

Ray Bike Supreme Game Plan

Randy Gulak

BCIT FWDP 2021

Table of Contents

1. [Ray Bike Supreme Overview](#)
2. [Ray Bike Supreme Design Statement](#)
3. [Ray Bike Supreme Play-By-Play](#)
 - a. [Start up](#)
 - b. [Bike Selection](#)
 - c. [Gameplay!](#)
4. [Ray Bike Supreme Ingredients List](#)
 - a. [Visible Elements \(DOM Elements\)](#)
 - b. [Invisible Elements](#)
5. [Ray Bike Supreme Concept Art/Resources](#)
6. [Questions / Comments](#)
 - a. [Questions](#)
 - b. [Comments](#)

Ray Bike Supreme Overview

Ray Bike Supreme is a fast paced and action packed thrill ride set in a cyberpunk future where players duke it out on their hover-cycles in the metaverse. Two players at a time battle in an arena, leaving deadly energy walls trailing behind their hover-cycles; which if hit are a one hit K.O.! Players collect shield power ups, use boosts, and avoid obstacles while battling in either single matches or tournament mode. With 3 difficulty modes, and randomly generated obstacles, Ray Bike Supreme has a lot of replayability!

Ray Bike Supreme Design Statement

Cyberpunk theme, bright blue/orange neon color palette, fixed non-scrolling game area, 1200px x 800px. Top down perspective, with keyboard controls.

Ray Bike Supreme Play-By-Play

Start up

- when the game first start-up to the start screen the player will input their name and click to select the difficulty level and match type
- A wicked sick guitar rift will also play during start-up and the title will have cool animations
- each player will input their name
- once two player have joined, they can click the start game button

Bike Selection

- Next, the hover-bike selection screen appears where each player selects their hover-bike

Gameplay!

- the game screen launches
- The grid and hover-bike for both player appears on screen. the grid-size growing or shrinking depending on the difficulty level selected, as well as rendering obstacles. a HUD displays each player count of shield powerups and amount of boost
- There are also help and quit buttons
- the game countdown starts from 3 and then both player start controlling their hover-bike
- as the hover-bike move through the grid they emit a energy-wall behind them, while players can move in 4 directions and boost around the grid using the keyboard controls trying to trap the other player into hitting a energy-wall, they also seek out shield powerups which allow them to temporarily pass-through an energy-wall
- if a player hitting a obstacles or energy-wall the round is over. Best 2 out of 3 round determines the winner and goes to the game over screen
- from there player can play again or quit

Ray Bike Supreme Ingredients List

Visible Elements (DOM Elements)

- div.game-container
 - div.start-screen
 - input#player-name
 - h1#game-title
 - button#add-player
 - select.difficulty-select
 - audio#start-theme
 - ul.players-joined
 - li.player
 - button#start-bike-select
 - div.bike-select-screen
 - img.hover-bike-blue
 - img.hover-bike-red
 - img.hover-bike-green
 - ul.players-joined
 - li.player
 - button#start-game
 - div.game-screen
 - div.hud
 - button.help
 - button.quit
 - div.player-1
 - 1. h3.player-boost
 - 2. h3.player-shield-count
 - div.player-2
 - 1. h3.player-boost
 - 2. h3.player-shield-count
 - canvas.game-grid
 - div.help-screen
 - p.game-description
 - p.controls
 - div.game-over-screen
 - h2.game-over-text
 - button.quit

- button.play-again

Invisible Elements

- class Player(name)
 - this.name
 - this.boost = Integer
 - this.powerups = []
 - this.isAlive = true
 - this.isShielded = false
 - this.activeLine
 - this.location = [x, y]
 - this.lineColor
 - this.playerBike
 - this.bikes = [redBike, blueBike, greenBike]
 - setPlayerBike(bike)
 - useBoost()
 - usePowerUp()
 - moveUp()
 - moveRight()
 - moveDown()
 - moveLeft()
- object game
 - obstacles = [will contain starting obstacles, then add player lines as drawn]
 - screens = start-screen, bike-select-screen,
game-screen, game-over-screen, help-screen
 - audio = SickGuitarRift.mp3, gameTheme.mp3
 - players = []
 - difficultyLevel = easy || hard
 - matchType = normal || tournament
 - startGame()
 - resetGame()
 - helpPopup()
 - gameOver()
 - switchScreen(screen)
 - setDifficulty(difficulty)
 - setMatchType(match)
 - addPlayer(name)
 - canvasFunctions
 - drawGame()
 - spawnPowerUps()
 - clearGrid()

