

CANNONBALL SIMULATION

Cem OZTURAN

1061211

Department of Computer Science

University of Warwick

ABSTRACT

This project is about creating a cannonball simulation using Empirical Modelling tools, where the user simply tries to hit a target shown on the screen by firing. The project mainly focuses on the physical movement of the ball taking all the defined dependencies into consideration, and it has been developed using tkeden 1.67 by considering the Empirical Modelling concepts.

TABLE OF CONTENT

1. Introduction.....	3
1.1 Empirical Modelling.....	3
1.2 Empirical Modelling Tools used for Cannonball Simulation.....	3
1.2.1 EDEN.....	3
1.2.2 SCOUT.....	3
1.2.3 DoNaLD	4
2. Research.....	4
3. Design and Implementation.....	4
4. Evaluation of the Model.....	5
5. References.....	5

1. INTRODUCTION

1.1 Empirical Modelling

Empirical Modelling (EM) is considered as a new approach to computing, which has been developed for many years in Computer Science department at the University of Warwick since 1981 [1]. The EM concepts are defined to be “centrally concerned with the role that model-building can play in personal understanding of situations” [1]. EM brings an original conceptual framework which focuses on sense-making activities in software development [2].

Empirical Modelling projects aim to simulate real-world situations by considering the observations and defining relevant dependencies. The models can be improved more and more as new dependencies or functionalities are added to the existing models depending on the experimentations and comparisons between the model and the real-world situations.

A number of different tools have been developed for EM projects since 1981. Brief information about the tools used in the development process of “Cannonball Simulation” is given in the following subsection.

For the rest of the paper, definitions of the key concepts are as follows[1]:

SCOUT: Screen LayOUT

DoNaLD: Definitive Notation for Line Drawing

EDEN: Evaluator of Definitive Notations

1.2 Empirical Modelling Tools used for Cannonball Simulation

The main tool used is *tkeden*, where the three notations (EDEN,SCOUT,DoNaLD) described below is implemented. The files for the notations can be imported and executed using *tkeden*, or the codes can be run from the console window of *tkeden*.

1.2.1 EDEN

EDEN is the current primary language used for EM[3]. Although it has a similar syntax to C language, it also has very important differences since EM is the main area for the usage of EDEN. The most important aspect of EDEN is that there are definitive statements in EDEN [4]. The difference between a traditional assignment statement (**variable = new value**) as in many programming language and definitive statements (**variable is new value**) of EDEN is that variable values are changed dynamically in a definitive statement.

Besides, procedures in EDEN can also be defined as dependent on certain variables. So, if the value of a variable which a procedure depends on changes, then the procedure is executed again using the latest value of the associated variable.

1.2.2 SCOUT

SCOUT basically deals with items displayed on the screen. Windows and buttons can be defined using SCOUT and they can be shown on or removed from the main window, which the user actually interacts with the model.

As well as EDEN, SCOUT is also a definitive notation so the variables in SCOUT can be changed using EDEN and displayed items are updated accordingly.

1.2.3 DoNaLD

DoNaLD is used to draw two-dimensional basic shapes such as lines, points, circles and so. Like the two notations explained above, DoNaLD is also purely definitive. Another aspect of DoNaLD is the usage of viewports where the drawings are created. By assigning certain drawings to different viewports, the developer can display some objects on the screen while hiding some other objects.

The objects drawn using DoNaLD have also some certain attributes like color, position, size. Using EDEN, these attributes can be changed as the developer wishes.

2. RESEARCH

Although the EM area is such a wide one in itself, the usage of the tools is mainly restricted to the Department of Computer Science at the University of Warwick. The materials provided online cover a lot of basic concepts, and the projects archive is of great help.

For the cannonball simulation proposed, the most important two things to study are physical movement of an object which has an initial speed and is subject to gravity, and how to implement the model in an Empirical Modelling environment.

For the physics part, equations of motion and the effects of the dependencies are the key points [5]. For the implementation of these formulas in the simulation, tutorials and the previous projects must be well understood. It is obvious that EM is a new area even for a good traditional programmer and many unexpected results can occur if the concepts are not understood very well.

