlayer1.FileName := CurrentSong;
        MediaPlayer1.Open;
        MediaPlayer1.Play;
      END;
    StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
    StatusIni.WriteString('location', 'oldsong', CurrentSong);
    StatusIni.Free;
   end;

   RoomsIni :=
TIniFile.Create('d:\delphi\source\launch\rooms.INI');
   CurrentLocationExe := RoomsIni.ReadString('rooms',
CurrentLocation,'ERROR');
   RoomsIni.Free;

   if Copy(CurrentLocationExe,Length(CurrentLocationExe)-
2,3) = 'dll' then begin
    if CurrentLocation = 'living' then
    begin
      LivingroomForm;
    end else if CurrentLocation = 'kitchen' then begin
      KitchenForm;
    end;
    Timer1.Enabled:=True;
   end else begin
    StrPCopy(cmd,CurrentLocationExe);

    Other:=WinExec(cmd,SW_SHOW);
    if Other < HINSTANCE_ERROR then begin
    end else begin
      Timer1.Enabled:=True;
    end;
   end;
  end;
 end;
 end.

unit Bath1;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
  Forms, Dialogs, Inifiles, StdCtrls, ExtCtrls;

type
  TForm1 = class(TForm)
    Label1: TLabel;
    Button1: TButton;
    ToothpasteStatus: TLabel;
    Image1: TImage;
    Label2: TLabel;
    Label3: TLabel;
    Button2: TButton;
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
  GameIni : TIniFile;
begin
  GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
  GameIni.WriteString('location', 'current', 'bedroom');
  GameIni.Free;
  Halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
  GameIni : TIniFile;
begin
  GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
  GameIni.WriteString('location', 'current', 'toolshed');
  GameIni.Free;
  Halt;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
  StatusIni: TIniFile;
begin
  StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
  StatusIni.WriteString('location','current','EXIT');
  if StatusIni.ReadBool('items', 'toothpaste', False) = True then
  begin
    ToothpasteStatus.Caption := 'There''s toothpaste in the
cabinet.';
  end else begin
    ToothpasteStatus.Caption := 'You need to buy toothpaste.';
  end;
  StatusIni.Free;
end;

end.

unit Bedroom3;

interface

uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
  Forms, Dialogs, StdCtrls;

type
  TForm3 = class(TForm)
    Label1: TLabel;
    procedure Label1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;
```

```
implementation

{$R *.DFM}

procedure TForm3.Label1Click(Sender: TObject);
begin
 Close;
end;

end.

unit Drug1;

interface

uses
 SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
 Forms, Dialogs, Inifiles, StdCtrls;

type
 TForm1 = class(TForm)
  Label1: TLabel;
  Button1: TButton;
  ListBox1: TListBox;
  ListBox2: TListBox;
  ToothpasteStatus: TLabel;
  Button2: TButton;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\STAT
US.INI');
 GameIni.WriteString('location', 'current', 'kitchen');
 GameIni.Free;
 Halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteBool('items', 'toothpaste', True);
 GameIni.Free;
 ToothpasteStatus.Caption := 'You now have some
toothpaste.';
end;

procedure TForm1.FormCreate(Sender: TObject);
var
 StatusIni: TIniFile;
 B : Integer;
begin
 StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
 StatusIni.WriteString('location','current','EXIT');
 StatusIni.ReadSection('location', ListBox1.Items);
 for B := 0 to (ListBox1.Items.Count - 1) do
 begin
  ListBox2.Items.Add(StatusIni.ReadString('location',
ListBox1.Items[B],'ERROR'));
 end;
 if StatusIni.ReadBool('items', 'toothpaste', False) = True then
 begin
  ToothpasteStatus.Caption := 'You already have some
toothpaste.';
 end else begin
  ToothpasteStatus.Caption := 'You need to buy toothpaste.';
 end;
 StatusIni.Free;
end;

end.

unit Grocery1;

interface

uses
 SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
 Forms, Dialogs, Inifiles, StdCtrls;

type
 TForm1 = class(TForm)
  Label1: TLabel;
  Button1: TButton;
  ListBox1: TListBox;
  ListBox2: TListBox;
  FoodStatus: TLabel;
  Button2: TButton;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
```

```
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteString('location', 'current', 'kitchen');
 GameIni.Free;
 Halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteBool('items', 'food', True);
 GameIni.Free;
 FoodStatus.Caption := 'You now have some food.';
end;

procedure TForm1.FormCreate(Sender: TObject);
var
 StatusIni: TIniFile;
 B : Integer;
begin
 StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
 StatusIni.WriteString('location','current','EXIT');
 StatusIni.ReadSection('location', ListBox1.Items);
 for B := 0 to (ListBox1.Items.Count - 1) do
 begin
   ListBox2.Items.Add(StatusIni.ReadString('location',
ListBox1.Items[B],'ERROR'));
 end;
 if StatusIni.ReadBool('items', 'food', False) = True then
 begin
   FoodStatus.Caption := 'You already have some food.';
 end else begin
   FoodStatus.Caption := 'You need to buy groceries.';
 end;
 StatusIni.Free;
end;

end.

