Design Project - Object Orientated Programming

Project idea: Movie Simulator
Group: Nathan Shepherd, Adam Tonkin

Description ~

The main focus of our code is to create a simulation experience of going to the movies. The user will be able to purchase tickets, shop items, and experience a simulation of going to the cinemas, along with all the fun experiences of munching on all your snacks to spontaneously having to go to the bathroom!

Code structure flow (main code) ~

- Initialisation Section
  - **~ Default Information**
  - **String cinema_name =** "Shepherd and Ward Cinemas, Adelaide"
  - Generate food objects for shop

  - **~ Player Information**
  - Generate Player class
  - Input information at the start of program
    - Print menu information and request user input
    - "Welcome to Shepard and Ward Cinemas, Adelaide!"
    - "Thanks for signing up to our mandatory customer rewards program! What is your name?"
    - => User input for name into Player Class via GetName() function.
    - "And Age?"
    - => User input for age (check for valid age (>0) into Player Class via GetAge() function.
  - Run GenerateHunger and GenerateMoney function to populate Money and Hunger

  - **~ Movie Information**
  - Create 3 movie pointers
    - Name: "Death Blow: All American Action", Value: $14, Rating R18+
    - Name: "My Dear Why Do You Cry So?", Value: $13, Rating: M
    - Name: "The Instantly Forgotten Story", Value: $9, Rating PG
  - Create Movie type array to store these movies in

- Cinema Menu Section
    - "What would you like to do?"
    - "1. Choose Movie"
    - "2. Visit Candy Bar"
    - "3. Enter Theatre"
    - "4. Leave"

        - User input selection =>

If a user requests to look at Choose Movie options…

- Scan inventory for current tickets and money and display that info above menu
- New menu will generate with a choice of three movies (print from movie classes)
    - “1. Death Blow: All American Action - $14, R18+”
    - “2. My Dear Why Do You Cry So? - $13, M”
    - “3. The Instantly Forgotten Story - $9, PG”
    - “4. Leave Ticket Menu”
    -
    - “Which of these tickets would you like to buy?: ” User input =>
- If a ticket is selected…
    - “How many would you like?: “ => User input
        - Make sure <=0 otherwise rerequest
    - Add to player TicketInventory and deduct cost
    - “Thankyou for your purchase! :)
    - (Press enter to return to main menu)” => Pause
    - Return to menu
- If leave ticket menu is selected...
    - Return to menu

If a user requests to look at Visit Candy Bar options…

- “Candy Bar Menu Options ~:
- “1. *print pointer name* “Print pointer item value*”
- “2. *print pointer name* “Print pointer item value*”
- “3. *print pointer name* “Print pointer item value*”
- “4. *print pointer name* “Print pointer item value*”
- “5. *print pointer name* “Print pointer item value*”
- “6. *print pointer name* “Print pointer item value*”
- “7. Leave Candy Bar”
- The user can buy items which have a hidden satisfaction rating, it will increase or lower the player’s hunger.
- Similar buying method to movie but with food objects
- The string of noise the player makes will be produced when they select the item

If a user requests to look at Enter Theatre options…

- They will be sent to the movie do movie sequence, (check for and remove ticket etc. if the ticket isn’t there then they are sent back to the buy ticket screen.)

Cinema Dialogue (Press enter to go through the Dialogue)

- “You find your seat. After the infinite amount of ads, 1-Hour later…”
- “*movie noises*... *gasp from the crowd*...”
- “*hunger noises intensifies*”
    - Cycle through entire Food Inventory
        - If food item
            - “*name* eats *food item*”
            - ConsumePrint() noise from food item class

                - EatItem() player function
            - If drink item
                - “*name* drinks *drink item*
                - ConsumePrint() noise from food item class
                - EatItem() player function
            - If no items
                - “But there’s nothing to eat…”
- There will be an intermission prompt where a bathroom or candy bar break is offered. If Candy Bar is selected they’re sent to Candy bar menu Boolean (in movie is triggered)
    - This can add additional items to the player’s FoodInventory
- If the Bathroom is selected prompt relief string “Ahhhhhh”.
- Once they return to the movie from the candy bar prompt string “2-Hours later…”
- “*hunger noises intensifies*”
    - Cycle through entire Food Inventory
        - If food item
            - “*name* eats *food item*”
            - ConsumePrint() noise from food item class
            - EatItem() player function
        - If drink item
            - “*name* drinks *drink item*
            - ConsumePrint() noise from food item class
            - EatItem() player function
        - If no items
            - “But there’s nothing to eat…”
- *surprised crowd noises and cheering*
- *credits rolling*
- “That was a good movie!”
- “Are you ready to leave? (Y/N): “
    - If yes, do leave sequence
    - If no, go back to menu and reset any necessary dialogue variables

If a user requests to leave...

- The pointers that had been created during the program will be deleted.
- “Thanks for visiting Shepherd and Ward Cinemas, Adelaide! Please come again!”
- They will leave the cinema and end the code..

