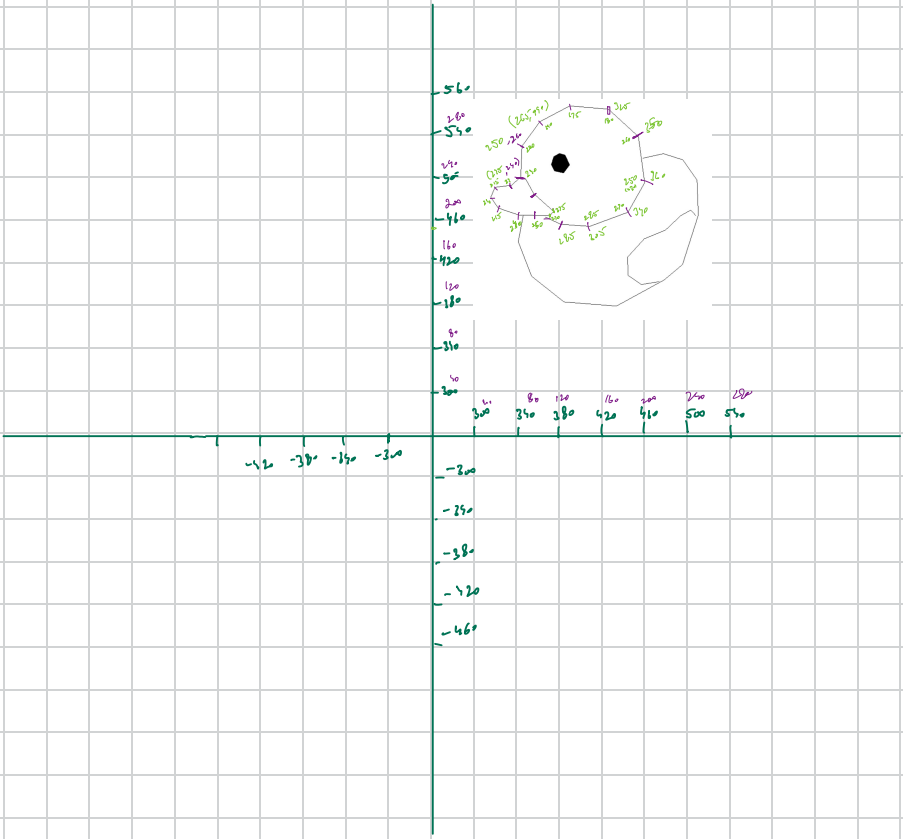
Rough

$$y - y_0 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_0)$$

$$y = y_0 + \frac{y_2 - y_1}{x_2 - x_1}(x - x_0)$$



Final Project

Interactive Game.

$A = x, y.$

P

$\sigma = \frac{P}{A}$ $\sigma = E \epsilon$

Detailed Design Architecture

Game Mechanics Level ups, Power ups, Tile, Visuals Interface.

Main File :-

Ball has fallen → check lives
- ↳ Power ups
- ↳ Tiles

Paddle width
Ball speed.

