Breaking Effect

Xiaoye Ma, max58@mcmaster.ca

November 15, 2017

1 Revision History

Date	Version	Notes
2017-10-01	1.0	New document

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	**unit**	**SI**
m	length	metre
kg	mass	kilogram
s	time	second
J	energy	Joule

2.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the heat transfer literature and with existing documentation for solar water heating systems. The symbols are listed in alphabetical order.

symbol	**unit**	**description**
S	unit of length	displacement
t	s	the time in seconds since the start of the game
v_0	m/s	initial speed
a	m/s^2	acceleration
g	m/s^2	gravity acceleration
ΔE_k	J	variation of kinetic energy
ΔE_p	J	variation of potential energy
W_f	J	work done by kinetic friction
x_n	m	x coordinates of gravity center of piece n, $n \in \mathbb{N}$, which represents just one piece. [Explain the meaning of n? Also, you should format it like a mathematical variable (n, not n). —SS][Add explanation for n and format it as a mathematical variable. —Author]
y_n	m	y coordinates of gravity center of piece n
z_n	m	z coordinates of gravity center of piece n
θ_1	degree	angle between initial speed and horizontal

θ_2	degree	angle between x axiom and projection on horizontal of initial speed
S_x	m	displacement on direction of x axiom
S_y	m	displacement on direction of y axiom
S_z	m	displacement on direction of z axiom
μ_k		coefficient of friction

2.3 Abbreviations and Acronyms

symbol	**description**
A	Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
Breaking Effect	Breaking Effect
T	Theoretical Model

Contents

3 Introduction

Breaking effect presents how the pieces of an object move after it separates into parts with suddenness or violence. [Add explanation for BE here —Author]

Because of the development of video games industrial and hardware such as CPU and GPU, there is a higher demand for high level experience in game visualization. Breaking effect plays a more important role in the visualization level of large scale video games. [Tell the reader what the breaking effect is. —SS] This project simulates the process of 3D objects destruction.

The following section provides an overview of the Software Requirements Specification (SRS) for a breaking effect program. The developed program will be referred to as Breaking Effect (BE). This section explains the purpose of this document, the scope of the system, the characteristics of the intended readers and the organization of the document.

[The text is better for version control, and for reading in other editors, if you use a hard-wrap at 80 characters —SS]

3.1 Purpose of Document

The main purpose of this document is to describe the modeling of breaking effect. The goals and theoretical models used in the breaking effect code are provided, with an emphasis on explicitly identifying assumptions and unambiguous definitions.

3.2 Scope of Requirements

The scope of the requirements is limited to breaking effect of a single 3D object applied by force. Interact force between objects and collision among several objects are not in the scope. Given the appropriate inputs, the code for BE is intended to calculate pieces motion and display the process of target 3D object breaking in vision. The project is implemented in a game engine. 3-D objects and piece generation function are provided by platform.

3.3 Characteristics of Intended Reader

Reviewers of this documentation should have a strong knowledge in physics motion theory and an understanding of differential equations. A third year Physics course on this topic in high school is recommended. [For the characteristics of intended reader try to be more specific about the education. What degree? What course areas? What level? —SS] [Add specific course and level —Author]

3.4 Organization of Document

The organization of this document follows the template for an SRS for scientific computing software proposed by Smith and Lai (2005); Smith et al. (2007). [Copy the format from original template. —Author] The presentation follows the standard pattern of presenting

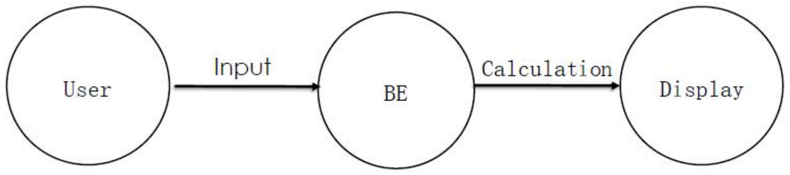

Figure 1

goals, theories, definitions, and assumptions. [You should use BibTeX for this. The original template showed you how to do this. If you have questions about how to use BibTeX, please ask your classmates, or me. —SS]

4 General System Description

This section identifies the interfaces between the system and its environment, describes the user characteristics and lists the system constraints.

