Project 6 Update

Thomas Mahre, Nathan Reed

GitHub Link: https://github.com/ColoradoRequiem/OOAD-Final/tree/main/src

Work Done:

> We split up the work on this project into the architecture team (Thomas) and the UI team (Nathan). As the majority of our code revolves around the traversal of islands the architecture team focused initially on creating the Decorator pattern for each event on the island. Once done, functionality of Actions were added using an Abstract Class. This then lent itself to a Strategy pattern for Requirements. This allowed actions being available only at certain times or for certain people.

> The UI team (Nathan) focused on familiarizing themselves with the Java Swing library in order to create a UI with implicit understanding of its functionality (buttons where people expect and a sleek interface).

Changes of Issues Encountered:

> The decorator pattern went more or less how we expected. A few key differences, not issues, we ran into included the description of events from other angles (North, South, …) saved as parameters in that event. This way, at run time, your current event can still give you information about another. This extended to being able to bar traversal to certain events as well via a saved 'movableRequirement' in each event as well.

Patterns:

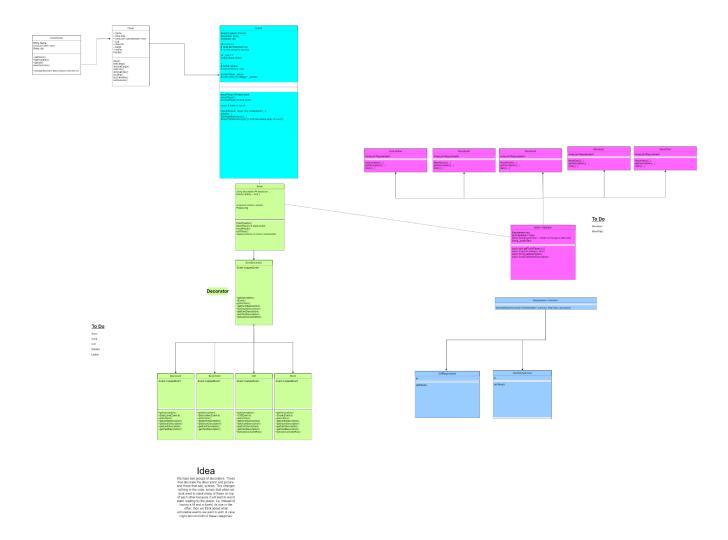> Decorator: Island Events - BasicLand, BasicWater, Coast, Cliff
> Strategy: IslandType - Star
> Abstract Class: Action - moveNorth, moveSouth, moveEast, moveWest, actionAdmin
> - Used in both Islands and likely outside
> Strategy: Requirement - CliffRequirement, NameRequirement
> - Used in both Islands and likely outside

CrewMember

String Name
ArrayList<item> item
String Job

+getName()
+getInventory()
+getJob()
+hasItem(item)
+changeItem(item item)/returns item for inv

Player

+ Name
+ Ship Ship
+ ArrayList<crewMember> Crew
+ Log
+ Favorites
+ Cargo
+ money
Position

Save()
addCargo()
removeCargo()
addCrew()
removeCrew()
buyShip()
buyCannons()
sellCannons()

Island

Season season # enum
Geography geog
dropIsNot seq

on exposed:
could be more than one
i.e. two camps or species

int _use = ?
shape shape # new

partial random
ArrayList<Event> map

private Player _player
private ArrayList<integer> _position

placePlayer() # starts event
movePlayer()
removePlayer() # ends event

case() # ability to move?

Island(Season, Geog, Veg, Inhabitabol) (...)
island() {...}
GoToIsland(action)(...)
Move() (randomized type) (// edit the double array of events)

Event

String description RI based on ...
Enemy enemy ~ NULL

ArrayList<Action> actions
Picture img

Event(season)
placePlayer() # starts event
movePlayer()
exitPlayer()
displayActions() # checks requirements

EventDecorator

+Event wrappedEvent

Decorator

+getDescription()
+Event()
+getActions()
+getNorthDescription()
+getEastDescription()
+getWestDescription()
+getSouthDescription()
+getActionNameMap()

To Do

Meow
Cove
Cliff
Dessert
Ladder

BaseLand

-Event wrappedEvent

+getDescription()
+BaseLand(Event e)
+getActions()
+getNorthDescription()
+getSouthDescription()
+getEastDescription()
+getWestDescription()

BaseWater

-Event wrappedEvent

+getDescription()
+BaseWater(Event e)
+getActions()
+getNorthDescription()
+getSouthDescription()
+getEastDescription()
+getWestDescription()

Cliff

-Event wrappedEvent

+getDescription()
+Cliff(Event e)
+getActions()
+getNorthDescription()
+getSouthDescription()
+getEastDescription()
+getWestDescription()
+getActionNameMap()

Shore

-Event wrappedEvent

+getDescription()
+Shore(Event e)
+getActions()
+getNorthDescription()
+getSouthDescription()
+getEastDescription()
+getWestDescription()
+getActionNameMap()

Idea

We have two groups of decorators. Those
that decorate the description and picture,
and those that add, actions. This changes
nothing in the code, simply that when we
dont want to stack many of these on top
of each other because it will lead to weird
static reading by the player. Ex. instead of
having a fill and a forest, its one or the
other, then we think about what
actionable events we want to add. A crew
might fall into both of these categories

ActionVote

ArrayList<Requirement>

ActionVote(c,...)
getDescription(c,...)
click(c,...)

MoveSouth

ArrayList<Requirement>

MoveSouth(c,...)
getDescription(c,...)
click(c,...)

MoveNorth

ArrayList<Requirement>

MoveNorth(c,...)
getDescription(c,...)
click(c,...)

MoveEast

ArrayList<Requirement>

MoveEast(c,...)
getDescription(c,...)
click(c,...)

MoveWest

ArrayList<Requirement>

MoveWest(c,...)
getDescription(c,...)
click(c,...)

**Action **

Requirement req
Bool disabled = False
String backgroundColor = "White" # Change to HEX ideal
String _buttonText

public bool getTruth(Player p) {}
public ArrayList<integer> click()
public String getDescription()
public String getActionDescription();

To Do

MoveWest
MoveWest()

**Requirement **

bool getValue(ArrayList<CrewMember> crewList, Ship ship, Log quests)

CliffRequirement

#

getValue()

NameRequirement

#

getValue()

Plan for next Iteration:

- Implement full game flow from menu (Main menu->Sea navigation->Island exploration)
- Crew Items and Jobs
- In depth Events that give Crew or Items
- Requirements for special events