# An Empirical Approach of The Air Battle Game

1155566

**Abstract**

Air battle is a kind of classic strategy game which is similar to the battleship game, but instead of navy ships it has airplanes. The shape of plane is like "士", which is mainly different with the shape of battleship in battleship game. The purpose of this paper is to illustrate the applicability of Empirical Modeling principle to such a process by establishing the model of the game. Generally, there are two players will be needed in this game, and every player hide three planes respectively, however this model only constructs simple model which only have one plane. In addition, depending on every action, the potential position of the tip of plane will be discussed and implemented in the third section.

## 1. Introduction

### 1.1 Empirical Modeling

With the strengths of capability the model an open, experimental and experiential environment, EM tools tend to easily experiment the models of many strategy games or models based on real world [1]. As the definitions based language, empirical modeling is an essentially agent-oriented method to programming. More precise, three types of aspects, namely observables, dependency and agency, are included in the EM, and a categorical analysis is given by identifying and documenting them. Firstly, observables are entities which can get seizures depend on experiment. Then, dependency is some variable or constant to connect or relate observables. Thirdly, agency is an action to change the current situation of the entities or observables in the model.

Depending on the principle of EM, in this paper, a classic game named air battle is constructed simply using EDEN, the Evaluator of Definitive Notations. The EDEN interpreter, which was designed by University of Warwick in 2005, has ability to observe the changes through the updating the dependencies instantaneously by program [2]. In EDEN, three tools which are EDEN, DoNaLD and SCOUT combine to construct the model of the game. Firstly, SCOUT defines the layout of screen, and then with the DoNaLD, some required 2-D line, shape or other type of data can be drew in a specific window which has been defined in SCOUT. Finally, in EDEN, some essential operations or functions are defined to operate and calculate the constraints. However, during the process of modeling the game, some weaknesses, such as incomplete, poorly documented, are appeared, and cause some difficulties to us.

### 1.2 Air Battle Game

Air battle is a kind of classic strategy game which is similar to the battleship game, but instead of navy ships it has airplanes. The shape of plane is like "士", which is mainly different with the shape of battleship in battleship game. Generally, in this game, two players hide each three planes ant try to shoot down the planes of enemy by hitting in turns. A plane is killed by a hit at the tip. No further hits are required. However, for the reason of simply model the game and the some complex issues which will also mentioned at the end of report, this study only implements one plane, and two players play the game in a only 8*8 grid.

In terms of the process of the game, firstly, the first player arranges a plane in the grid, and the tip of the plane must be arranged firstly. Based on the correct method of the arrangement of the plane, the button of "hide plane" will available to press, which make the plane disappear from the grid. Then, the second player tends to hit the tip of the plane. If he clicks the every grid to find the tip, and based this action, the grid which is clicked will change the color to show this grid is the tip, part of plane or empty, and the related word description will be produced. If player two hit the tip successfully, the "show plane" button will available and the whole plane can be demonstrated. At the same time, the step count will be recorded and showed finally.

This study not only describes the structure of the model, and the method to implement the every step, but also discuss a strategy to provide a forecast of the potential of the tip of the plane. Additionally, the dependencies between observables will analysis and express in next part.

## 2. The Air Battle Game

The following section will document the implement of game model using the EDEN, DoNaLD and SCOUT.

### 2.1 Interface

The construction of the interface is mainly based on the tools of SCOUT and DoNaLD. Using SCOUT to draw the main windows on the interface can construct the layout easily.

In terms of the 8*8 grid, the DoNaLD will be used to draw 14 lines to separate the grid into 64 little grids. Based on that, the 64 little grids will be defined as "rectangle" type. There are 4 buttons and four text boxes on the interface. Precisely, the first text box is to show the current result of hitting action, and also guiding the player to play the game. The second one is to record the history of the hitting. Each step will be shown sequentially in this box. The last two boxes are to record the count of step in current game and the best score respectively. The figure3 give the interface of the game.

### 2.2 Design of Game

The game is designed into two main steps. They are arrangement and hitting, and 4 detailed steps will be included in these two majority step. In arrangement step, player one are guided to arrange the plane correctly, and player two tend to hit the tip of plane with minimum times.

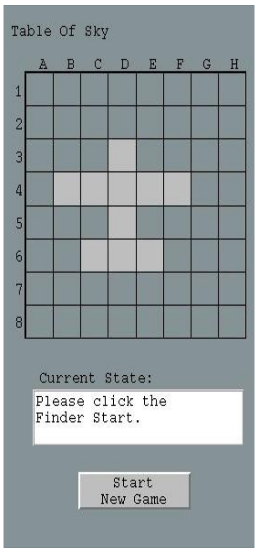
### 2.2.1 Dependencies

In each main step, there are several of dependencies to establish the relationship between different observables and different steps. In the arrangement step, some constraints declare as follow.

