Jiya Patel
Period 2

Jiya’s Pizzeria

Description

My project is a replication of the game “Papa's Pizzeria” except with some modifications and additional features. The objective will be to get the most points you can by playing the game. You can get points by following the recipe on the receipt accurately and completing your pizza before the time runs out. Following the receipt instructions will include putting your pizza in the oven at the right temperature, for the right amount of time, putting the right toppings on, the right amount of toppings on, and more.

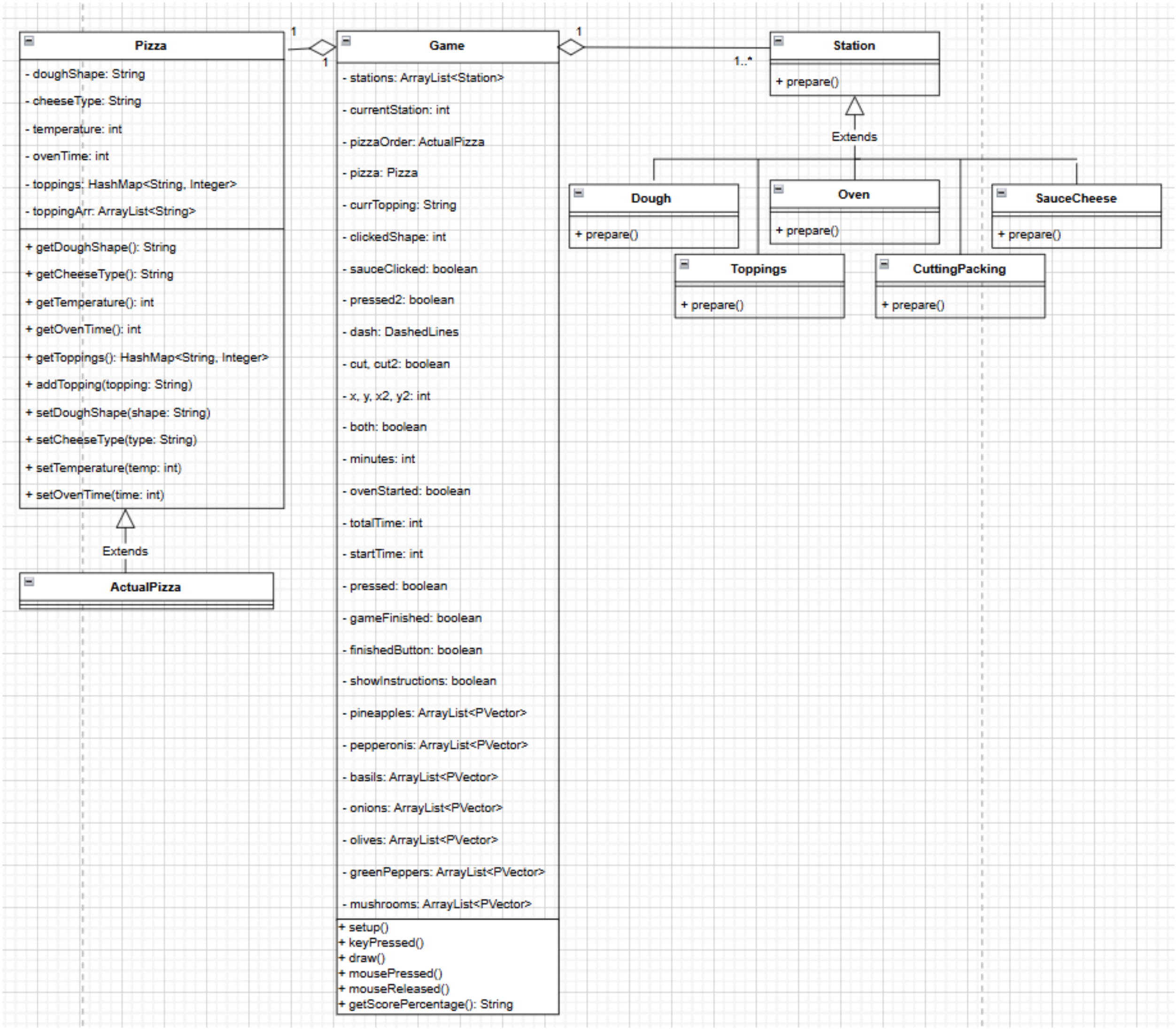
The libraries I will need for now will be Dashed Lines by Jose Luis Garcia del Castillo, which I will be using to allow the user to cut the pizza by clicking the mouse at the position I want to end the line. Other than that, I will be using buttons, shapes, and a display timer. I will also use the arrow keys to switch between the different stations in the game such as the dough-making station, the oven, the cheese/sauce station, toppings station, and cutting/packing station.

