CS 162 - Proj2: Zoo Tycoon - Design + Reflection Document
Jeovani Vela
2/4/18

Project 2 was based on creating a game called Zoo Tycoon. User buys animals to start the game, feeds them daily and has random events occur everyday which have an affect on the outcome of the zoo. Additionally, this project requires utilizing multiple classes with inheritance playing a major role on how these various classes are used.

Requirements:

- Animal Class with the following member variables

Age
- Adult if age >= 3 days
- Baby if age < 3 days

Cost
- Tiger cost $10,000
- Penguin cost $1,000
- Turtle cost $100

Number of Babies
- Tigers have 1 baby
- Penguins have 5 babies
- Turtles have 10 babies

Base Food Cost
- You can get this base food cost from the user or set it as a constant. Example base food cost per animal per day: $10.
- Tigers have a feeding cost of 5 times the base cost
- Penguins have a feeding cost that is the same as the base cost
- Turtles have a feeding cost that is 50% the base cost

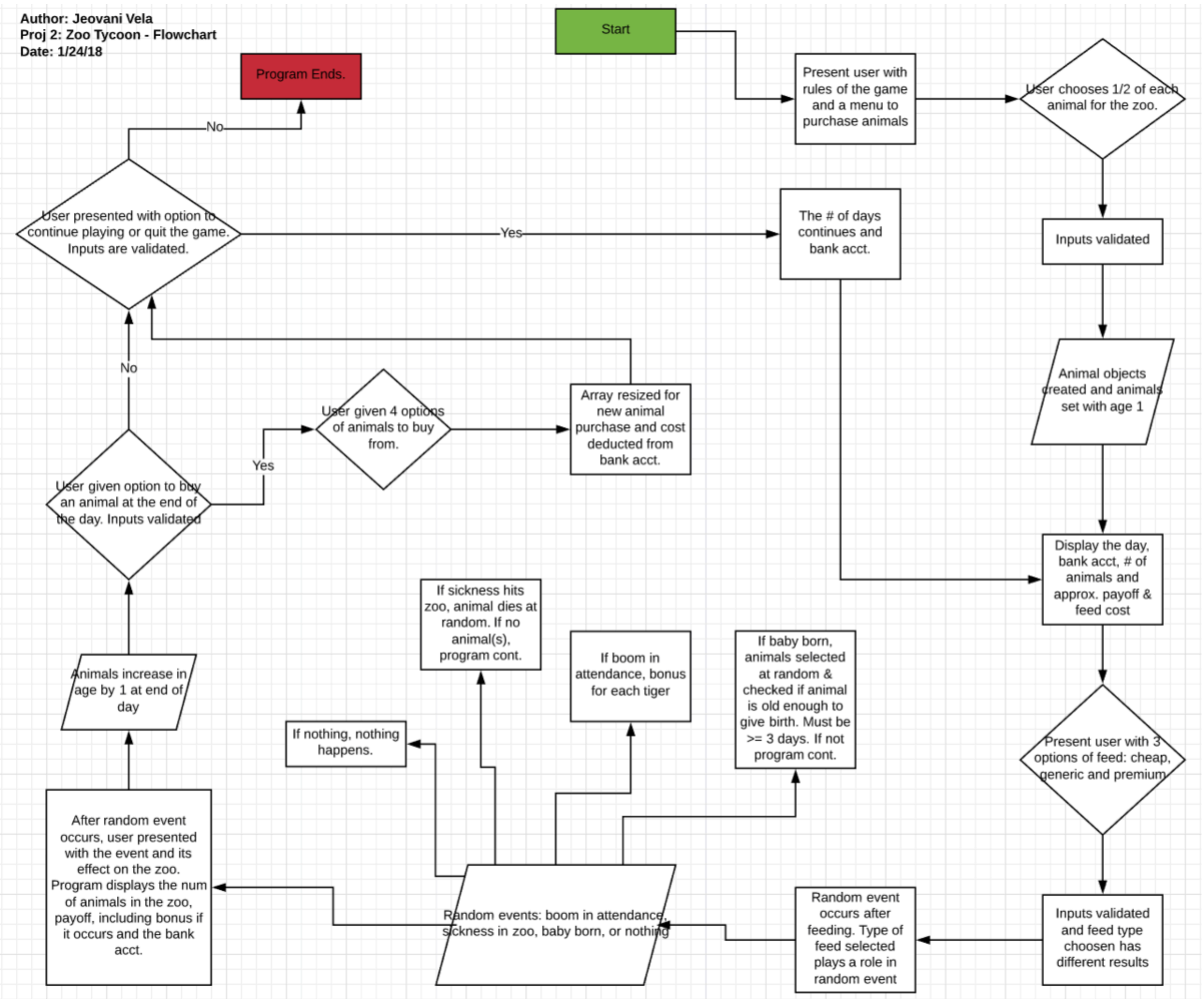
Payoff
- A tiger's payoff per day is 20% of their cost per animal. (not counting bonus)
- A penguin's payoff per day is 10% of their cost per animal
- A turtle's payoff per day is 5% of their cost per animal