4.1 System Context

- User Responsibilities:
  - Provide inputs and make sure they are in an appropriate range including position of target object, initial momentum, coefficient of friction on the ground. [Try to be more specific about what inputs the user provides. Determining whether inputs are in the appropriate range seems like something your program could easily test. —SS][Yes ! Make this more specific by listing necessary inputs. —Author]
- Breaking Effect Responsibilities:
  - Detect data type mismatch, such as a string of characters instead of a floating point number
  - Determine if the inputs satisfy the required physical and software constraints as shown in 5.2.6. [Is the appropriate range (from the user responsibilities above) included in the constraints? —SS][Refer to data constraints section. —Author]
  - Calculate the required outputs

4.2 User Characteristics

The end user of Breaking Effect should have a basic knowledge of Physics and Dimensional geometry as well as experience of using Unity3D. A background of high school level Physics and Dimensional geometry course are recommended. [Please be more specific on the users education background and level. —SS][Add specific course and level. —Author]

4.3 System Constraints

This project is implemented in Unity3D because it relies on 3D models and existing function for object cutting. Users have to use Break effect in Unity3D project. The algorithm can be implemented in other physics engines but for practical reasons they will not be considered in this project. [I think you have to include Unity as a system constraint, or remove the mention of it elsewhere in the document. You have obviously (for practical reasons) decided that you are going to use Unity. You should explicitly inform the user of this. I also suggest that you mention that there are other options for physics engines, but for practical reasons they will not be considered in this project. —SS][Include Unity3D as a system constraint and provide more information for users. —Author]

5 Specific System Description

This section first presents the problem description, which gives a high-level view of the problem to be solved. This is followed by the solution characteristics specification, which presents the assumptions, theories, definitions and finally the instance models.

5.1 Problem Description

This project tries to implement running time breaking effect in codes for 3-D models in unity3D without help from any similar plug-in. Including different shapes 3-D objects breaking based on physics and pieces interacting with the momentum provided by the breaking force. The breaking effect program simulates 3-D objects destruction process in vision by implementing scientific computing functions. This project concentrates on calculation while HCI or GUI are not important parts. Applied force is decided in codes in advance as input and trace of motion is the output after calculation.

5.1.1 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Focus point: The location where explosion happens [spell check! —SS][Fixed. Sorry for the mistake ! —Author]
- initial momentum level: Initial momentum of pieces

5.1.2 Physical System Description

The physical system of Breaking Effect includes visual terrain [spell check! —SS][Fixed. Sorry for the mistake ! —Author] and a target 3-D object which will break into pieces.

5.1.3 Goal Statements

Given the target object, coefficient of friction, focus point and initial momentum level, the goal statements are:

GS1: Calculate initial speed of each piece.

GS2: Calculate trace of motion for each piece.

5.2 Solution Characteristics Specification

The instance models that govern Breaking Effect are presented in Subsection 5.2.5. The information to understand the meaning of the instance models and their derivation is also presented, so that the instance models can be verified.

5.2.1 Assumptions

This section simplifies the original problem and helps in developing the theoretical model by filling in the missing information for the physical system. The numbers given in the square brackets refer to the theoretical model [T], general definition [GD], data definition [DD], instance model [IM], or likely change [LC], in which the respective assumption is used.

A1: The forms of energy that are relevant for this problem are kinetic energy and potential energy. Thermal energy is considered on the ground but it is assumed to be negligible in the air. All other forms of energy are negligible. [but you have friction? Work will be done that eventually uses up the initial energy. —SS][Yes, I do consider kinetic friction on the ground and only air friction is ignored. I modify A1 and A2 to make them more specific. Sorry for the confusion ! —Author]

A2: Air friction will be ignored while kinetic friction is considered on the ground and it is assumed to be constant.