How Does it Work?

Objective: Finish making the pizza accurately in the given time + optimize the amount of points

1. User can click on the question mark button in the bottom right corner to see instructions and other notes on how to play the game and how it works.
2. User will press the “play” button and receive a receipt which lists their instructions/ingredients. Once they press “play,” their timer will start and show up on the top of the screen.
3. By clicking the right and left arrow keys, the user can switch between different parts of the pizza-making process, such as the dough-making station, the oven, the cheese station, toppings station, and cutting/packing station.
4. At the dough station, the user will draw a circle with their mouse and the pizza will form into that circular shape. They can change their original shape by clicking the other and it will change.
5. At the SauceCheese station, the user will first put sauce on the pizza by clicking on the tomato sauce bottle. The user will then click the type of cheese that the receipt says to put. Cheddar will be yellow cheese and the other types will be a lighter, whiter color. The user cannot put cheese on the pizza before the sauce is put on.
6. At the toppings station, the user will put toppings based on what the receipt says (pepperoni, mushrooms, basil leaves, pineapples, olives, onions, green peppers, etc). The user will have to drag each topping from the right bowl of toppings and put them evenly throughout the pizza. Points will be deducted if the wrong toppings are put and also if the wrong amount is put (the amount is specified on the receipt). The user will not be able to put toppings outside of the pizza, they must place them on the pizza once dragged out of the bowl.
7. At the oven station, there will be a temperature button where the user has to adjust the temperature to be the right temperature and the time as well. The user may click the plus and minus signs to do so. Then the user will click the button to start the oven. If the pizza is cooked the right amount or less than that, the dough will turn slightly darker. If the pizza is cooked for a longer time or higher temperature, the pizza dough will be burnt and much darker. If the user cooks it for a longer/lesser time or at a higher/lower temperature, points will be deducted from their final score.
8. At the final station I will make the user cut the pizza using a dotted line animation feature that will draw the dotted line where the line is starting and the user will click where they want to cut and where they want the dotted line to end. The user can click a button that says FINISHED. They will then be given a score percent.

Updated UML Diagram



Current Functionalities

- Play button to start game
- Question mark button to click on home page of game for instructions and other notes
- 2 minute timer counting down once game started
- Dough station: selecting a pizza shape (either circular or sicilian)
- Using arrow keys to switch between stations
- Randomly generating an order/instructions for each time a game is started
- SauceCheese station: add sauce when clicked, and add respective cheese type when clicked
- Toppings: dragging and placing different toppings onto the pizza
- Oven Station: adjusting the oven temperature and time
- CuttingPacking station: use mouse coordinates and dashed lines to cut the pizza and ending the game to get back a final score

