Design Proposal [20 pts]
Your Design Proposal should be submitted as a directory containing several files:

- **Project Proposal** [15 pts]
  Write up a proposal file (in the file proposal.txt, or .docx, or .pdf) which should include the following components:
  - **Project Description** [2.5 pts]: The name of the term project and a short description of what it will be.
  - **Competitive Analysis** [2.5 pts]: A 1-2 paragraph analysis of similar projects you've seen online, and how your project will be similar or different to those.
  - **Structural Plan** [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or objects.
  - **Algorithmic Plan** [2.5 pts]: A detailed algorithmic plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most complex, and include details of the algorithm(s) you are using in those cases.
  - **Timeline Plan** [2.5 pts]: A timeline for when you intend to complete the major features of the project.
  - **Version Control Plan** [1.5 pts]: A short description **and image** demonstrating how you are using version control to back up your code. Notes:
    - **You must back up your code somehow!!!**
    - **Your backups must not be on your computer** (ideally, store them in the cloud)
  - **Module List** [1 pts]: A list of all external modules/hardware/technologies you are planning to use in your project. Note that any such modules must be approved by a tech demo. If you are not planning to use any additional modules, that's okay, just say so!
- **Storyboard** [5 pts]
  Generate a storyboard that demonstrates how a user would interact with your finished project. Your storyboard should have at least six panels, and at least three of those should demonstrate features within the project. You may scan or take a picture of your storyboard and include it in the directory as the file storyboard.png (other acceptable filetypes include .gif, .jpg, and .pdf).

Project Description
My term project is titled La Barista. I aim to create a cafe game that represents an upgraded, ideal version of CMU's La Prima. In this game, the player controls both the barista who takes orders and makes drinks, and the waiter who serves the drinks to the customer.

Competitive Analysis
La Barista is similar to famous cooking games like Overcooked, Diner Dash, Cafe Tycoon where players build a cafe or restaurant and go to different cooking stations to make food and/or drinks. La Barista will be similar in that it also tests the players ability to quickly and accurately produce the required menu items to meet a score goal, before moving on to the next level.

However, La Barista incorporates more interesting elements such as using OpenCV to draw latte art, which makes the game more 'hands-on' and engaging. Another interesting aspect would be the pathfinding element that shows the most optimal path to serve drinks to the customers.

Structural Plan [2.5 pts]: A structural plan for how the finalized project will be organized in different functions, files and/or objects.
The project will be organized using

- Main page for appStarted and runApp
- Modes for different screens (cafe layout, drink making, scoring, log in, instructions, progress)
- A classes file for characters, furniture, drinks, customers
- Folders for OpenCV related material (finger tracking module, latte drawing function, images of latte art customers will ask for), images

Algorithmic Plan [2.5 pts]: A detailed algorithmic plan for how you will approach the trickiest part of the project. Be sure to clearly highlight which part(s) of your project are algorithmically most complex, and include details of the algorithm(s) you are using in those cases.

The latte art aspect would be done using fingertracking with the OpenCV and mediapipe modules. OpenCV supports computer vision and allows the program to take the player's camera as input. Following that, the mediapipe module supports finger detection (in particular, the index finger) to draw the latte art as the user moves the index finger around. The drawing is then mapped on a separate canvas integrated with the 112 graphics.

The pathfinding aspect is done using the A*STAR search algorithm. Coordinates on the board are taken in as input, while weights are given to people and other obstacles so that they can be avoided. The most optimal way to reach a customer is then shown by a lighted path on the ground for the waiter to follow.

The scoring aspect is done by using image processing on OpenCV to detect the similarity between the drawing and the preset desired art of the customer. The proportion of ingredients is scored by measuring the color of the drink by calculating the difference in RGB numbers.

Random customers with random orders would also be generated, using various combinations of latte bases, flavours and art designs.

