Evan Thompson
Aaron Grissom

<p style="text-align:center">Part 1: UML exercises and Semester Project Proposal – 25 points</p>

1.

<p style="text-align:center">Semester Project Proposal</p>

a): Legend of the Untitled Duck Game

b): Evan Thompson, Aaron Anthony Grissom

c): Pretty much the legend of zelda but you play as a duck. Puzzle focused top down RPG with enemies to fight. Duck has a sword and a bow and arrow to fight enemies and solve puzzles. Enemies randomly spawn in certain areas that the duck will be tasked to kill. The duck will have an inventory that holds items the duck has gained through chests and enemy loot. There will be NPCs that the duck can sell and buy items from. The duck will also have stats for its weapons and health that can be upgraded through leveling up.

d):

> Framework: Unreal Engine
> Language: C++

e): Evan Thompson:

1. Environment
2. Main Character and Enemy Implementation
3. Implementing Weapons, including different types
4. Implementing Game Manager

Aaron Grissom:

1. Inventory management including UI
2. Enemy Logic
3. Puzzle Design

2. (Next 3 Pages)

Average day at FNCD (Not a race day)

loop [While Each intern can work]

Intern.doJob()

alt [Intern can do job]

JobsDone <2

Intern.washing(Vehicle,Intern,OBSWashing)

GetVehileType()

VehicleType()

JobsDone >= 2

loop [While Each intern can work]

Mechanic.doJob()

alt [Intern can do job]

JobsDone <2

mechanic.Reparing(Vehicle)

GetVehileType()

VehicleType()

JobsDone >= 2

generate buyers

new Buyer

Wants Vehile_Type

Vehicle from Inventory to sell to buyer

Attempt to Sell Vehicle

alt [wants vehicle]

I want the vehicle

Update Inventory

Apply Bonus

[does not want vehicle]

Do not want Vehicle

most expensive Vehicle

most expensive Vehicle

Sell Vehicle request

alt [wants vehicle]

I want the vehicle

Update Inventory

Apply Bonus

[does not want vehicle]

I do not want the vehicle

3.

Tracker
day_ : String
washing_repairing_sales : String
raceAttendance_Results : String
Staff_Money : Double
FNCD_Money_ : Double
getInstance() : Tracker
update(day_, washing_repairing_sales, raceAttendance_Results, Staff_Money, FNCD_Money) : void
display() : void
static my_instance : Tracker

Singleton
Eager
Instantiation

interface Command
execute() : void

Command

CurrentTime
getTime() : String
execute() : void

Salesperson_Name
getName(Salesperson name) : String
execute() : void

New_Salesperson
newSalesperson(Salesperson salesperson) :
execute() : void

StoreInventory
getInventory() : Vehicles
execute() : void

Details
getDetails(Vehicle vehicle) : void
execute() : void

BuyVehicle
buy(Vehicle vehicle) : void
execute() : void

Buyer
chanceOfBuying: String[]
vehicleTypes: String[]
chanceStr: String
chance: Double
preference: String
GetPreference(): String
GetChance(): Double

Observer
ArrayList<String> subscription
addObservers(string) void
removeObservers(string) void
displayDailySummary(): String

Observer(s)

FNCD
salesPeople: Salespeople ArrayList
mechanics: Mechanics ArrayList
interns: Interns ArrayList
staffPerType: Integer
performanceCars: PerformanceCars /
cars: Cars ArrayList
pickups: Pickups ArrayList
carAmount: Integer
budget_: Double
soldVehicles: Vehicles ArrayList
departedStaff: Staff ArrayList
daysOfTheWeek: String[]
totalDayCount_: Integer
dayCount_: Integer
day_: String
daysToSimulate: Integer
totalSalesPerDay: Double
OBSwashing_repairing_sales : String
OBSraceAttendance_results : String
OBSchangeinMoney : String
OBSday: String
OBSStaff_Money: Double
OBSFNCD_Money : Double
Start(): void
Opening(): void
Washing(): void
Repairing(): void
Selling(): void
Ending(): void
Race(): void
HireStaff(): void
BuyVehicle(): void
PrintInventory : void

Logger
day_ : String
washing_repairing_sales : String
raceAttendance_Results : String
Staff_Money : Double
FNCD_Money_ : Double
getInstance() : Logger
update(day_, washing_repairing_sales, raceAttendance_Results, Staff_Money, FNCD_Money) : void
display() : void
static my_instance : Logger

Singleton
Lazy
Instantiation

FNCD_North

FNCD_South

Vehicles
name: String
salesPrice: Double
condition: String[]
conditionType: String
cleanliness: String[]
cleanType: String
cost: Double
enum VehicleType ()
DetermineCost(): void
GetCost(): Double
CreateName(): String
GetName(): String
GetVehicletype(): String
GetCondition(): String
SetCondition(String): void
GetCleanliness(): String
SetCleanliness(String): void
SetSalesPrice(double): void
GetSalesPrice(): double
static Vehicle Create(VehicleType)

Factory

STAFF
name: String
totalSalary: Double
JobsDone: Integer
dailySalary: Double
bonus_: Double
enum StaffType ()
CreateName(String str): void
GetName(): String
SetDailySalary(): void
GetDailySalary(): Double
GetTotalSalary(): Double
GetTotalBonusPay(): Double
Bonus(String): void
static Staff Create(StaffType)

Factory

Driver

Interns
GetJobsDone(): Integer
SetJobsDone(): void
WashingMethod(): void

Mechanics
GetJobsDone(): Integer
SetJobsDone(): void

Salespeople

Electric Cars
Range()

Cars

Pickups

Motorcycles
Engine Size()

Monster Trucks
(Stage Name())

Addon Decorator
getDescription()

Decorator

Satellite Radio
Vehicle vehicle
GetName(): String
GetSalesPrice(): Double

Extended Warranty
Vehicle vehicle
GetName(): String
GetSalesPrice(): Double

Undercoating
Vehicle vehicle
GetName(): String
GetSalesPrice(): Double

Road Rescue Coverage
Vehicle vehicle
GetName(): String
GetSalesPrice(): Double

Luxury Vehicle

Hatchback

Convertible

New Vehicle
Types

Algorithm
strategy : Strategy
Algorithm(Strategy strategy)
executeStrategy(Vehicle, vehicle, String washing

Strategy

interface Strategy
doOperation(Vehicle, vehicle, Intern intern, String washing) : void

Chemical
doOperation(Vehicle, vehicle, Intern intern, String washing) : void
chance_of_clean : bool
chance_of_sparkling : bool

Elbow Grease
doOperation(Vehicle, vehicle, Intern intern, String washing) : void
chance_of_clean : bool
chance_of_sparkling : bool

Detailed
doOperation(Vehicle, vehicle, Intern intern, String washing) : void
chance_of_clean : bool
chance_of_sparkling : bool