Issues

1. Before, when the user clicked the circle first (which would draw the circle on the screen), but then clicked the square, both the circle and square would be there. This resulted in the circle and square layered on top of each other which would happen if the user changed their mind and clicked the opposite shape after clicking one of them. To fix this issue, I added the variable “clickedShape” and initialized it to 0. 1 would represent the circle and 2 for the square. I added a condition to check if the shape on the screen was already a circle or square, because then I wouldn’t change it to be the same, but if it was something else, then I would change the clickedShape variable and redraw it in the draw function, thereby replacing the old one and making for a cleaner, more functional look.
2. My buttons wouldn’t work when clicked but I realized it’s because draw() will overwrite it every time it runs since it runs every single second and so whatever I’m doing that I want to put on my screen as a constant would have to be done outside of my mousePressed if statement.
3. Another major issue was the countdown. At first I used seconds(), minutes(), and hours() but realized that this was for the actual time and not for your own custom countdown. So I did more research and used millis() and also discovered nf() which adds leading zeros to make sure the time on the countdown made sense and looked like a traditional clock down. But then, the numbers would layer on top of each other, so I added a rectangle that would go on top of the old one each time in the draw() function in the Game class.
4. Also, when switching the current station the user is on through use of the arrow keys, I had originally copy pasted a snippet of code from my kernels project, which I then realized was different from mine since I didn’t want the user to loop backwards or forwards around but instead only go in a linear direction either back or front. Also, I added a condition to deal with the out of bounds error I was getting as a result of modifying my code.
5. Another issue was sizing the background for the Dough station (the wood counter background) to fit the screen since I kept getting the error that the background image Im importing must be the same size as screen. I found an image that would proportionally enlarge to my screen size, adjusted the screen size a little to match, and put it in a picture-enlarging generator to fix the issue.
6. The timer was showing on every station when switching between stations using only the arrow keys, but not showing up when I clicked on a pizza shape in the first station and continued playing. This was because my timer code in the draw() function was that the top, and the code below it would draw new backgrounds and other stuff on top of the timer. So I moved the timer code to be at the end of the draw() function.
7. A major issue I encountered was figuring out how to use either an Array or ArrayList for my toppings because when the user clicked and dragged a topping onto the pizza, it would be hard to search an Array or ArrayList made of Topping objects and search each Topping object to see if anyone of them matched the String of the topping that was just put on the pizza and to increment this in the Topping class. Instead, I learned what a HashMap was and how to use this, which would allow me to have key value pairs so that the key can be the topping name and the value would be the amount of the topping. This replaced my need for a Topping class, so I deleted that class entirely. Additionally, for HashMap I don’t need to search it anymore because if I add a key value pair with the same key (same topping name) it’ll overwrite the old one!

8. Another issue is because I only found one type of image on the internet for the cheese and it was yellow, which is for cheddar, I still need a white cheese just like that for the rest of the four types. So I discovered the tint function. However, the tint function won't allow me to make that yellow color into a brighter/lighter/whiter color, instead only allowing darker tints/shades so I think I am going to edit the cheese image on a separate editing app and then use that.
9. Another issue was with the dashed lines. First, importing this library was very hard. It took me very long to do. Another issue was getting it to start and stop when the mouse was clicked/pressed and not draw a bunch at the same time wherever the mouse was moving. To fix this I added a bunch of x y coordinate variables for each line and also boolean variables for each line to keep track of different stages in the line.
10. A minor challenge I faced was that the sauce and cheese kept showing up at the same time. Changing up the if statement conditions a bit, I was able to maneuver around this. It was hard to dissect the problem initially because I was confused where the issue was coming from, but by adding print statements in different places in both my Game class methods and SauceCheese class, I eventually found the issue.
11. Another issue was when the oven time and temperature only showed up sometimes. If I switch through the stations fast, it may show up, but other times it doesn't. I realized eventually that it was because I put it under the "else if" condition in that method rather than an "if."
12. An issue I had is the toppings and dragging and dropping them onto the pizza. I had difficulty trying to keep track of how I would move an image I imported on the screen and do this for all the different toppings I have on the screen and multiple times. I used mousePressed(), mouseReleased() and ArrayList of PVector as coordinates for each topping to keep track and store these values.
13. Another issue was that if I placed the toppings outside of the pizza, these values would also be stored and I only wanted the ones on the pizza to be shown. I used an if statement which encompassed my previous code about adding a coordinate to the arraylist for the specific topping. This if statement checked if the topping was within the coordinates of the pizza or the box the pizza would be placed in at the end.
14. A major issue was calculating the score. It took me very long for some reason because there was one minor error in it that resulted in me always getting a score below 50%. It took a lot of print statements and editing my code to fix this. I created a system of calculating the score where it would be a point for everything being correct except the toppings would be based on the same amount of topping types, the same amount of total toppings, the correct toppings being present, and the correct topping amounts correlated to topping if present.
15. Another issue was that the time kept going past zero and into negative numbers. Also, the time kept running even after the game finished. I fixed both these issues by manipulating the original time code in draw to only proceed under certain conditions.