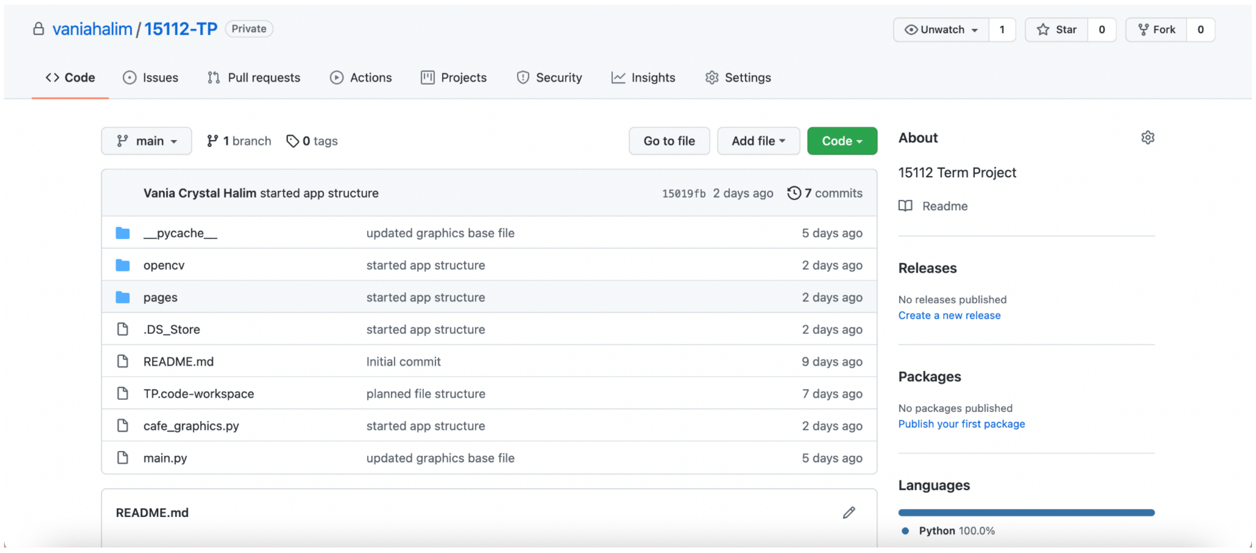
Timeline Plan

Feature	**Target date**
~~Code structure~~ • ~~Organize folders and files~~ • ~~Incorporate 112 graphics~~ • ~~Create classes~~	~~14/11~~
~~Cafe layout screen + UI (3D images)~~ • ~~Grid~~ • ~~Movable barista and waiter~~	~~15/11~~
~~Drink making + latte art screen~~ • ~~Choosing espresso shots, flavours and proportions~~ • ~~Animation for filling up cup, changing color of coffee, overflow~~ • ~~Incorporate OpenCV with 112 graphics~~	~~17/11~~
~~Game AI~~ • ~~Randomization of order~~ • ~~Randomization of location customer will stand while waiting~~	~~18/11 (TP1)~~
~~Pathfinding~~ • ~~Finding optimal pathway to serve drinks to customer~~	~~19/11~~
~~Scoring screen~~ • ~~Formulas to calculate the score of player's created drink~~ • ~~OpenCV for image processing of latte art drawing (compare vector points?)~~ ~~End of day screen~~ • ~~Progress bar (1,2,3 cups)~~ • Drinks made summary	~~21/11~~
Misc screens • Menu w pricing page (drinks + pastry) + menu icon in grid	23/11 (TP2)
Increasing level difficulty • Adding more features (more drink options, pastries, lowering image analysis threshold, increase obstacles)	23/11 (TP2)
Game AI	24/11

• Taking orders for other customers • Customers moving in and out of cafe	
UI • Furniture (doors, order station, drink-making counter, pastry counter, tables) • Customer sprites • ~~Day progress bar (Tkinter)~~ • ~~Day timer (Tkinter)~~	Post MVP (23/11)
Log in feature • Saves player info so they can continue to next level when they log in • Progress chart of past days scores	Post MVP (23/11)
Customizable features • Unlock new flavours/pastries with cups	Post MVP (23/11)

Version Control Plan

Git is used for version control. After a day of working on the term project from VSCode, changes would be committed and pushed to Github


vaniahalim / 15112-TP Private

Vania Crystal Halim started app structure 15019fb 2 days ago 7 commits

__pycache__	updated graphics base file	5 days ago
opencv	started app structure	2 days ago
pages	started app structure	2 days ago
.DS_Store	started app structure	2 days ago
README.md	Initial commit	9 days ago
TP.code-workspace	planned file structure	7 days ago
cafe_graphics.py	started app structure	2 days ago
main.py	updated graphics base file	5 days ago

README.md

About

15112 Term Project

Readme

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Python 100.0%

Module List

- OpenCV
- Numpy
- Mediapipe
- PIL
- Tkinter

- Pickle

TP2 Update

- Changes to MVP definition
    - Increase difficulty of game by adding more features (more drink options, pastry options, lowering image analysis threshold, increase obstacles) as game goes on
- Pathfinding shown as dots on grid
    - Characters now move one grid at a time
- Scoring aspect done using OpenCV module