Sangyun Lee

Leesa6@oregonstate.edu

Oct 22 2019

Project 4

About - Create a zoo tycoon game using classes and inheritance. This game allows users to run a zoo business. Different types of animals cost different prices, have different maintenance costs, and of course, return a different profit at the end of each day.

Requirements

- Get users' response and validate it
- Have animal class which is base class and inherit its function derive class
- Create dynamic array for game play
- Random event
- Different cost and profit for each animal

Challenges

- How to initialize the dynamic array and expand its size at runtime
- How to create the base class for other derive class
- How to generate random event
- How to differentiate the cost and profit

How to solve the requirement and challenges

- Modify and re-use previous menu & getInput function
- Since it doesn't require any override functions, so simply create parent class and bring those variable and set up on child class
    - Ex) class tiger : public Animal

- Generate random event, I decide to perform it, by simply using srand() function. And name the event using enum data type, so it can be understand easily.
    - Ex) enum zEvent {sick = 1, baby, bood, nothing} / enum AnimalType {tiger = 1, penguin, turtle}
- Since derive class' object has own constructor, simply set up with different value.
- Create dynamic array, at the beginning it was designed for 2d array, but later, each animal has each dynamic array. And to create new array, delete previous one, and make new one.
    - Ex) tiger = new tiger[size]
    - Temp[i] = tiger[i].getAge()
    - And double the array size then create new array
    - Tiger = new tiger[new size] …..

Test Table

1. Game Start – it is to check whether the game is started well and is exited well based on user input

Test Case	Input	Driver Func	Expected Output	Actual Output
Wrong	'-3'	Menu() & getInput()	"your input is wrong…."	"your input is wrong…."
Wrong	'f'	Menu() & getInput()	"your input is wrong…."	"your input is wrong…."
Correct	1	Menu() & getInput()	Return 1, and "welcome to…"	Return 1, and "welcome to…"
Correct	2	Menu() & getInput()	Return 2, and close the program	Return 2, and close the program

2. Buy Animal – when game started, user have to buy each animal: tiger, penguin and turtle, one or two of each of them.

Test Case	Input	Driver Func	Expected Output	Actual Output

Wrong	‘0’	buyHowmany()	“your input is wrong….”	“your input is wrong….”
Wrong	‘c’	buyHowmany()	“your input is wrong….”	“your input is wrong….”
Correct	2	buyHowmany()	“you entered 2..”	“you entered 2..”
Correct	2	buyHowmany()	“you entered 2..”	“you entered 2..”

3. addAnimal – it adds animals into array. To check this function, use it with getPop() for each animals/ *This function works with buyAnimal(), and I pre-coded animal type, so users can decide integer only

Test Case	Input	Driver Func	Expected Output	Actual Output
correct	‘t’, 2	addAnimal () & getTpop	2	2
Correct	‘p’ , 2	addAnimal () & getTpop()	2	2
Correct	‘r’, 2	addAnimal () & getTpop()	2	2

4. buildCage() – this function is to expand dynamic array when it needs.

Test Case	Input	Driver Func	Expected Output	Actual Output
Penguin > 10	‘r’ totalPpop	buildCage()	New cage is built	New cage is built
Tiger < 10	‘t’ totalTpop	buildCage()	No output	No outPut

5. setZevent() – This function generate randome events for the zoo game. Since it is random function, there is no expected out. But will see whether it has a pattern or not. *I am writing the first event only, so I can check it created ‘random’ event

Test Case	Input	Driver Func	Expected Output	Actual Output
Day 1	‘no input’	setZevent()	One of events For one of animals	No event

Day 1	'no input'	setZevent()	One of events For one of animals	Tigers are too young to have babies
Day1	'no input'	setZevent()	One of events For one of animals	Your penguin got sick

6. buyAdult() – User allow to buy an adult animal which is 3days old. To test it out, I controlled the event, so I can check animal can have a baby or not

Test Case	Input	Driver Func	Expected Output	Actual Output
Tiger	One adult tiger	buyAdult()& setZevent()	Congratulation!, your tiger….	Congratulation!, your tiger….
Penguin	One adult penguin	buyAdult()& setZevent()	Congratulation!, your penguin….	Congratulation!, your penguin….
Turtle	One adult turtle	buyAdult()& setZevent()	Congratulation!, your turtle….	Congratulation!, your turtle….

7. playGame() – This function is to play the zoo game. All functions are ran in this function. I want to check, that this program is kept playing game or is closed upon users' selection

Test Case	Input	Driver Func	Expected Output	Actual Output
Wrong input	-1	Menu() & playGame()	"Your input is wrong…"	"Your input is wrong…"
Correct, keep playing	1	Menu() & playGame()	Start next day game and display output	Match with expected output
Correct, end game	2	buyAdult()& setZevent()	"Thank you for playing this game.."	"Thank you for playing this game.."

Reflection

This assignment was difficult for me to get to the right track, at the beginning, I thought it will be good to have 2d dynamic array, because animal types can be row, and number of animal can be column. But, double the size was not easy with 2d array. Because I had to find the way to make the specific array larger. Of course, I could index the row and column. However, I decide to go with easy way so that I changed my code from 2d array to single dynamic array, and even instructor said it should be initialized as dynamic array, not 2d dynamic array. Therefore, if its size need to be expanded, I simple passed the animal type then, only the animal type's array is doubled. In order to implement that I move array values into temp array, and created new doubled array and bring the data back into it. So, I can control each array size.

And, this time, I used srand() for generating random event. I used the rand(), but when I tested the program few times, I realized the random numbers from rand() has a certain pattern. So events were made the same order. So, I was possible to anticipate what the next event will be. To sovle this issue, I used srand() with time, so it could have random seed. After change it, events were more likely 'random event.

I intend to create a single function with multiple purpose which is not good. Rule of thumbs, OOP should have a single purpose of function. So, I was trying to chop it and created a function that perform a one purpose. It seems that there are some improvement that I can make. But, I can see I am one the way to the right way. One of the benefit of split the function was that I was able to find and fix an error of my program. I will spend more time to design planning phase.