unit Kitchen2;

interface

uses
 SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
 Forms, Dialogs, Inifiles, StdCtrls, ExtCtrls;

type
 TForm1 = class(TForm)
   Label1: TLabel;
   Button1: TButton;
   Button2: TButton;
   Button3: TButton;
   FoodStatus: TLabel;
   ToothpasteStatus: TLabel;
   Button4: TButton;
   Image1: TImage;
   Button5: TButton;
   procedure Button1Click(Sender: TObject);
   procedure Button2Click(Sender: TObject);
   procedure FormCreate(Sender: TObject);
   procedure Button3Click(Sender: TObject);
   procedure FormDestroy(Sender: TObject);
   procedure Button4Click(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form1: TForm1;

procedure KitchenForm; export;

implementation

{$R *.DFM}
procedure KitchenForm;
begin
 try
  Form1:=TForm1.Create(Application);
  Form1.ShowModal;
 finally
  Form1.Free;
 end;
end;

procedure TForm1.Button1Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\STAT
US.INI');
 GameIni.WriteString('location', 'current', 'drug');
 GameIni.Free;
 close;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\STAT
US.INI');
 GameIni.WriteString('location', 'current', 'living');
 GameIni.Free;
 close;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
 StatusIni: TIniFile;
begin
 StatusIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\status.I
NI');
 StatusIni.WriteString('location','current','EXIT');
 if StatusIni.ReadBool('items', 'toothpaste', False) = True then
 begin
   ToothpasteStatus.Caption := 'You have some toothpaste
now.';
 end else begin
   ToothpasteStatus.Caption := 'You need to buy toothpaste.';
 end;
 if StatusIni.ReadBool('items', 'food', False) = True then
 begin
   FoodStatus.Caption := 'There''s some yummy food here.';
 end else begin
   FoodStatus.Caption := 'The cupboards are bare!';
 end;
 StatusIni.Free;
end;

procedure TForm1.Button3Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\STAT
US.INI');
 GameIni.WriteString('location', 'current', 'airstrip');
 GameIni.Free;
 close;
end;

procedure TForm1.FormDestroy(Sender: TObject);
var
StatusIni : TIniFile;
begin
StatusIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\status.I
ni');
StatusIni.WriteString('location','previous','kitchen');
StatusIni.Free;
end;

procedure TForm1.Button4Click(Sender: TObject);
var
Statusini : TIniFile;
begin
StatusIni :=
TIniFile.Create('c:\data\k\cddf\nasagame\titan\launcher\status.I
ni');
StatusIni.WriteString('location','current',statusini.readstring('loc
ation','previous','Error'));
Statusini.Free;
close;
end;

end.

unit Living1;

interface

uses
 SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
 Forms, Dialogs, Inifiles, StdCtrls;


type
 TForm1 = class(TForm)
   Label1: TLabel;
   Button1: TButton;
   Button2: TButton;
   ListBox1: TListBox;
   ListBox2: TListBox;
   procedure Button1Click(Sender: TObject);
   procedure Button2Click(Sender: TObject);
   procedure FormCreate(Sender: TObject);
   procedure FormDestroy(Sender: TObject);
 private
   { Private declarations }
 public
   { Public declarations }
 end;

var
 Form1: TForm1;
implementation


{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteString('location', 'current', 'bedroom');
 GameIni.Free;
 Halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteString('location', 'current', 'kitchen');
 GameIni.Free;
 Halt;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
 StatusIni: TIniFile;
 B : Integer;
begin
 StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
 StatusIni.WriteString('location','current','EXIT');
 StatusIni.ReadSection('location', ListBox1.Items);
 for B := 0 to (ListBox1.Items.Count - 1) do
 begin
   ListBox2.Items.Add(StatusIni.ReadString('location',
ListBox1.Items[B],'ERROR'));
 end;
 StatusIni.Free;
end;

procedure TForm1.FormDestroy(Sender: TObject);
var
 StatusIni: TIniFile;
begin
 StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
 StatusIni.WriteString('location', 'previous','living');
 StatusIni.Free;
end;
```

```
end.


unit Tools1;

interface

uses
 SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
Controls,
 Forms, Dialogs, Inifiles, StdCtrls, ExtCtrls;

type
 TForm1 = class(TForm)
  Label1: TLabel;
  Button1: TButton;
  Image1: TImage;
  Image2: TImage;
  procedure Button1Click(Sender: TObject);
  procedure Button2Click(Sender: TObject);
  procedure FormCreate(Sender: TObject);
  procedure Image2Click(Sender: TObject);
 private
  { Private declarations }
 public
  { Public declarations }
 end;

var
 Form1: TForm1;


implementation

{$R *.DFM}

procedure TForm1.Button1Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteString('location', 'current', 'ball');
 GameIni.Free;
 Halt;
end;

procedure TForm1.Button2Click(Sender: TObject);
var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteString('location', 'current', 'toolshed');
 GameIni.Free;
 Halt;
end;

procedure TForm1.FormCreate(Sender: TObject);
var
 StatusIni: TIniFile;
begin
 StatusIni :=
TIniFile.Create('d:\delphi\source\launch\status.INI');
 StatusIni.WriteString('location','current','EXIT');
 StatusIni.Free;
end;

procedure TForm1.Image2Click(Sender: TObject);
 var
 GameIni : TIniFile;
begin
 GameIni :=
TIniFile.Create('d:\delphi\source\launch\STATUS.INI');
 GameIni.WriteString('location', 'current', 'bath');
 GameIni.Free;
 Halt;
end;

end.
```

ACKNOWLEDGMENTS

I appreciate the tremendous help of my NASA Research Colleague, Kenneth C. Jenks, this summer. I am grateful for his advice in selecting this project and to NASA for the opportunity to work here. I also appreciate the ASEE Summer Faculty Program and its JSC coordinator, Prof. Richard B. Bannerot for all his cheerful enthusiasm, patience and assistance in showing the new faculty around JSC with tours and seminars. I also am pleased to acknowledge the excellent software programming of a number of students: John D. Smith, Donna Panter, Jimmy Elmore, Shruti Bahadur, and Hong Zhang.