CS 426 Asgn. 5 - NeonBites Design Document

Game Title: **NeonBites**

Team:

1. Bianca Jankiewicz
2. Alexa Osuna
3. Pranav Mishra (Project Manager)

Tools:

- Version Control: Github https://github.com/bjank2/cs426_NeonBites
- Bug Tracking: Github https://github.com/bjank2/cs426_NeonBites
- Communications: Discord

Gameplay:

NeonBites is a thrilling cyberpunk food delivery game where players take on the role of a delivery boy / robot navigating the neon-lit streets of a futuristic city. Players must run, jump, and ride their bikes through the bustling cityscape, picking up food orders and delivering them across various locations. With ramps, shortcuts, and a dynamic driving mechanic, players must use their skills to navigate the urban terrain efficiently.

Features:

- Run, jump, and ride your bike through a neon cyberpunk city.
- Pick up food deliveries and transport them to their destinations.
- Utilize ramps, shortcuts, and dynamic driving mechanics to navigate the city.
- Use minimaps and routing systems to find the quickest routes to deliver food.
- Manage time, damage, and fuel, with the ability to refill at certain stations.

- Encounter traps and enemies such as roadblocks, thug areas, and crime hotspots.
- Avoid traps and enemies to prevent fighting for your package.
- Single-level gameplay is focused on timely delivery and earning rewards.

Avoiding Traps:
In order to get to their delivery locations, the player must get past a variety of obstacles and traps as they move across the city. These traps might be competitor delivery service roadblocks, criminal-infested alleyways, or crime hotspots where the player might run against hostile NPCs. The player may lose their delivery or have to fight to protect themselves and their package if they don't escape these traps.

Level Specifics:
The city itself is the main level, setting the scene for the player's delivery adventures with its neon-lit streets, tall towers, and expansive urban landscape. For the player to move quickly across the city and arrive at their goals on time, they must make use of ramps, shortcuts, and dynamic driving mechanics.

Time System:
An essential feature of the game is the time system, which requires the player to fulfill orders in a set amount of time in order to please clients and receive rewards. Managing time becomes crucial when the player must juggle several deliveries and navigate through the crowded streets of the city.

Rewards & Point System:
Players can upgrade their character, vehicle, and equipment by using the currency they earn from successfully making deliveries within the given time range. A player's ranking in the delivery app is enhanced by maintaining a high delivery success rate and customer happiness level, which opens up new options and incentives.

Overall Goal:
By earning money through precise and prompt deliveries, the player's ultimate objective is to effectively update and repair their robotic character. The user advances through the game, unlocking new challenges and rising to the position of top delivery robot in the cyberpunk city of NEON-Bites by overcoming time limits, dodging traps, and pleasing customers.

FEs:

- Interaction Pattern: Player vs. Game. There is no multiplayer, the player interacts with NPCs only.
- Objective: Successfully and accurately deliver food to customers (rescue, get the food to the customer safely)
- Resources: Currency, character/vehicle upgrades, health, delivery app ranking
- Conflicts: time, unhappy customers, malfunctioning self.
- Boundaries: Physical boundaries: Player is confined to city.

- Outcome: Successfully upgrade and repair yourself with money or exchange parts by completing orders accurately and timely.

Rules:

1. Rules for Defining Objects and Concepts:
    - Player character: The robotic food delivery driver is the main character, with attributes such as health
    - Orders: Different types of food orders represent the main objective of this game. Each order has attributes such as location, food type, and time limit.
2. Rules Restricting Actions:
    - Time Limit: Restrict the amount of time players have to complete each delivery
    - Resource Management: Money and fuel are limited, and you have to strategize how to effectively spend your money. Upgrades can be bought with money, and specific objectives can be completed for customers to exchange parts instead of money.
    - Combat: Characters can only attack when the customer turns hostile, or when the delivery package is threatened.
3. Rules Restricting Effects:
    - Damage and Health: Health is tied to the functionality of the robot, and this can be decreased by attack or lack of maintenance.
    - Customer Satisfaction: Restrict the effects of delivering orders successfully or failing to meet customer expectations, affecting tips, ratings, and future orders.