Class contents ~

Public info = Red
Private info = Green
Protected info = Blue

- Player
  - Variables
    - Name (string)
    - Age (int)
    - Hunger rating (int)
    - Money (double)
    - Food Inventory (string vector)
    - Ticket Inventory (string vector)
  - Player (constructor)
    - Includes blank name, age, date, hunger, money, and inventory
  - SetName (string mName)
    - Sets Name
  - SetAge (int mAge)
    - Sets Age
  - GetName ()
    - Returns Name
  - GetAge ()
    - Return Age
  - GetHunger ()
    - Returns Hunger
  - GetMoney ()
    - Returns Money
  - AddFood void function (string mFood)
    - Return item to FoodInventory
  - AddTicket void function (string mTicket)
    - Return item to TicketInventory
  - GenerateHunger int function
    - Generate a random number between 0-10
    - Return Hunger rating
  - GenerateMoney int function
    - If younger than 15
      - Generate money between 10-20 dollars
      - Return Money
    - If between 15 and 18
      - Generate money between 20-40 dollars
      - Return Money
    - If between 19 and 25
      - Generate money between 40-100 dollars
      - Return Money
    - If greater than 25
      - Generate money between 1000-5000 dollars (essentially infinite money)
      - Return Money
  - TicketCheck bool function (string name)

        - For loop through TicketInventory
            - If (mTitle == current looped title) (using strcomp)
                - Remove ticket from ticket inventory
                - Vector erase ticket from ticket inventory
                - Return true
        - If not in inventory, return false
    - EatItem void function (string item)
        - Scans the FoodItem vector for item string
        - If in inventory
            - Restore/Deplete hunger
            - Remove item
            - Return
        - Else
            - Return

- Movie
    - Variables
        - Title (string)
        - Genre (string)
        - Price (double)
        - Rating (string)
        - Availability (bool)
    - Movie constructor
        - (default title, genre, price, rating, availability)
    - SetRating void function (string mRating)
        - Set Rating
    - SetGenre void function (string mGenre)
        - Set Genre
    - SetTitle void function (string mTitle)
        - Set Title
    - SetPrice void function (float mPrice)
        - Set Price
    - GetRating string function ()
        - Return Rating
    - GetGenre string function ()
        - Return Genre
    - GetTitle string function ()
        - Return Title
    - GetPrice float function ()
        - Return Price
    - GeneratateAvalibility void function (string mTitle) (for the sake of testing)
        - 20% chance to return false
        - If true, set Availability to true
        - Else, set Availability to false

- FoodShopItem
    - Variables
        - Item name (string)
        - Item values (double)
        - Hunger restoration value (int)
    - GetItemName string function (string mItem)

        - Return Item name
    - GetItemValue int function (int mValue)
        - Return value
    - GetResorationValue virtual int function
        - Pick a random int
        - Return to Hunger Restoration
    - ConsumePrint virtual string function
        - Return blank eating message
    - RemoveItem function (class deconstructor)

- Chocolate Frogs => Inheritance from FoodShopItem (subclass)
    - GetResorationValue virtual int function
        - Pick a random int between -2 and 0
        - Return Hunger Restoration
    - ConsumePrint virtual string function
        - Return “damn that’s good” message for chocolate frogs

- Choc Tops => Inheritance from FoodShopItem (subclass)
    - GetResorationValue virtual int function
        - Pick a random int between 0 and 2
        - Return Hunger Restoration
    - ConsumePrint virtual string function
        - Return “nom nom nom” message for chocolate frogs

- Popcorn => Inheritance from FoodShopItem (subclass)
    - GetResorationValue virtual int function
        - Pick a random int between -1 and 1
        - Return Hunger Restoration
    - ConsumePrint virtual string function
        - Return crunching message for Popcorn