A3: Powders and airborne [spell check! —SS] [Fixed, sorry for the mistake ! —Author]dust will not be considered. Because they both have random shapes, motions and they are seriously influenced by air friction. Powders are considered as very tiny size pieces that are generated in explosion. [missing period. —SS] [What makes a piece powder? How is powder defined? —SS][Add explanation for powder and airborne dust. —Author]

A4: Before explosion, the kinetic energy and potential energy of target object are zero. Target object cannot be a moving object. An initial momentum will be assigned to each piece after explosion happens. [You say in your goal statement that there is an initial momentum. If the kinetic energy is zero, then the momentum will always be —SS][There is some misunderstanding in this assumption. The initial energy here represents the energy before explosion. And initial momentum in GS refers to the energy after explosion happens. So they are totally different status because of different periods. I modify the expression to make it clearer. Sorry for the confusion ! —Author]

A5: The same coefficient of friction everywhere on flat ground.

A6: Pieces are not decomposable.

A7: Collision between pieces will not be considered.

5.2.2 Theoretical Models

This section focuses on the general equations and laws that Breaking Effect is based on.

Number	T1
Label	**Conservation of mechanical energy**
Equation	$E_{mech} = E_k + E_p$
Description	The total mechanical energy (defined as the sum of its potential and kinetic energies) of a particle being acted on by only conservative forces is constant. The potential energy, E_p, depends on the position of an object subjected to a conservative force. The kinetic energy, E_k, depends on the speed of an object and is the ability of a moving object to do work on other objects when it collides with them.Assumption 2 is necessary for this equation to be true when pieces move in the air. This TM only works for motion in the air. After explosion happens, pieces do parabolic movement on vertical direction in which acceleration is always g, and do uniform motion on horizontal direction in the air. [What about the work done by nonconservative forces, like friction? This should be part of your conservation of energy. The equation you have written is only true in the case where the work done by nonconservative forces is zero. You should also reference the assumptions that are necessary for your theoretical model to apply. —SS][When pieces move in the air, nonconservative forces are ignored so that only energy are considered. When pieces move on the ground, I do consider kinetic friction. This TM only works for motion in the air. While T2 below is used for motion on the ground. I modify the description to make it clearer. —Author]
Source	http://www.nuclear-power.net/laws-of-conservation/law-of-conservation-of-energy/conservation-of-mechanical-energy/
Ref. By	IM2

Number	T2
Label	**Work done by kinetic friction**
Equation	$\Delta E_k = W_f$
Description	Kinetic energy loses due to kinetic energy. W_f is work done by kinetic friction. As a result, kinetic energy transform to internal energy.When piece move on the ground, they do constant decelerated motion and acceleration is constant and negative because of constant kinetic friction on the ground.
Source	http://teacher.pas.rochester.edu/phy121/lecturenotes/Chapter07/Chapter7.html
Ref. By	IM3

5.2.3 General Definitions

There is no general definition for current problem.

5.2.4 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given.

Number	DD1
Label	**Displacement in uniformly accelerated/decelerated motion**
Symbol	S
SI Units	M
Equation	$S = v_0 t + \frac{1}{2}at^2$
Description	The above equation gives us the distance traveled without having to know the final velocity of the object. Where t is time duration and v_0 is initial speed. Acceleration a is defined as the rate of change of velocity with respect to time, in a given direction. This would mean that if an object has an acceleration of 1 m/s^2 it will increase its velocity (in a given direction) 1 m/s every second that it accelerates. This equation is tenable under A1, A2 and A5 in this project. [This equation only applies for constant acceleration. As the object loses energy the acceleration will decrease. None of your assumptions mention assuming constant acceleration. The equation you are using comes from kinematics, where you don't need to worry about forces. I think you do have to worry about forces, which moves it to the area of kinetics. —SS][Sorry that I made it confused here. Actually I believe it is a misunderstanding due to my unspecific expression. I do consider deceleration because of work done gravity force when pieces move in the air and work done by friction when pieces move on the ground. I add decelerated motion in label above and modify assumption 2 to consider kinetic friction on the ground as constant force. And since gravity and friction are considered as constant force, acceleration are constant positive and negative respectively. —Author]
Sources	http://ibphysicsstuff.wikidot.com/uniformaccmotion
Ref. By	IM2,IM3