Level Design

1) Paper drawn sketch:

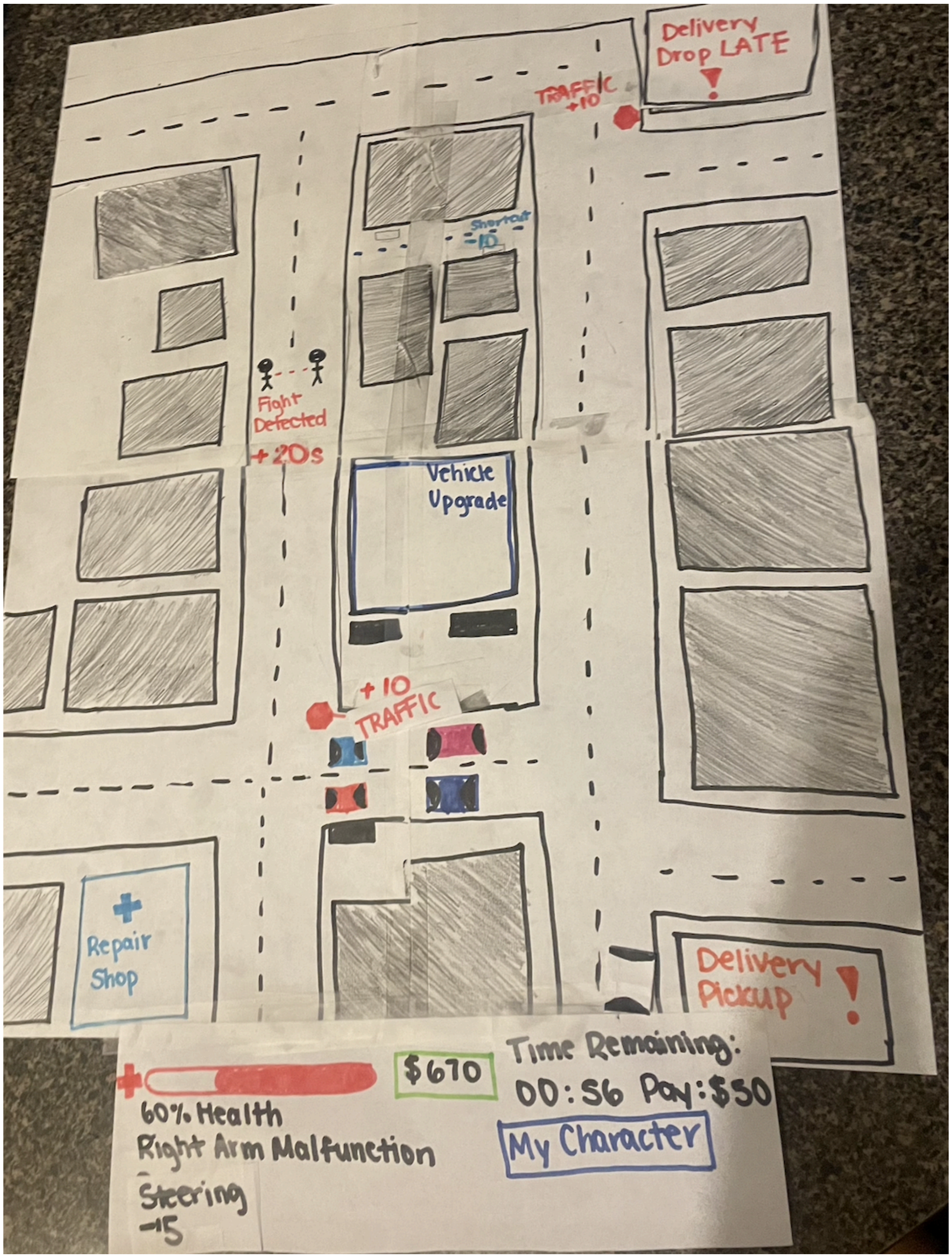

Figure 1: Rough sketch of city from a top view with UI elements as well

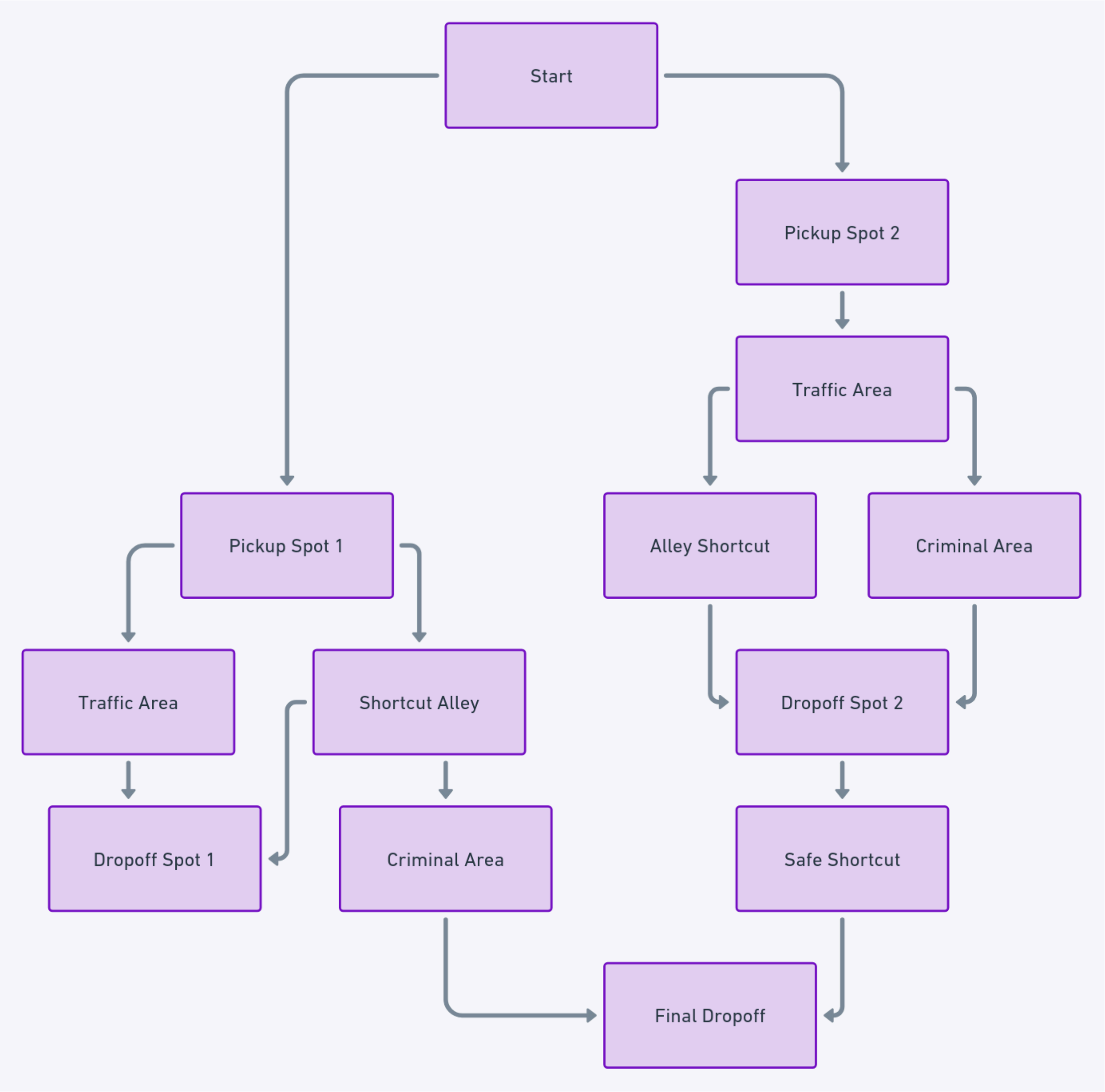

Figure 2: Flowchart of how a delivery might go by

2) Level Design Principles

Floor Plans:

- Design diverse terrain features that reflect the neon cyberpunk aesthetic.
- Include narrow alleys, expansive streets, towering skyscrapers, and hidden shortcuts.
- Create dynamic environments that encourage exploration and strategic navigation.
- Shortcuts are in place to help player make faster deliveries

Traps:

- Integrate competitor roadblocks, malfunctioning machinery, and environmental hazards.