- $stat_{ij}$, i represent horizontal of grid {A,B,C,D,E,F,G,H}, and j represent ordinates{1,2,3,4,5,6,7,8}. If $stat_{ij}$ =1 means $grid_{ij}$ is a part of plane, $stat_{ij}$ =2 means $grid_{ij}$ is tip, and 0 means there is nothing. The value of it is based on the position of mouse click in arrangement step. Also, it can also avoid click same grid repeatedly. Essentially, hitting step is based on these parameters.
- $size_{ij}$ is from1 to 64, and represent each little grid from A1 to H8 respectively. This parameter is mainly to judge whether the plane is arranged correctly.
- Limitecount is to limite player 1 click more grid, and judge which step should be entire. For instance, limitecount=10, step 3 is available, and if limitecount=11 which means step3 is finished, step4 is available.

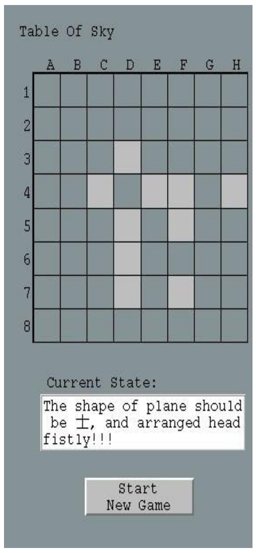
As shown from figure1 and figure2, the judgment of arrangement can be recognized effectively.

Judgment of the arrangement is quite a thorny problem in this study. However, according to the values of parameters above, the problem will be resolved effectively. I use “sum” to record the sum the sizes which has been clicked.



Please click the Finder Start.

Figure1:Correct



The shape of plane should be 士, and arranged head fistly!!!

Figure2: Incorrect

For the reason of requirement of arranging tip first, we can com “sum” and $size_{ij}$ to calculate the shape and the tip of current plane. In study, I us function named “isitPlane” to realize this process.

Hitting step is quite complex, the dependencies are as follow.

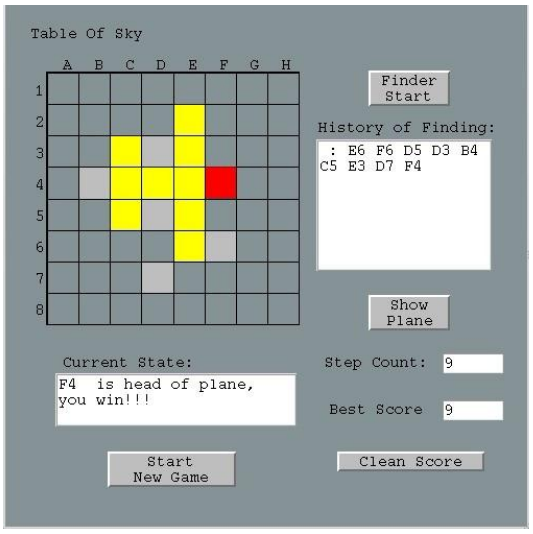
- $stat_{iijj}$ is to judge whether “ij” grid has been clicked in hitting step. It quite same as$stat_{ij}$. The “step count” text box is dependent this parameter.
- $size2_{ij}$ is not use in this model so far, it is designed to implement the function of tip position forecast. the forecast is essentially related to this parameter. The detailed design will

be given in next subdivion.

- Findcount is used to record the sum of hitting times for current turn of game.

As demonstrated of figure4, a case of successful hitting with 9 steps is given. More precise, yellow means player2 hits the body of plane and red means hits tip and the game finished. The drawing of color dependent the value of$stat_{ij}$, and the related the value of x coordinates and y coordinates which is produced by action of mouse press. Here take b2 as a case to show the method of hitting step.

```
if (tablemouse_mouse[4]<200 && tablemouse_mouse[4]>100
 && tablemouse_mouse[5]<700 && tablemouse_mouse[5]>600
              && statbb2==0){
  statbb2++;findcount++;HeadP(statb2);}
```



History of Finding:

```
 : E6 F6 D5 D3 B4
C5 E3 D7 F4
```

Current State:

```
F4  is head of plane,
you win!!!
```

Step Count: 9

Best Score 9

Figure4: The hitting process

### 2.2.2 design of tip forecast

Actually, the method to realize the forecast of the potential tip position has been designed. Firstly, the two of list are built, which named chequer and sizelist to store the $stat_{ij}$ and $size2_{ij}$ which is mentioned above. Therefore, each list will include 64 elements. A function named potentialHead is defined, the $stat_{ij}$, $size2_{ij}$, values of mouse press land on x coordinate and y coordinate will be the inputs. Future more, in this function, depending on judge the area of x and y value in the grid, every $size2_{ij}$ of potential tip will be calculate, then execute code as follow, we can figure out the potential tip position.

```
execute("A_table_"//str(chequer[sz+/-x])//"
          is "fill=solid, color=\"\"\";");
```

## 3. Conclusion

In this study, we introduce the method and the tool of modeling the air battle game, and provide detailed step to implement the model, such as the judgment of arrangement, hitting process. Finally, a function of forecast is also to discuss.

Moreover, this study is to contribute that how such EM principle is useful in constructing the strategy, and how to construct a model through the EM tool, EDEN.

## Reference

[1] Airport Layout Planning, University of Warwick, 2010.

[2] Hunt the Wumpus: an Empirical Approach, University of Warwick, 2007.