Final Project FA21

Duncan McDonald, 12/12/21

Contents

[Project 3](#_Toc90238073)

[Overview 3](#_Toc90238074)

[Framework 3](#_Toc90238075)

[Experience 3](#_Toc90238076)

[Systems 3](#_Toc90238077)

[Producer/Consumer/Decomposer System 3](#_Toc90238078)

[Weather System 4](#_Toc90238079)

[Playtest Sessions 4](#_Toc90238080)

[UML Diagrams 4](#_Toc90238081)

[Intermediate Object-Oriented Programming (C#) 5](#_Toc90238082)

[Concepts 5](#_Toc90238083)

[Separation of Concern 5](#_Toc90238084)

[Polymorphism 5](#_Toc90238085)

[Inheritance (“is a”) 6](#_Toc90238086)

[Containment (“has a”) 6](#_Toc90238087)

[General Association (“uses a”) 7](#_Toc90238088)

[Skills 7](#_Toc90238089)

[Delegate 7](#_Toc90238090)

[Enum 8](#_Toc90238091)

[Exception 8](#_Toc90238092)

[External Data 9](#_Toc90238093)

[Interface 10](#_Toc90238094)

[Research 10](#_Toc90238095)

[Provided Research 10](#_Toc90238096)

[Credits 12](#_Toc90238097)

# Project

## Overview

A simulation of the Bracken Cave Preserve near San Antonio, Texas that houses approximately 15 million Brazilian free-tailed bats (*Tadarida brasiliensis*). Although the inspiration for this simulation is based on actual producers, consumers, and decomposers, some decisions were made in abstraction so that this project could be completed in the 4 week timeframe.

## Framework

My framework for this assignment was similar to the way I approach any program that emulates a real world action, I identify the actions within the overall goal and I break them out into smaller code components. This allowed me to break up components of the ecosystem, weather and item trading in a logical manner. This was incredibly valuable when trying to develop complex interwoven systems.

## Experience

My experience building this program was an excellent learning and explorative experience. This type of simulation is something I find very interesting given my infatuation with the natural world and my natural curiosity towards anything related to game systems. This project felt like an excellent stepping stone into thinking about the ecosystems found in games. That being said, there were a few moments of frustration caused by errors that I was not identifying at first. After taking a step back I managed to identify the issues I was dealing with and apply fixes.

# Systems

## Producer/Consumer/Decomposer System

The producer/consumer/decomposer system is the most intricate of the simulation and drives most of the functionality. Producers generate new producers each turn based on the creation of spores and the availability of nutrient rich soil. This is a foundational block of the system that allows the rest of the ecosystem to flourish. Worms eat producers and are in turn consumed by bats which are consumed by hawks. Each level of consumer keeps the previous in check and they are all kept in check by the availability of producers. These consumers are also vital to the ecosystem because they produce biotic remains in the form of Guano and other various biotic remains. These biotic remains are generated by consumers dying naturally or through predation. The biotic materials produced by these interactions are converted into soil nutrient by the available decomposers. Each aspect of the ecosystem ties into the rest and every component is essential to long term growth.

## Weather System

The other system at work in this simulation is the current weather. Weather has an overarching impact on the speed of propagation within the ecosystem. Colder temperatures will slow the production of new entities while hotter temperatures will hasten ecosystem production. The initial temperature is read into the program by an API but can be altered by player powerups.

# Playtest Sessions

An ideal opportunity to gain feedback on this project was during our playtest sessions. During these sessions, I got to observe users interacting with my program to identify shortcomings. The first playtest allowed me to make changes regarding the pace of the timer and overall aesthetic. This also served as a great opportunity to observe the programs made by other classmates and gain both inspiration and insight. During the playtests I noticed a distinct lack of organization within the UI, I alleviated this issue somewhat but UI formatting was not my priority with this project.

# UML Diagrams

A picture containing text, indoor, wall, screenshot

Description automatically generated

# Intermediate Object-Oriented Programming (C#)

## Concepts

### Separation of Concern

Definition: The separation of code components so individual components have an individual concern.

