APCSA Final Project

Period: 1 Group: Yu Lu

Group name: pizzacool **Project Title**: Radish

Description:

A Rhythm Game written in processing based on designs of Super Hexagon with static input designs from ddr and cytus

Player controls:

Notes instead come from all around the screen into the middle, with your mouse controlling a small radius to sort of parry the notes OR a set of 8 directional keys to hit notes in 8 directions.

Lens Notes

- Emanate from random points on the screen perimeter.
- Travel in straight lines toward the exact center.
- Hit window is tight—timing and positioning both matter.

Central Safe Zone

- A small target circle (radius r, can be set) at screen center.
- If a note enters this circle, you're penalized (health loss or point deduction).

Full-Circle note

- Occasionally, a "wave" circle expands inward.
- When it hits the designated gray outline, you must have no notes in the safe zone—or take heavy damage.

Terms:

Lens note - regular note, with timing approaches the middle of the screen
Full circle note - Secondary note mechanic, a full circle that approaches the middle of the
screen, timing should be separate from main notes?
Outline gray circle - timing outline for full circle notes
Breaking point - point at which the player will be punished for not hitting the note

Essential UMLS

GameManager

- AudioManager audio
- BeatmapLoader loader
- NoteSpawner spawner
- InputManager input
- ScoreManager score
- + setup()
- + draw()
- + startSong(String chartFile)

ScoreManager

- int score
- int combo
- + registerHit(String judgement)
- + display()

InputManager

- + handleInput()
- + checkHits(float currentTime)

AudioManager

- SoundFile song
- + loadSong(String path)
- + play()
- + getCurrentTime(): float

BeatmapLoader

- int bpm
- List <NoteData> notes
- + loadChart(String path)
- + getNextNote(): NoteData

Note

- float spawnTime
- float hitTime
- int lane
- + update(float currentTime)
- + display()
- + isHittable(float currentTime, float

window): boolean

MVP (minimally viable product):

- Get music playing (done)
- Make a basic background and map first
- Code the basic spawning function/class for the notes, should ideally contain 8 directions for them to spawn. The notes should be like a part of a circle (like a lens shape), they start off big around the edge of the screen then get smaller as they get closer to the middle (this will mimic a sort of falling or 3d effect)
- Code the basic full circle note spawning, same idea with the normal notes but they should instead stop after the outline gray circle radius
- Try to add timing to the notes somehow, how can i do this? Timing should ideally determine when the notes will approach breaking point