Leaderboard history
Playboard.
Database Management
State Transition

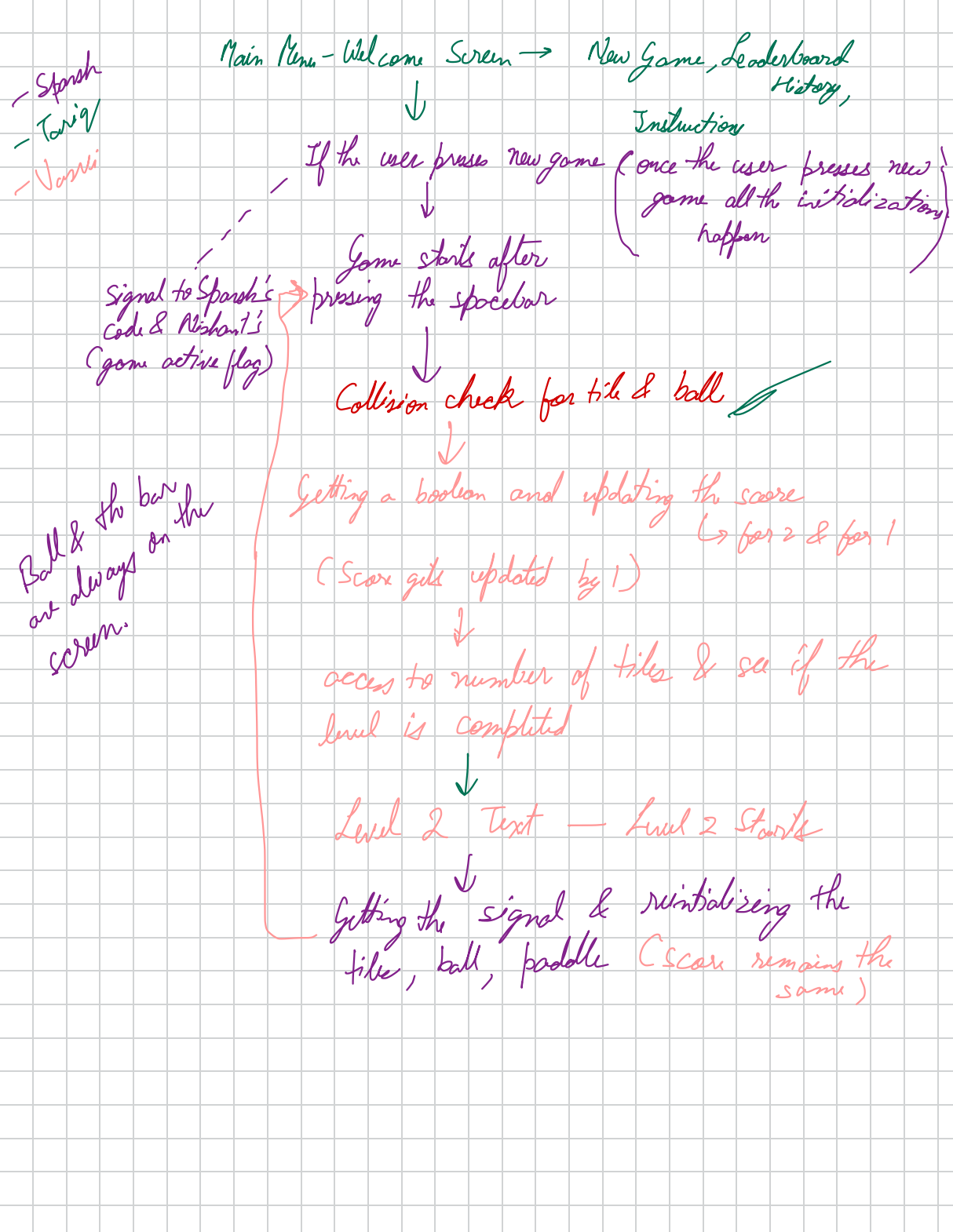
Paddle width for 10 sec.
Power ups — Ball speed.
↳ Paddle

Tiles

Level up — Speed
↳ Paddle

Sparsh
Obstacles
Tiles Design
Sound — Toriq
↳ Esc Screen

Vasu:
Power ups — Lives, Paddle width, Speed of ball.
2 Lives.
Level ups


- Sparsh
- Tariq
- Vansh
Main Menu - Welcome Screen → New Game, Leaderboard, History, Instructions
If the user presses new game (once the user presses new game all the initializations happen
Game starts after pressing the spacebar
Signal to Sparsh's code & Nishant's (game active flag)
Collision check for tile & ball
Getting a boolean and updating the scoore
↳ for 2 & for 1
(Score gets updated by 1)
access to number of tiles & see if the level is completed
Level 2 Text — Level 2 Starts
Getting the signal & reinitializing the tile, ball, padelle (Score remains the same)
Ball & the bar are always on the screen.

User presses Esc

↓

State of the game to be preserved

↓

Pos of ball, width, paddle (store the values)

↓

options — Resume, exit, main menu,

Nishant's module — Life check

↓

Reduce Life

Tile Colors (give boolean)

↓

To power up specifically

↓

loop for boolean → False in end

give signal ←

↓

Speed up / width

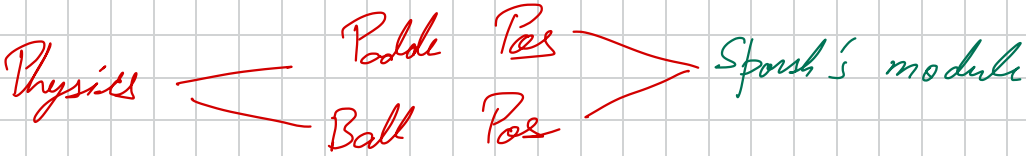
If flag (zero lives)

↓

Tanq, Sparsh, Nishant's module

↓

Better luck next time
Leaderboard, Play Again,
Exit Game.



Collision check of tiles

Rendering of balls, paddles — Nishant

Rendering of tiles — Sparsh

Function call to Nishant.

Main — for loop

Collision ck — Nishant
Ball pos / Paddle pos → Nishant

Pos of Paddle / Ball → render → Sparsh