-The Tiger, Turtle and Penguin class must inherit from the Animal class.
-User starts with 100K and must purchase 1 or 2 of each animal to start.
-Must feed the animals everyday. Random Event must occur after feeding.
-Random events may include:
1. -Sickness at the zoo. Animal chosen at random and dies.
2. -Birth at the zoo. Animal chosen at random, checked for age and gives birth to certain # of babies.
3. -Boom in attendance. If so, $250-500 bonus for each Tiger.
4. -Nothing happens.

-Random event may or may not have an affect on the zoo.
-User inputs must be validated.
-Payoff calculated at the end of the day.
-User asked to buy an animal at the end of day and if he/she would like to continue playing.

Design:

I first created the Animal class in order to make sure the member variables to be used are initialized correctly. I also included member functions that would need to be inherited by the other classes, namely the tiger, turtle, penguin and longhorn class. I decided to add another animal to my zoo and thus give the game another element. After the Animal class was created, I began to create the classes that would be deemed the derived classes. With a majority of the member functions being inherited, these classes mainly consisted of the functions that are not otherwise inherited. Such as the constructor, copy constructor, and the destructor. I also included an overloaded assignment function in order to avoid any deficiencies when I assign one object to another. After the derived classes were created, I began working on my Zoo class. This class would contain all the functions necessary to run the game. Including some member variables that will carry values needed to keep track of certain aspects of the game. It will first present the user with rules followed by a menu of animals to choose from and buy. It will then run a typical day at the zoo which includes displaying the user the bank account amount for the day and number of animals in the zoo followed by a menu to purchase certain feed for the animals. User can choose from cheap, generic and premium feed. Some food choices can play a role on the random effect that occurs after feeding. After feed is selected and validated a random event occurs. I included functions to check whether certain animals could be affected by the random event in order to prevent the program for crashing. This includes checking for the age of animal before giving birth or checking if a certain animal is available before being inflicted with a sickness and passing away. Each random event triggers functions that will also re-size the array, deallocate the old array and dynamically allocate a new array. These function calls are in various other functions depending on the random event that will occur. After the random event occurs and the effect has taken place, all animals will increase in age by one at the end of day. Also the user will be asked whether to buy another animal or not before being prompted again if they want to continue playing. If an animal is bought, the cost is deducted from the bank account. If they decided to continue playing, the amount in the bank carries over, as do the number of animals and the number in days will change as well. Program will only terminate at the end of the day when the user is prompted to continue playing or quit.


Author: Jeovani Vela
Proj 2: Zoo Tycoon - Flowchart
Date: 1/24/18
Start
Program Ends.
Present user with rules of the game and a menu to purchase animals
User chooses 1/2 of each animal for the zoo.
No
User presented with option to continue playing or quit the game. Inputs are validated.
Yes
The # of days continues and bank acct.
Inputs validated
No
Animal objects created and animals set with age 1
User given 4 options of animals to buy from.
Array resized for new animal purchase and cost deducted from bank acct.
Yes
User given option to buy an animal at the end of the day. Inputs validated
Display the day, bank acct, # of animals and approx. payoff & feed cost
If sickness hits zoo, animal dies at random. If no animal(s), program cont.
If boom in attendance, bonus for each tiger
If baby born, animals selected at random & checked if animal is old enough to give birth. Must be >= 3 days. If not program cont.
Animals increase in age by 1 at end of day
If nothing, nothing happens.
Present user with 3 options of feed: cheap, generic and premium
After random event occurs, user presented with the event and its effect on the zoo. Program displays the num of animals in the zoo, payoff, including bonus if it occurs and the bank acct.
Random events: boom in attendance, sickness in zoo, baby born, or nothing
Random event occurs after feeding. Type of feed selected plays a role in random event
Inputs validated and feed type choosen has different results

Test Plan:

Test Case	Input Values	Driver Functions	Expected Outcomes	Observed Outcomes
Start screen - user given option to buy 1/2 tigers	3dfg,hsgs, [][34], 00, -1, 3, a341, 124,u6h,9035, 1	Input validator will only accept a 1 or 2	User will get message stating input is incorrect and choose either 1 or 2.	Program does not continue until user enters correct number, 1 or 2. Program continues after user inputs 1.
Start screen - user given option to buy 1/2 turtles	-1, 342dfas., 4gfsg, [;1., 44, 2	Input validator will only accept a 1 or 2	User will get message stating input is incorrect and choose either 1 or 2.	Program does not continue until user enters correct number, 1 or 2. Program continues after user inputs 2.
Start screen - user given option to buy 1/2 penguins	111, -0. dafdsf., rr333, v, -3, -2, -1, 01	Input validator will only accept a 1 or 2	User will get message stating input is incorrect and choose either 1 or 2.	Program does not continue until user enters correct number, 1 or 2. Program continues after user inputs 01.
Start screen - user given option to buy 1/2 longhorns	23, trwe,]01, ! nvww, 1!?, add, 02	Input validator will only accept a 1 or 2.	User will get message stating input is incorrect and choose either 1 or 2.	Program does not continue until user enters correct number, 1 or 2. Program continues after user inputs 02.
feeding time - user given 3 options to buy feed: 1.cheap, 2. generic or 3. premium	4, 12dfsa, pp., 0, 12, fdsg, , 1214zzf, 4, 09, -1, 3	Input validator will only accept a 1, 2 or 3.	User will get message stating input is incorrect and choose either 1, 2, or 3.	User receives error message until a number 1-3 is chosen. 3 is selected and random event - baby is born occurs.
Random event occurs - baby animal is born, but no animal is of age at start of game	Input is 3 from the feed selection - premium feed	Baby born event triggers for animal to be checked if they are old enough.	Program will check if their is animal old enough to have a baby. If no animal is old enough a message is returned and game continues.	No animal was old enough on day 1 and program responds by sending message that there is no animal old enough and program continues.
Random event occurs - sickness hits the zoo and random animal chooses is not available	Input is 1 from the feed selection - cheap feed	Sickness event triggers for the animal to be randomly selected to get sick and die. Checks if animal is available before sickness occurs and animal dies.	Program will randomly choose an animal to get sick and die. If animal is not available, program advises user and program continues.	Tiger is randomly selected to die. There are no tigers, program informs user and program continues to work.
End of the day - user is asked to buy an animal. Y/N	1, fdgsfd., ; [];. 121., fg, y	Program will validate the input before proceeding. User chooses y/n.	Program will not continue unless 'y' or 'n' is entered.	Program progresses after 'y' is entered and user presented with a menu with the animals to choose from to buy.
End of day - user presented with a menu to buy 1 out of 4 animals	dda4, 435, 0., -111, fdsg2432, [] [s, fdg, 3	Program will validate input before proceeding and dedicating the cost from the bank.	User will get message stating input is incorrect and choose either 1, 2, 3, or 4.	Program continues after uses chooses correct input, 3. Penguin is bought and program continues.
End of day - user is asked to continue playing or quit.	453,fsg, 123, af4324, 1[].>, y	Program will validate the input before proceeding. User chooses y/n.	User will get message stating input is incorrect and choose either 'y' or 'n'	Program continues after user selects 'y'. Game continues.

Reflection:

This project was tedious to build however it gave me the task of truly understanding how inheritance works and how it can be beneficial to writing code. Having member functions other than those that cannot be inherited (i.e. constructor(s), destructor, etc) does make for writing the derived classes more efficient. I did struggle in some aspects however as I built the program. It involved utilizing more private member variables in the Zoo class than I expected as I kept running into the issue of having some variables become out of scope because I was defining them in the function alone and thus could not have their values traverse through the program. For example, I had to make the number of days to be displayed as a member function as it did not work for me to declare it in the function and use a loop. Another issue I ran into was increasing the age of my animals. I was placing my increase age function in wrong areas throughout my Zoo class which was not enabling the number of days to increment along with the game. I also struggled with my random events. I use utilizing just rand() however I kept getting the same random number after each iteration. This involved me learning how to have a different random number appear every time. I learned I needed to include the srand((unsigned)time(0)) snippet in my code. This allowed my program to give me a different random number every time and thus helped me when a random animal was chosen at well. Putting this program together helped me close some of the learning gap from the book and gave me confidence into the idea of inherency and how it can enable me to become a better programmer.