Code excerpts from your project:

public void ShowData()

{

CottonWormCount.Text = FindEntity(ecosystem.entities, "Cotton Bollworm").Count.ToString();

CornWormCount.Text = FindEntity(ecosystem.entities, "Corn earworm").Count.ToString();

CornCount.Text = FindEntity(ecosystem.entities, "Corn").Count.ToString();

CottonCount.Text = FindEntity(ecosystem.entities, "Cotton").Count.ToString();

BatCount.Text = FindEntity(ecosystem.entities, "Brazilian free-tailed bat").Count.ToString();

HawkCount.Text = FindEntity(ecosystem.entities, "Red-tailed hawk").Count.ToString();

SoilCount.Text = FindEntity(ecosystem.entities, "Soil").Count.ToString();

BioticCount.Text = (FindEntity(ecosystem.entities, "Guano").Count + FindEntity(ecosystem.entities, "Biotic").Count).ToString();

BeetleCount.Text = (FindEntity(ecosystem.entities, "Dermestid beetle").Count + FindEntity(ecosystem.entities, "Guano Beetle").Count).ToString();

CurrentDay.Text = $"Day: {currentDay}";

}

public static Entity FindEntity(List<Entity> entities, string name)

{

Entity p = null;

foreach (Entity i in entities)

{

if (i.Name == name)

{

p = i;

}

}

return p;

}

Explain usage in your project:

The example above shows separation of concern between the method writing text to textboxes and the method that uses logical operators to find the right content to print.

### Polymorphism

Definition: The ability of code components to behave differently in different situations.

Code excerpts from your project:

public virtual void Growth(int weatherImpact)

{

count += (count / weatherImpact);

}

public override void Growth(int weatherImpact)

{

int breedingPairs;

breedingPairs = Count / 2;

Count += breedingPairs / random.Next(1,4) / weatherImpact;

}

public override void Growth(int weatherImpact)

{

int sporeCreation;

sporeCreation = Count \* 5;

Count = Count + sporeCreation / weatherImpact;

}

Explain usage in your project: I used polymorphism to allow different entities to grow in different ways.

### Inheritance (“is a”)

Definition: Inheritance allows classes to inherit properties or methods from a parent class.

Code excerpts from your project:

class Person : Entity

{

public List<Powerup> powerups;

}

class Shop : Person

{

}

class Player : Person

{

}

Explain usage in your project: Inheritance is used abundantly throughout the project to allow objects to behave in similar ways.

### Containment (“has a”)

Definition: Instead of inheriting properties form a class, a class might contain an instance of an object.

Code excerpts from your project:

class Ecosystem

{

public TemperatureReading currentTemperature = CurrentTemperature();

public List<Entity> entities = LoadEntities("../../data/entities.xml");

public EnvironmentStatus status = EnvironmentStatus.Balanced;

Explain usage in your project: In my project the ecosystem has a weather system and a list of entities. These are a part of the ecosystem class but they originate elsewhere in the code.

### General Association (“uses a”)

Definition: General association is when a class can use data or information from a class.

Code excerpts from your project:

Ecosystem ecosystem = new Ecosystem();

Player player = new Player();

Shop shop = new Shop();

private void InitializeShop() => shop.powerups = new List<Powerup>

{

new Powerup("Rejuvenate","Replenishes Ecosystem", "Biotic", 1000),

new Powerup("Eviscerate","Destroys Ecosystem", "Red-tailed hawk", 100),

new Powerup("Renew Predators","Replenish the predators of the ecosystem", "Dermestid beetle", 100),

new Powerup("Winter Frost","Make the weather colder", "Corn earworm", 1000),

new Powerup("Summmer Heat","Make the weather warmer", "Cotton Bollworm", 1000),

new Powerup("Blank","Blank","Blank", 0)

};

Explain usage in your project: In the example above, the simulation has a player and shop but a player or shop can use powerups to effect the simulation.

## Skills

### Delegate

Definition: A type that represents references to methods with particular parameters and return type.