Derivation of how to derive the equation from relationship of position, velocity and acceleration

We assume v_0 is initial velocity and v_t is the final velocity.

Then we have: $v_t = v_0 + at$

We can get average velocity v_a
$v_a = \frac{v_0+v_t}{2}$

So we can get displacement

$S = v_a t = \frac{v_0+v_t}{2} t$

$S = \frac{v_0+v_0+at}{2} t = v_0 t + \frac{1}{2} a t^2$

Number	DD2
Label	**Kinetic friction**
Symbol	F_k
SI Units	N
Equation	$F_k = \mu_k F_n = \mu_k mg$
Description	Kinetic friction F_k is a force that acts between moving surfaces. An object that is being moved over a surface will experience a force in the opposite direction as its movement. The magnitude of the force depends on the coefficient of kinetic friction between the two kinds of material. Every combination is different. The coefficient of kinetic friction is assigned the Greek letter "mu" μ, with a subscript "k". [Use "quote" to get correct quotation marks —SS] The force of kinetic friction is μ_k times the normal force on an object, and is expressed in units of Newtons (N). In this project, F_n equals to mg. [gravity is an acceleration, not a force. Your force is mg, where m is the mass of the object. Do you know the mass of the objects? —SS][Yeah actually I know force is mg, g is acceleration. However I am not sure if gravity in English means gravitational force Maybe the confusion here is because I misuse the word. —Author]
Sources	http://www.softschools.com/formulas/physics/kinetic_friction_formula/92/
Ref. By	IM3

5.2.5 Instance Models

This section transforms the problem defined in Section 5.1 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 5.2.4 to replace the abstract symbols in the models identified in Sections 5.2.2 and 5.2.3.

The goals GS1 and GS2 are solved by IM1 to IM3. Piece generation is done by calling cutting function in game engine.

Number	IM1
Label	**Find the angle between v_0 and horizontal. Find the angle between x axiom and projection on horizontal of initial speed**
Input	x_n,y_n,z_n
Output	$\theta_1 = arctan\frac{\|z_n\|}{\sqrt{x_n^2+y_n^2}}$ $\theta_2 = arctan\frac{y_n}{x_n}$
Description	x_n is x coordinates of gravity center of piece n. y_n is y coordinates of gravity center of piece n. z_n is z coordinates of gravity center of piece n.
Sources	
Ref. By	IM2

Number	IM2
Label	**Uniformly accelerated motion to find displacement in the air**
Input	v_0,θ_1 from IM1,θ_2 from IM1,t,g
Output	$S_x = v_0 \cdot cos\theta_1 \cdot cos\theta_2 \cdot t$ $S_y = v_0 \cdot cos\theta_1 \cdot sin\theta_2 \cdot t,$ $S_z = v_0 \cdot sin\theta_1 \cdot t - \frac{1}{2}gt^2$
Description	v_0 is the initial speed. t is time from beginning. θ_1 is the angle between v_0 and horizontal. θ_2 is the angle between x axiom and projection on horizontal of initial speed. The above equation applies as long as the piece moving in the air.
Sources	
Ref. By	