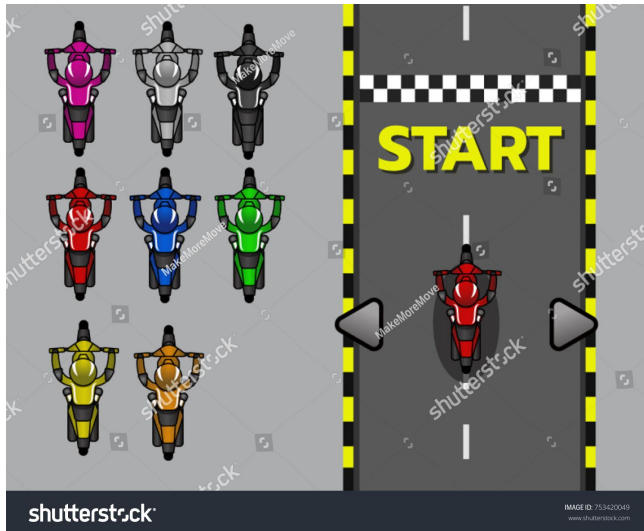
- drawGrid
- drawPlayers()
- drawObstacles()
- addObstacle(x, y)
- checkCollisions()
- checkCollectPowerUp()

Ray Bike Supreme Concept Art/Resources

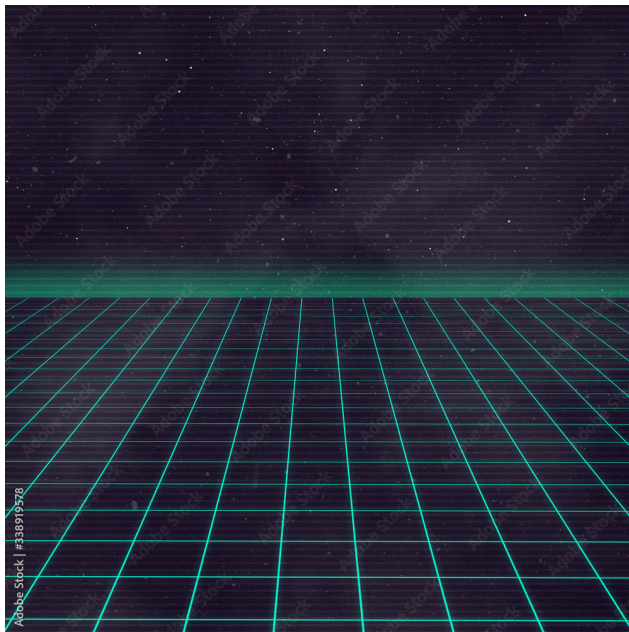


RAY BIKE
SUPREME

Logo Concept (made by me)



Bike Concept ("borrowed" from internet)



Game Grid Concept ("borrowed" from internet)

Questions / Comments

Questions

- Not sure whether I should have controls for the player in the player class or in the game?
- Not sure if the player should track their own active line then pass it to the game once it they turn and “finish” the line? Or if game should track 100%.
- Generally a little confused on whether I should have the bikes
- I currently have each bike object storing an array of all possible bike colors.. Perhaps that is better refactored to game as well?
- I did not include above the function to handle player death, should this be something added to the player class? Perhaps some kind of animation and sound effect..?

Comments

- Decided to create 4 movement commands (up, right, down, left) rather than having accelerate/decelerate and turn commands. I thought the 4 movement commands would make more sense to the user. My plan is for bikes to move at constant speed and be unstoppable similar to the game snake. Players just control the ability to turn. Hitting any obstacle (the 4 edges of the grid will also be made of “obstacle”) kills the player.