Code excerpts from your project:

private void Counter()

{

//if you want to hide something while

//the counter is counting down

//one way is to modify the Visibility property

//btnNextDay.Visibility = Visibility.Collapsed;

//DispatchTimer example by kmatyaszek (https://stackoverflow.com/users/1410998/kmatyaszek)

timeSpan = TimeSpan.FromSeconds(5);

timer = new DispatcherTimer(

new TimeSpan(0, 0, 1),

DispatcherPriority.Normal,

delegate

{

//CurrentDay.Text = timeSpan.ToString("c");

ShowData();

if (timeSpan == TimeSpan.Zero)

{

timer.Stop();

GameLoop();

//Show the "next day" button

//btnNextDay.Visibility = Visibility.Visible;

//call game loop

}

timeSpan = timeSpan.Add(TimeSpan.FromSeconds(-1));

},

Application.Current.Dispatcher);

timer.Start();

}

Explain usage in your project: In this project, a delegate is used to operate the counter functionality.

### Enum

Definition: a value type defined by named constants tied to an integral numeric type.

Code excerpts from your project:

enum EnvironmentStatus

{

Balanced,

Unbalanced,

Critical

}

class Ecosystem

{

public TemperatureReading currentTemperature = CurrentTemperature();

public List<Entity> entities = LoadEntities("../../data/entities.xml");

public EnvironmentStatus status = EnvironmentStatus.Balanced;

Explain usage in your project: An enum was used in this project to set three easily usable statuses. I don’t want the status to be anything beside these three types so an enum makes sense.

### Exception

Definition: An error that occurs during runtime.

Code excerpts from your project:

try

{

temp.Count = Int32.Parse(entity.GetAttribute("count"));

temp.Dailyintake = Convert.ToInt32(entity.GetAttribute("dailyintake"));

}

catch (Exception)

{

temp.Count = 0;

temp.Dailyintake = 0;

}

Explain usage in your project: In this project I implemented a try/catch to handle any exceptions that may arise during data load.

### External Data

Description: XML data

Code excerpts from your project:

public static List<Entity> LoadEntities(string fileName)

{

List<Entity> entities = new List<Entity>();

if (File.Exists(fileName))

{

XmlDocument doc = new XmlDocument();

doc.Load(fileName);

XmlNode root = doc.DocumentElement;

XmlNodeList entityList = root.SelectNodes("/environment/entity");

foreach (XmlElement entity in entityList)

{

Entity temp;

if (entity.GetAttribute("type") == "Producer")

{

temp = new Producer();

}

else if (entity.GetAttribute("type") == "Consumer")

{

temp = new Consumer();

}

else if (entity.GetAttribute("type") == "Decomposer")

{

temp = new Decomposer();

}

else if (entity.GetAttribute("type") == "Player" || entity.GetAttribute("type") == "Vendor")

{

temp = new Person();

}

else

{

temp = new Entity();

}

temp.Name = entity.GetAttribute("name");

temp.Species = entity.GetAttribute("species");

try

{

temp.Count = Int32.Parse(entity.GetAttribute("count"));

temp.Dailyintake = Convert.ToInt32(entity.GetAttribute("dailyintake"));

}

catch (Exception)

{

temp.Count = 0;

temp.Dailyintake = 0;

}

temp.Foodsource = entity.GetAttribute("foodsource");

temp.Identity = entity.GetAttribute("identity");

entities.Add(temp);

}

//Add count load in as an integer.

//Add foodsource load.

}

return entities;

}

Explain usage in your project: I used an XML document to load in the ecosystem entities. This was very helpful given I had to change the starting counts a few times.

### Interface

Definition: an interface can apply shared methods to classes that already have a parent.

Code excerpts from your project:

class Bat : Consumer, Guano

{

int Guano.GuanoMultiplier()

{

return Count;

}

}

interface Guano

{

int GuanoMultiplier();

}

Explain usage in your project: I used an interface to provide bats and hawks a method to add guano production. I did not want worms to do this which is a consumer so it had to be implemented as an interface.

