AP Computer Science Final Project - README

HYPERGUNNER

Authors: Daniel Im, Edward Lu, Dhruv Sharma
Revision: 5/24

Introduction:
The project will be a rhythm game in the style of galaga with a player controlled ship at the bottom of the screen that can move left and right between various "lanes" to the beat of a song to avoid oncoming debris and move through a specific path (basically, there will be lanes that you will need to switch between for you to not crash based on the rhythm). Players can also press a button when in "energy fields" to shoot and destroy rocks in front of them at specific points in a song. The ship will have a set amount of health that decreases each time the player hits something(walls, debris, ectr). The map will scroll down past the player, and they will remain vertically stationary at the bottom to leave the most amount of space above the screen so they have time to react to oncoming obstacles. Maps can be stored as 2D arrays read from text files, and will likely have 3 or 5 lanes. Maps should ideally scroll quite quickly, though a balance between speed and playability will need to be considered, as moving too quickly for the sake of energy and rhythm could make the game impossibly hard. There isn't much of a story, but the goal of the program is to complete all the levels, possibly with a scoring system, either numerically or with an D/C/B/A/S scoring system. Fans of games like Thumper will probably enjoy this, as their rhythm-based gameplay is somewhat similar.

Instructions:

Use the A and D keys to move lanes left and right, and use W to shoot (there should be a delay so even if you do hold it down, there would be some instances where it does not hit anything), primarily keyboard controlled
A level select menu will be needed, and maybe another one at a higher level to decide between playing or level editing if that ends up being added

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):
Must-have Features:
[These are features that we agree you will *definitely* have by the project due date. A good final project would have all of these completed. At least 5 are required. Each feature should be fully described (at least a few full sentences for each)]

- The ship should be able to shoot and destroy specific types of rocks but not others

- Music should be able to be played and the screen should be able to scroll at a rate that matches the tempo of the song
- The user must be able to press keys to move, as well as use their mouse to select menu options
- Backgrounds and the ship should be represented by sprites
- Some kind of instruction will be necessary, the bare minimum would be a list of controls in the menu

Want-to-have Features:

- The ability to create maps using some kind of editor or the ability to drop in a txt file containing X's and O's that can be then shown in game. Hopefully players will also be able to add music and modify the scroll speed to match it
- Special Effect animations for perfect moves like a near perfect shoot(the rectangle representing the ship almost exactly aligns with a rectangle representing the area the player can shoot, for example)
- A scoring system based on how many times someone gets hit and/or how many perfect shots they hit
- The ability to add a custom sprite for users to have their own custom ships
- Additional levels with even more lanes, possibly around 7 or even 9
- Temporary powerups that give players powers like invincibility or the ability to shoot through previously indestructible walls just like they would for debris

Stretch Features:

- Smoother scrolling and player movement, possibly using movement curves and effects for more satisfying lane switches. Could even add effects like screen shake for player deaths
- Local competitive multiplayer where 2 players can compete for high scores
- Unique music made just for the game with their own maps

Class List:

Screen - Superclass for the different screens
StartScreen - The beginning title screen containing the title and a start button
MenuScreen - The screen containing level select
Game - The screen where maps are drawn, music is put on, and the main game is played
RandomGame - The screen where the endless infinite game is played
GameOverScreen - The screen the player goes to once they die
ScreenSwitcher - Class for switching from one screen to another
AmmoCrate - Represents Ammo crates that can replenish ammo in endless mode

Bullet - represents the bullets that can be shot to destroy oncoming blocks in endless mode
Coins - represents the coins that can be picked up in endless mode
HealthPackage - represents the endless mode pickup that replenishes health
Map - Uses a txt file to generate a grid map and draw it
RandomMap - randomly generates an infinite map for the endless mode with additional pickups
StarShip - represents the player ship that can move and shoot
Sprite - the superclass for other classes represented with x/y coordinates and images
WallChunk - represents indestructible walls

Credits:

- Edward - Start Screen, Sprite/Ship Class, Destructible Terrain, Infinite Gamemode
- Dhruv - Normal Map Class, Game/Menu Screen, Art Assets, Video editor, javadocs, shapes library, readme updates
- Daniel - Game Over Screen