Emma Hardison

Anjala Katuri

Katy Luttrell

Gabriela Tolosa Ramirez

CSCI 4448 – OOAD – Cat Cafe

Project 7: Semester Project – Final Submission

Final State of the System Statement:

Cat Cafe is a time management/puzzle/action game that allows the player to not only play as a barista, but also a cat taker. Our team was able to implement a functioning barista player that is able to make different kinds of drinks, as well as a menu screen, multiple levels, and customer spawning. Unfortunately, due to time constraints, scope had to be reduced and the implementation for the cats in the cafe were left unimplemented, along with their patience levels. There are graphics, as well as code skeletons and ideas to create that functionality in the future. The time constraints were heavily influenced by the change in framework (from LibGDX to SceneBuilder and JavaFX). Based on the deliverable for project 5, the scope was reduced to focus on the barista and customer control, with project 6 giving Cat Cafe a good starting point. Although the cat features were left unimplemented, the main game logic was thoroughly designed and allowed our team to develop a functioning game. The game generates customers with random orders composed of our three ingredients that build 4 different drinks – coffee, latte, lavender coffee, and lavender latte. By clicking on the ingredients, the barista builds the drink requested by the customer first in line. If the drink is wrong, the barista can click on the trashcan to throw away the current drink and try again. When the register is clicked and the drink is correct, the cost of the drink is added to the register. We implemented a patience meter for the customer, which determines the amount of tip the customer adds.

Final Class Diagram and Comparison Statement

Proj 5:
https://drive.google.com/file/d/1UprR90nSky5Jx6Zh5AHeD6hG_56ADoq_/view?usp=sharing

Proj 6:
https://drive.google.com/file/d/1oDH4mUGjsP3YtNddUwC2c717eIq4D9AR/view?usp=sharing

Proj 7:
https://drive.google.com/file/d/1JZDW87Md1MapwaHyrA729BE0SCV_P-ud/view?usp=sharing

From project 6, we added in the View class and its children: CharacterView, DrinkView, and LevelScreenView. These View classes are included in the MVC pattern. An instance of CharacterView was added to GamePlayController, as well as a PlayableCharacter object. A trash button was added to GamePlayController as well as a handling method for it. Many new methods were added to GamePlayController to trigger changes in images according to game logic and mouse events. DemoLevel, which was a temporary set up for project 6, was replaced with the Level class which has 3 levels that inherit from it. We also made PlayableCharacter a singleton since project 6. Since project 5's UML, we did not implement or keep the classes: Driver, InventoryItem, CatCustomerInteraction, CoffeeShop, Cat, CatItem, CatRequest, Clickable and the interface Walkable. All the cat related ones we did not have time for and the other ones ended up being unnecessary. We also took away the second Command pattern for the menu options outside of gameplay since it was not needed.

Third-Party code vs. Original code Statement:

The bulk of our code is original, and everyone on the team created the graphics from scratch. There were multiple sources that allowed Cat Cafe's creation. Tutorials allowed us to learn more about Aseprite – graphics and animation – JavaFX and SceneBuilder. We worked through multiple tutorials and utilized many resources to build Cat Cafe. Sources of third-party code are attached in comments throughout the code, as well as included in the list below:

- https://edencoding.com/check-whats-been-clicked/#:~:text=When%20a%20button%20is%20clicked,when%20the%20button%20was%20created
- https://www.youtube.com/watch?v=T3NlWMzPyXM&ab_channel=ProgrammingKnowledge
- https://www.youtube.com/watch?v=WDaXpDtYk3E&list=PLrzWQu7Ajpi26jZvP8JhEJgFPFEj_fojO&ab_channel=Randomcode
- https://youtu.be/AYeFmhQALqM
- https://mkyong.com/java/how-to-write-an-image-to-file-imageio/
- https://stackoverflow.com/questions/3489543/how-to-call-a-method-with-a-separate-thread-in-java
- https://www.tabnine.com/code/java/methods/javafx.scene.image.ImageView/setVisible
- https://stackoverflow.com/questions/3489543/how-to-call-a-method-with-a-separate-thread-in-java

- https://www.demo2s.com/java/javafx-pathtransition-setonfinished-eventhandler-actionevent-value.html
- https://stackoverflow.com/questions/37752207/javafx-wait-for-animation-method-to-finish-before-going-to-next-method
- https://stackoverflow.com/questions/17850191/why-am-i-getting-java-lang-illegalstateexception-not-on-fx-application-thread
- https://www.tutorialspoint.com/find-minimum-element-of-hashset-in-java#:~:text=To%20get%20the%20minimum%20element,min()%20method.
- https://www.geeksforgeeks.org/double-compare-method-in-java-with-examples/#:~:text=The%20compare()%20method%20of,returned%20by%20the%20function%20call.
- https://www.tabnine.com/code/java/methods/javafx.scene.image.ImageView/setVisible
- https://attacomsian.com/blog/java-get-unix-timestamp
- https://stackoverflow.com/questions/767759/occurrences-of-substring-in-a-string
- https://stackoverflow.com/questions/3489543/how-to-call-a-method-with-a-separate-thread-in-java
- https://stackoverflow.com/questions/3489543/how-to-call-a-method-with-a-separate-thread-in-jav
- https://stackoverflow.com/questions/1972392/pick-a-random-value-from-an-enum
- https://www.baeldung.com/java-initialize-hashmap

Statement on the OOAD process for your overall Semester Project:

1. Our analysis phase was very positive overall- while making our UML we tried to keep a bird's eye view of the system and how it will all fit together very carefully before diving into coding. The only challenge of this process was that since none of us had used the framework we were planning on using, there were some parts of the system design that we either didn't know how would interact with the framework, or ended up being unnecessary once we started working with the framework.
2. We went into design knowing we needed to use patterns, so we used Design Pattern Driven Design before we started building the UML. We talked about what patterns seemed relevant or helpful for our project. Since it is a GUI with internal game logic, MVC quickly came up and drove how we thought about building the game. Since we

wanted to build coffee drinks and the textbook had a very similar example in the decorator chapter, we knew we wanted to incorporate that. This worked really well for us and gave us a nice, easy way to communicate ideas and a good foundation to build on.

3. Commonality and Variability Analysis allowed us to organize our UML diagram in a much more readable way, and made it more useful to us when we started implementation. We started with a long list of requirements we wanted in the game, so finding commonalities between items and processes let us reduce the amount of classes we had.

Code Submission – GitHub Repository:

https://github.com/ramirez-gabriela27/CatCafe

All code and artwork is originally created by the four team members of Cat Cafe (unless otherwise denoted). The GitHub repository includes all of the code under the `game` directory, artwork created to inspire the game assets under `Graphics` as well as all project deliverables. Within every directory, there is a README with information on the directory, and all code is commented and cited properly.

Demonstration – recorded demo video:

https://drive.google.com/drive/folders/1b-OL77BXjBaoVHK2XuKhWlKkJsbJF7fS?usp=sharing

Project demo presented by all team members of Cat Cafe and edited by Anjala.