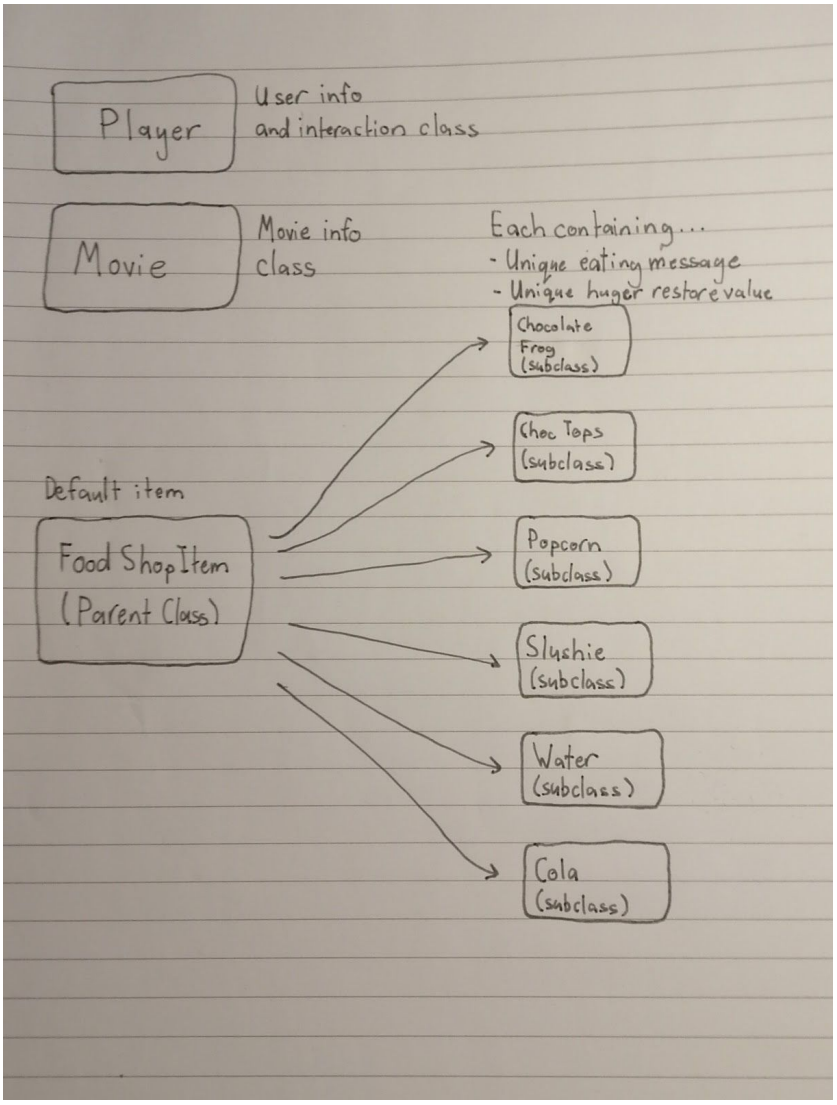
- Slushie => Inheritance from FoodShopItem (subclass)
    - GetResorationValue virtual int function
        - Pick a random int between -1 and 3
        - Return Hunger Restoration
    - ConsumePrint virtual string function
        - Return slurping message for Slushie

- Water => Inheritance from FoodShopItem (subclass)
    - GetResorationValue virtual int function
        - Pick a random int between 2 and 5
        - Return Hunger Restoration
    - ConsumePrint virtual string function
        - Return drinking message for Water

- Cola => Inheritance from FoodShopItem (subclass)
    - GetResorationValue virtual int function
        - Pick a random int between -2 and 0
        - Return Hunger Restoration

- ConsumePrint virtual string function
  - Return “ahhhh” message for Cola

Class Hierarchy Flowchart ~



Testing plan ~

Testing FoodShopItem Class and Subclasses ~

FoodShopItem class tests (to do individually on all food sub classes)

- Print out names, restoration value and price.
- Edit values and reprint
- Re-generate generation values 10 times and print result (expect to be all within given range)-
- Call consume print to make sure to prints correctly
- Call all the get functions to make sure they work correctly

Included subclasses to test on ~
Chocolate Frogs
Choc Tops
Popcorn class test
Slushie class test
Water class test
Cola class test

Testing Player class ~

- Name of the class_test.cpp
- Print out default variables: name, age, hunger, money.
- Stock user input sections with new variables and test that they changed, (name, age.)
- Select an age that is 13 and try to see Death Blow which has a 18+ rating.
- Print the hunger rating, Have default money be high enough to one of every food item, Print money. stock the food inventory with one of each item and test that it is there. Then consume each, then print the hunger rating of the player. Print money.
- Have the vector stocked with one of each of the shop items and then consume all of them. (probably a good player class test as it has the inventory vectors)

Movie class testing ~

- Call the generate availability function ten times and check that it is around 2 times that the movie is full.
- Check that the Instantly Forgotten Story is always available.
- Call get ratings() for each movie.
- Call get prices().

Main function example test 1 ~
enters name Nathan
enters age 19
at menu
selects choose movie
Buys 2 death blow tickets
goes back to menu
at menu
goes to candy bar
buys water
buys popcorn
leaves candy bar
watches death blow movie

ticket get removed from inventory
movie dialogue happens
goes to toilet
doesn’t buy more food
movie ends
leave

Main function example test 2 ~ (rating and age related money test)
enters name T-rads
enters age 11
at menu
selects choose movie
Buys 2 death blow tickets
Is rejected
Buys 2 My Dear Why Do You Cry So? tickets
Is rejected
Buys 2 Instantly Forgettable Story tickets
Could succeed or fail based on randomly generated money
If fails - Buys 1 Instantly Forgettable Story tickets
goes back to menu
at menu
goes to candy bar
buys nothing
leaves candy bar
watches movie
ticket get removed from inventory
movie dialogue happens
goes to toilet
buys a water food
movie ends
leave

Main function example test 3 ~ (infinitely regenerating money test)
enters name T11
enters age 11
at menu
selects choose movie
Buys 200 My Dear Why Do You Cry So? tickets
Is rejected
Buys 6 My Dear Why Do You Cry So? tickets
If rejected
Buys 4 My Dear Why Do You Cry So? tickets
goes back to menu

at menu
goes to candy bar
Attempts to buy one of everything
leaves candy bar
watches movie
ticket get removed from inventory
movie dialogue happens
goes to toilet
Attempts to buy one of everything
movie ends
leave