- Place traps strategically to challenge players and add excitement to gameplay.
- Ensure traps are balanced to provide a fair challenge without overwhelming players.
- Traffic, or fights on the road can delay delivery.

Item Placement:

- Strategically position food delivery items and interactive elements throughout the city.
- Repair shops and charging stations are placed as well.
- Players must pick up and drop off orders they are going to deliver
    - Specific building allocated for player to pick up order
    - Different orders go to different drop off sites.

Power-Ups:

- Implement temporary advantages such as speed boosts, shield barriers, and repair kits.
- Distribute power-ups strategically to balance gameplay and maintain challenges.
- If player needs to upgrade vehicle, must go to specified location (Vehicle upgrade shop)

3) Level Components

1. Level Geometry
    - Buildings
    - Delivery order
    - Robot Models (Player/Customer/Enemy)
    - Dropoff Point
    - Road blocks
    - Repair shop
    - Mini-Map
    - NavMesh - for routes

2. Characters and Topology for animation

    Player Character - Delivery Robot:

    - Designed for efficient transportation of food orders within the city.
    - Follows different routes based on optimal delivery time.
    - Visits the repair shop when health is low or requires maintenance.
    - Visit Vehicle Upgrade Shop for vehicle enhancements.
    - Utilizes shortcuts to expedite delivery time.

    NPC Characters - Fighters:

    - Disrupt player's gameplay by causing delays in delivery.

        - Encounters with NPCs prolong delivery time for the player.
        - NPCs may engage in combat or obstruct the player's path.

    Vehicles - Traffic:

        - Adds to the challenges by creating traffic congestion.
        - Encounters with traffic vehicles increase delivery time.
        - Players must navigate through traffic to reach delivery destinations efficiently.

3. Character animations
    - Idle
    - Running
    - Getting on Vehicle - Bike
    - Getting off vehicle
    - vehicle crashes
    - Vehicle repair
    - Characters fighting
    - Satisfied/Unsatisfied emotions
    - Player jumping

4. Animations for Room, Puzzle & Artifact.
    - Order pickup
    - Order completion/Customer delivery
    - damage/repair of player/robot
    - Roadblocks

Timeline schedule

Week 1 (March 13 - March 19):

- Player Character Movement: Implement basic movement mechanics for the player character, including running, jumping, and bike riding.
- Pickup System: Develop functionality for the player to pick up food orders and items within the game environment.

Week 2 (March 20 - March 26):

- Minimaps and Routing System: Create minimaps and implement a routing system to guide the player to delivery locations efficiently.
- Time and Resource Management: Introduce time constraints and resource management mechanics, such as managing fuel and time limits for deliveries.

Week 3 (March 27 - April 2):

- Enemy Encounters: Implement enemy NPCs and obstacles for the player to encounter during deliveries, such as roadblocks and criminal-infested areas.
- Combat Mechanics: Develop combat mechanics for the player to engage in

fights with enemies when necessary, protecting themselves and their deliveries.

Week 4 (April 3 - April 9):

- Sound and UI Design: Integrate sound effects and background music into the game, as well as design and implement user interface elements for a polished player experience.
- Alpha Release: Prepare and submit the alpha version of the game, featuring basic gameplay mechanics and levels for testing and feedback.

Week 5 (April 10 - April 16):

- Testing and Bug Fixing: Conduct thorough testing to identify and fix any bugs, glitches, or gameplay issues present in the alpha release.
- Beta Release: Submit the polished beta version of the game with all levels and mechanics fully implemented, ready for final testing and feedback.

Due Dates:

- Level Design due at the end of class on Mar 13th
- Asgn5 (Design Document, Tools; Level Design and Sw Prototype) due or 03/13
- Asgn6 (AI and Mecanim out) due 04/03
- Asgn7 (alpha, sound, UI and testing) due 04/10
- Asgn 8 (Beta Release) due 04/17
- Asgn 9 (Public Demos) due 04/24
- Asgn10 (Completed Design Document and Personal evaluation) 04/24
- Asgn11 (Feedback) 04/24