# Research

## Provided Research

*A guide to the micro moths of Austin and Travis County, Texas* · iNaturalist. (2021). Retrieved 17 November 2021, from <https://www.inaturalist.org/guides/5062>

*Ann W. Richards Congress Avenue Bridge* - Wikipedia. (2021). Retrieved 17 November 2021, from <https://en.wikipedia.org/wiki/Ann_W._Richards_Congress_Avenue_Bridge>

*Bats are Austin's favorite neighbors!* (2018). Retrieved 17 November 2021, from <https://www.youtube.com/watch?v=OlzAb4Jg4cA>

*Bracken and its Bats: A Natural Wonder of the World* - Bat Conservation International. (2021). Retrieved 17 November 2021, from <https://www.batcon.org/bracken-and-its-bats-a-natural-wonder-of-the-world/>

*Brazilian Free-Tailed Bats - Congress Avenue Bridge Bat Colony — Congress Ave. Bridge Bats*. (2021). Retrieved 17 November 2021, from <https://www.austinbats.org/bats>

*Calculate the Value of Bats.* (2021). Retrieved 17 November 2021, from <https://www.fs.usda.gov/Internet/FSE_DOCUMENTS/fseprd476773.pdf>

Cleveland, Cutler J., Margrit Betke, Paula Federico, et al. *Economic value of the pest control service provided by Brazilian free-tailed bats in south-central Texas.* [https://doi.org/10.1890/1540-9295(2006)004[0238:EVOTPC]2.0.CO;2](https://doi.org/10.1890/1540-9295(2006)004%5b0238:EVOTPC%5d2.0.CO;2). First published: 01 June 2006. Retrieved 17 November 2021, from <https://esajournals.onlinelibrary.wiley.com/doi/10.1890/1540-9295%282006%29004%5B0238%3AEVOTPC%5D2.0.CO%3B2>.

*corn earworm - Helicoverpa zea.* (2021). Retrieved 17 November 2021, from <https://entnemdept.ufl.edu/creatures/veg/corn_earworm.htm>

*Dermestidae* - Wikipedia. (2021). Retrieved 17 November 2021, from <https://en.wikipedia.org/wiki/Dermestidae>

*Guano* - Wikipedia. (2021). Retrieved 17 November 2021, from <https://en.wikipedia.org/wiki/Guano>

*Jacobsoniidae* - Wikipedia. (2021). Retrieved 17 November 2021, from <https://en.wikipedia.org/wiki/Jacobsoniidae>

*Journeying With Bats Across Mexico*. Perpetual Planet: Mexico. National Geographic. (2018).

Maine, Josiah J., and Justin G. Boyles. *Bats initiate vital agroecological interactions in corn* (2021). Retrieved 17 November 2021, from <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4603461/>

*Mexican free-tailed bat* - Wikipedia. (2021). Retrieved 17 November 2021, from <https://en.wikipedia.org/wiki/Mexican_free-tailed_bat>

*Red-tailed Hawk Identification*. All About Birds, Cornell Lab of Ornithology. (2021). Retrieved 17 November 2021, from <https://www.allaboutbirds.org/guide/Red-tailed_Hawk/id>

*Standards and Medical Management for Captive Insectivorous Bats*. *Chapter Seven: Feeding Adult Bats* (2021). Retrieved 17 November 2021, from <https://batworld.org/wp-content/uploads/2013/08/Chapter7-feeding-adults.pdf>

*The Bat Bridges of Austin, Texas*. Urban Nature. (2021). Retrieved 17 November 2021, from <https://interactive.wttw.com/urbannature/bat-bridges-austin-texas#!/>

*The Truth About Bats*. Wandering Path Productions. (2020). Retrieved 17 November 2021, from <https://youtu.be/kW--eOLOrq0>

*The world's largest mammal migration*. DW Documentary. (2020). Retrieved 17 November 2021, from <https://youtu.be/TIr0xKk2YJ0>

*This is the Largest Mammal Migration in the World*. Smithsonian Channel. (2019). Retrieved 17 November 2021, from <https://www.youtube.com/watch?v=lFrB9zNkVcw>

*UC IPM: UC Management Guidelines for Cotton Bollworm on Cotton*. (2021). Retrieved 17 November 2021, from <http://ipm.ucanr.edu/PMG/r114300511.html>

# Credits

*Credit to Janell Baxter for in class examples of delegates, API calls and Event handlers. All other code was conceptualized and implemented by me.*

*From the syllabus:* ***“All work submitted in this course for academic credit must be your own original work, the original work of the group of students cooperating in a project (and appropriate credit given), and/or adhere to all relevant copyright and intellectual property ownership laws.***

***Academic integrity is giving credit to the ideas, research, and creations of others; and part of one’s education is learning how to give this credit.****”*