Anushka Iyer
ai2

Project Proposal

- **Project Description:** My term project will be an implementation of Fruit Ninja, and will be called "Fruit Ninja." In this game, different fruits are thrown up from the bottom of the screen, and the user must "cut" the fruit to get points. If the user misses a fruit, they lose a life. There are only three lives for each round played, and once all three lives have been lost, the game is over. Another way for the game to end is if the user accidentally cuts a bomb. The bombs are thrown up similarly to the fruit, however, these are to be avoided. This is the original version of Fruit Ninja, however, there is a different mode called "Zen Mode" in which there are no lives and no bombs, so users are simply playing to get a high score.

- **Similar Projects:** There are similar projects inspired by Fruit Ninja online, however in those versions, you "cut" the fruit by clicking on the fruit, or using the mouse to click and drag. In my implementation, I want to use OpenCV to track the brightest pixel on the screen, so that users can use their phone flashlight to cut the fruit. I saw people using the flashlight method in the term project gallery as well. Another difference I saw was that when the fruit was being cut, other games just changed the image to a cut piece of fruit, however, I actually want to make it so that the image is cut where the user cuts the fruit and watch the two halves fall.

- **Structural Plan:**
  Game Class: Manages the overall game flow and controls.
  Methods:
    - Start game
    - End game
    - Update game state
    - Handle user input

  Fruit Class: Represents individual fruits on the screen.
  Attributes:
    - Type of fruit (apple, banana, etc.)
    - Position (x, y)
    - Velocity

  Methods:
    - Update position
    - Check for collisions
    - Slice (if intersected by the player)

Player Class: Represents the player's avatar.
Attributes:
- Score
- Lives

Methods:
- Update score
- Decrease lives on a missed fruit
- Perform slicing motion

GameView Class: Manages the visual representation of the game.
Attributes:
- Canvas for drawing

Methods:
- Draw fruits
- Draw player
- Draw background
- Update display

GameController Class: Handles user input and communicates with the Game class.
Methods:
- Handle touch/swipe input
- Pause/Resume game

SoundManager Class: Manages game sounds.
Methods:
- Play slice sound
- Play miss sound
- Play the game over sound

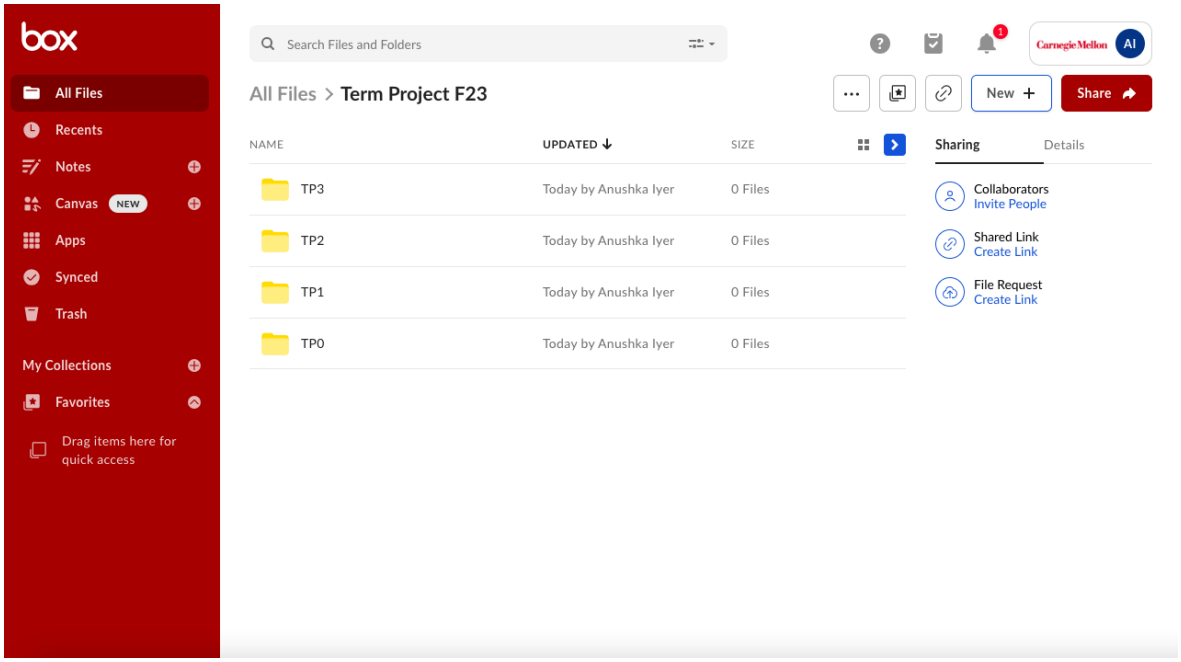
onAppStart: Instantiates and orchestrates the main game components.
- Initialize game
- Game loop (update, reload, handle user input)

- **Algorithmic Plan:** The most difficult part of building Fruit Ninja would be figuring out how to slice the fruit images where the user moves their mouse, and breaking the image into two pieces so that they fall separately. To try and attack this problem I would first Use touch or mouse input to detect when the player starts and ends a swipe gesture. Then I would record the initial and final positions of the swipe to determine its direction. Determine the area covered by the swipe and check for intersections between the swipe

area and active fruit objects on the screen. For each intersected fruit, create two new fruit objects representing the sliced halves. I would then adjust the velocities and trajectories of the new fruit objects based on the original fruit's properties. Within this, coding up the new velocities and trajectories will probably be the trickiest part. I will also be drawing my fruit by hand and I will draw the whole fruit as well as the sliced version of the fruit.

- **Timeline Plan:** I have added a schedule of all the main due dates, and I am writing out what specifically I need to get done between each of those dates.
    - *November 20th - TP0 due*
        - Prepare drawn images of whole and sliced fruit
        - Code fruit being thrown up from the bottom of the screen
        - Code random bombs are also being thrown up
        - Develop slicing mechanism
    - *Novemver 27th- TP1 due*
        - Implement scoring lives and game-over
        - Add a fruit ninja background and sound effects
        - Create varying levels of difficulty
        - Adding in OpenCV to use to slice fruits
    - *December 1st - TP2 due*
        - Polish up the UI of the app, making it look clean
        - Create Zen Mode, without bombs and no lives
    - *December 6th - TP3 due*

- **Version Control Plan:** I will be using the Box cloud storage to back up my code files. I have created a Term Project Folder, and within that, I will have a folder for each checkpoint where I will save all the work that I am doing for each respective checkpoint.



- **Module List:**
    - OpenCV

TP1 Update

- I am adding the ice cube feature. Ice cubes are thrown up, and if the user slices the ice cube, the screen is "frozen" for a short period and the fruits are thrown up slower so they are easier to slice.
- I am considering not using images for my fruit and instead using CMU graphics to create them, that way I can maintain my line intersect the circle function

TP2 Update

- I added a second bomb in zen mode, when this is hit the screen is cleared and it wastes 2 seconds.
- I also added lots of sound effects

TP3 Update

- Watermelon and dragonfruit hairs rotate
- The timer has been updated and polished
- Splash screens have been polished

- Instructions Page added when you click question mark on start screen