Number	IM3
Label	**Uniformly accelerated motion to find displacement on the ground**
Input	v_0,θ_1,θ_2,t,g,μ_k
Output	$a = \mu_k g$ $S_x = v_0 \cdot cos\theta_1 \cdot cos\theta_2 \cdot t - \frac{1}{2}at^2$ $S_y = v_0 \cdot cos\theta_1 \cdot sin\theta_2 \cdot t - \frac{1}{2}at^2$,
Description	v_0 is the initial speed. t is time from beginning. θ_1 is the angle between v_0 and horizontal. θ_2 is the angle between x axiom and projection on horizontal of initial speed. μ_k is the Coefficient of friction. The above equation applies as long as the piece moving on the ground.
Sources	
Ref. By	

Derivation of how to get angle between v_0 and horizontal

We have initial location (X,Y,Z) of target object, and location (x_n, y_n, z_n) for piece n, then we can calculate the angle between v_0 and horizontal by:

$$tan\theta_1 = \frac{|z_n - Z|}{\sqrt{(x_n - X)^2 + (y_n - Y)^2}}$$

If location of target object is (0,0,0), then we have:

$$tan\theta_1 = \frac{|z_n|}{\sqrt{x_n^2 + y_n^2}}$$

$$\theta_1 = arctan\frac{|z_n|}{\sqrt{x_n^2 + y_n^2}}$$

[You really need to work on the physics in your project. You might understand the physics better when you proceed to the next steps. For instance, you will learn about the problems with applying DD1 - your particles would never stop, they would just keep accelerating. For your project faking a rational design process should be very helpful. —SS] [Sorry for the confusion here ! Actually when pieces move on the ground, the acceleration a in DD1 is a negative number because I do consider friction on the ground. So that particles do constant decelerated motion and will actually stop eventually. In fact, after explosion happens, pieces do parabolic movement on vertical direction in which acceleration is always g, and do uniform motion on horizontal direction in the air. When piece move on the ground, they do constant decelerated motion and acceleration is constant and negative because of constant kinetic friction on the ground. I add more expression in my TM to make my models clearer. —Author] [You don't say how the object will be broken into pieces. You have this goal, but as far as I can tell, you never return to it. You should add goals to your traceability matrix, so that you would see this omission. —SS] [I prefer to use existing function provided by Unity3D because how to split the object into pieces is not emphasis. I can specify this in system constrain and delete object cutting in GS1. —Author]

5.2.6 Data Constraints

Tables 1 and 3 show the data constraints on the input and output variables, respectively. The column for physical constraints gives the physical limitations on the range of values that can be taken by the variable. The column for software constraints restricts the range of inputs to reasonable values. The constraints are conservative, to give the user of the model the flexibility to experiment with unusual situations. The column of typical values is intended to provide a feel for a common scenario. The uncertainty column provides an estimate of the confidence with which the physical quantities can be measured. This information would be part of the input if one were performing an uncertainty quantification exercise.

The specification parameters in Table 1 are listed in Table 2.

Table 1: Input Variables

Var	Physical Constraints	Software Constraints	Typical Value	Uncertainty
X		$L_{\text{min}} \leq X \leq L_{\text{max}}$	0	10%
Y		$L_{\text{min}} \leq Y \leq L_{\text{max}}$	0	10%
Z		0	0	0%
x_n		$L_{\text{min}} \leq x_n \leq L_{\text{max}}$	0	10%
y_n		$L_{\text{min}} \leq y_n \leq L_{\text{max}}$	0	10%
z_n		$L_{\text{min}} \leq z_n \leq L_{\text{max}}$	0	10%
v_0	$v_0 > 0$	$v_{\text{min}} \leq v_0 \leq v_{\text{max}}$	20 m/s	10%
μ_k	$\mu_k > 0$	$\mu_{\text{min}} \leq \mu_k \leq \mu_{\text{max}}$	0.05	10%
E		$E_{\text{min}} \leq E \leq E_{\text{max}}$	5	10%

5.2.7 Properties of a Correct Solution

A correct solution must exhibit the principle of motion as well as conservation of energy.

