Chris Kumor

LTU Project

Calculus-Based Simulations

09/20/2021

Blacksmith

Summary:

This simulation is a first-person perspective role-playing game, the player’s role is that of a blacksmith. The blacksmith is given a workshop with various metals, a forge, and a cooler. The current implementation of the simulation allows the blacksmith the heat the metal bars and quench them in the water bath. The heating mechanic is based on Newton’s Law of Cooling and is applied to the blacksmith’s metals and the cooling bath. The blacksmith will be able to see the differences in heating and cooling of the metal bars visually (through cues such as metal incandescence) and a temperature reading given in the User Interface. The picture below shows the most current implementation of this project and what it will look like to the user.

A picture containing text, fire, brick, nature

Description automatically generated

Current Implementation:

In the upper-left and upper-right hand corners of the viewport we can see the blacksmith’s UI. The UI gives the blacksmith’s immediate feedback of the variables in play and provides the user another form of feedback. Towards the center of the screen, we can see the player using a tool to hold a metal bar. The metal bar is shown in the UI to be over 1,000 degrees Celsius in the UI. The metal bar is radiating the color above as an attempt to show a non-UI cue to the player. Without having to focus on the UI the player can be given in-game cues to show off the differences in properties. For example, when compared to other metals iron has a high thermal conductivity. This along with other conditions causes the iron to heat up faster than other metals. This difference is not only represented in the UI but in the in-game mechanics as well. In a “learning by doing” way we can teach people the intricacies behind scientific theorems by showing the ways they can manifest in the activities people carry out in our day-today lives.

Heating Mechanic:

The logic for calculating the temperature of heating/cooling object at t time is stored within a C# static class. When the object in question reaches a state of heating/cooling the heating/cooling function from the static class is called. The following picture is a screenshot of the static class containing the heating/cooling function.

Text

Description automatically generated

From the screenshot we can see that for either the cooling equation or the heating equation the same parameters are passed through. The source temperature, the temperature of the object at time t, the cooling constant, and the name of the object. The cooling constant is calculated in the script attached the object that is being heated/cooled. By keeping it in a static class I can call it from any script which in turns makes programming this mechanic to other objects easier.

Deriving the formula:

Through internet sources such as khan academy, Wikipedia, and academic resources I found Newton’s Cooling defined as:

Schematic

Description automatically generated

dT is the change in Temperature and dt is the change in time, k is the cooling constant, T is the Temperature of the object and T­2 is the temperature of its surroundings.

By applying separation of variables, I derived:

dT/(T-T2) = -k\*dt

Taking the integral of both sides results in:

∫(1/T-T2)dT = -k∫dt

ln│ T-T2│ = -kt + C

The following steps are to derive the equation in terms of T:

eln│T-T2│= e-kt+C

T-T2 = e-kteC

T = T2 + e-kteC

Now we solve for the constant:

@ t = 0, T(t) = Ti, Initial Temperature of object

Ti = T2 + eC

eC = Ti – T2

ln|eC| = ln| Ti – T2|

C = = ln| Ti – T2|

Resulting in:

T = T2 + (Ti – T2)e-kt

Heating function:

DT/dt = -k(T2-T)

Applying separation of variables

DT/(T2-T) = -k\*dt

Taking the integral of both sides:

∫(1/T2-T)dT = -k∫dt

ln│ T2-T │ = -kt + C

The following steps are to derive the equation in terms of T:

eln│T2-T│= e-kt+C

T2-T = e-kteC

-T = e-kteC-T2

T = T2 - e-kteC

Solving for the constant C by evaluating equation @ t= 0, where T(0) = Ti

Ti  = T2-eC

eC= T2 – Ti

ln|eC| = ln| T2 – Ti |

C = = ln| T2 – Ti |

Resulting in:

T = T2 - (T2 – Ti)e-kt

From source (1) the cooling constant k can be defined as the following:

k = K\*A/(m\*s\*d)

K = The Thermal conductivity of the material the object is made of.

A = The Surface Area exposed to the heat.

m = The mass of the object being heated.

s = The specific heat of the object.

d = The thickness of the material.

The cooling constant k of any object is calculated at the start of the applications runtime. A data structure called metalStruct has a float variable for each of these components of the cooling constant. A script that handles the given object is given a reference to an instance of metalStruct that is attached to the object calculates the cooling constant.

Sources:

1. (Source:B.L.Worsnop and H.T.Flint, Advanced Practical Physics for Students Ninth Edition, Macmillan)
2. <https://byjus.com/jee/newtons-law-of-cooling/>
3. vlab.amrita.edu,. (2011). Newton's Law of Cooling. Retrieved 21 September 2021, from vlab.amrita.edu/index.php?brch=194&cnt=1&sim=354&sub=1