3. DESIGN AND IMPLEMENTATION

The model has been designed to have two screens for ease of use. One screen is the main user interface screen, where the user can change any dependency variable for the simulation. Since it is a cannonball simulation, there are a number of dependencies, which allows users to experiment different shots. In the other screen, the user can observe the shooting and the ball's motion. This screen has no user interact.

So, the design of the model simply aims to be as user-friendly as possible, and not a very difficult task to achieve since the drawings in the model are just very simple models of real world objects.

The implementation stage, however, is more challenging. In the implementation process of the model, a step-by-step path has been followed. After the drawings of the objects and the user interface, physics formulas for the ball's motion are implemented. One of the key factors here is taking the vertical and horizontal components of the ball's initial speed into consideration. For this, the angle provided by the user is used. When these components are found, the horizontal speed does never change until the ball hits the ground since there is no collision and air resistance is ignored. However, the vertical speed is updated dynamically because of the gravitational force. This means, even if the user does not change the value of any dependency, ball's motion is still affected by gravity as it would be in the real world. Another important aspect used in implementation is EDEN clocks. When the ball is fired, a clock is associated with

a variable and updates the ball's position in such a way that the user can observe a smooth motion, which is important in visualizing such a real-world situation.

When the ball's motion is handled, simpler dependencies are also implemented, which are namely the height of the cannonball, and the physical attributes of the target area. Although this part is easier than the ball's motion, updating the drawings accordingly is a more difficult task than expected. The developer must understand the "definitive" statements provided by EDEN very well to get over such problems.

As the last part of the implementation process, a box is added to the user interface screen, which informs the user about the latest position in the simulation. It is also an important thing to include in the model. By checking this box, the user can know if the model is ready to shoot, or whether the shots are successful or not. The box simply checks the ball's position as well as the clicks on the other boxes, and updates itself accordingly.

4. EVALUATION OF THE MODEL

The model has almost all the dependencies for the simulation of a real-world situation. Also, these are provided very clearly in the user interface, which is very functional for any user. For ease of use, the units for distance and angle are converted to meters and degrees instead of pixels and radians. So, any user can experiment with the simulation in very different conditions without a problem.

However, there are some possible improvements as well. The physics implemented are almost 100% correct, but the real world theoretically has more dependencies. For example, the mass of the ball is ignored in the model since the mass has no importance if there is no air resistance. The effect of air resistance on such a ball is almost 0, but it still exists. Even if the ball is extremely heavy or has a very small volume, air resistance would still occur in real world.

Also, another possible improvement could be addition of some obstacles to the environment. Then, the developer must take object collisions into consideration. One last thing to do can be improvement of the drawings. For now, the model has a very simple visualization.

Still, the model provides everything necessary to experiment with a cannonball simulation satisfactorily as it is in real world.

5. REFERENCES

[1] Meurig Beynon. *The Warwick Empirical Modelling Homepage [Online]*

Available from: <http://www.dcs.warwick.ac.uk/research/modelling/> [Accessed January 2011]

[2] N. Pope, M. Beynon. Empirical Modelling as an unconventional approach to software development.

Available: <http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/papers/downloads/113.pdf>

[Accessed January 2011]

[3] Chris Roe, "The EMPTY project," MSc Thesis, Department of Computer Science, University of Warwick, UK (September 1999).

Available: http://www2.warwick.ac.uk/fac/sci/dcs/research/em/publications/mseprojects/croe/pdf/dis_2.pdf

[Accessed January 2011]

[4] Daniel Keer(2005), Modelling Navigation and Landmarking in Ants
Available: <http://empublic.dcs.warwick.ac.uk/projects/antnavigationKeer2005/docs/report.pdf>
[Accessed January 2011]

[5] Halliday, David; Resnick, Robert; Walker, Jearl (2004-06-16). *Fundamentals of Physics* (7 Sub ed.)