Table 2: Specification Parameter Values

Var	Value
v_{min}	10 m/s
v_{max}	100 m/s
L_{min}	-1000
L_{max}	1000
μ_{min}	0
μ_{max}	1

Table 3: Output Variables

Var	Physical Constraints
S_x	$S_x \geq 0$
S_y	$S_y \geq 0$
S_z	$S_z \geq 0$

6 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

6.1 Functional Requirements

R1: Input the following quantities, which define the target object and initial conditions:

symbol	unit	description
X		x coordinates of target object
Y		y coordinates of target object
Z		z coordinates of target object
E		Initial momentum level
μ_k		Coefficient of friction everywhere on flat ground

R2: Use inputs in R1 to find initial speed v_0, as follows:

$v_0 = E \cdot 10$

R3: Verify that the inputs satisfy the required physical constraints shown in Table 1.

R4: Generate pieces by calling cutting function in game engine. Then get gravity center of each piece.

R5: Calculate the angle between v_0 and horizontal(θ_1). Calculate the angle between x axiom and projection on horizontal of initial speed(θ_2) by IM1.

R6: Calculate and output trace of motion for each piece in the air(S_x, S_y, S_z) by IM2.

R7: Calculate and output trace of motion for each piece on the ground(S_x, S_y, S_z) by IM3.

[Functional requirements should reference the instance models. —SS] [Reference instance models. —Author]

6.2 Nonfunctional Requirements

Performance is influenced by amount of pieces, capability of GPU and CPU. Other nonfunctional requirements are correctness and reusability.

[More information on the NFRs would improve this document. —SS] [Haven't come up with other good points for NFRs. Will keep thinking ! —Author]

7 Likely Changes

LC1: A5 - In real situation, users may have kinds of terrians and textures. As a result, coefficient of friction needs to be changed correspondingly.

8 Traceability Matrices and Graphs

The purpose of the traceability matrices is to provide easy references on what has to be additionally modified if a certain component is changed. Every time a component is changed, the items in the column of that component that are marked with an "X" may have to be modified as well. Table 4 shows the dependencies of theoretical models, general definitions, data definitions, and instance models with each other. Table 5 shows the dependencies of instance models, requirements, and data constraints on each other. Table 6 shows the dependencies of theoretical models, general definitions, data definitions, instance models, and likely changes on the assumptions.

	T1	T2	DD1	DD2	IM1	IM2	IM3
T1						X	
T2				X			X
DD1				X			X
DD2							
IM1							
IM2							
IM3			X	X			

Table 4: Traceability Matrix Showing the Connections Between Items of Different Sections

	IM1	IM2	IM3	R1	R2	R3	R4	R5	R6	R7
IM1				X				X		
IM2	X							X	X	
IM3				X						X
R1					X				X	X
R2										
R3										
R4										
R5									X	
R6										
R7										

Table 5: Traceability Matrix Showing the Connections Between Requirements and Instance Models

	A1	A2	A3	A4	A5	A6	A7
T1	X						
T2					X		
DD1					X		
DD2					X		
IM1							
IM2		X					
IM3					X		
LC1		X				X	

Table 6: Traceability Matrix Showing the Connections Between Assumptions and Other Items

References

W. Spencer Smith and Lei Lai. A new requirements template for scientific computing. In J. Ralyté, P. Ågerfalk, and N. Kraiem, editors, *Proceedings of the First International Workshop on Situational Requirements Engineering Processes – Methods, Techniques and Tools to Support Situation-Specific Requirements Engineering Processes, SREP'05*, pages 107–121, Paris, France, 2005. In conjunction with 13th IEEE International Requirements Engineering Conference.

W. Spencer Smith, Lei Lai, and Ridha Khedri. Requirements analysis for engineering computation: A systematic approach for improving software reliability. *Reliable Computing, Special Issue on Reliable Engineering Computation*, 13(1):83–107, February 2007.

9 Appendix

9.1 Symbolic Parameters