

HACH Assignment 3: Mapping, Viewport, Interface, Inputs

I. Mapping

Beep! Beep! takes place in a bustling urban center, namely Seoul, South Korea, and our map design will simultaneously simplify and utilize common city sights and phenomena to create intuitive and immersive environments for the player to navigate using the player character. As the game progresses, each new level's map will expand upon the previous one, which can make it hard to assign specific levels to individual group members. Therefore, we are giving our group members the opportunity to design the map for one or more levels. We will then work together to identify any commonalities across the levels and come to an agreement on the final design. Figure 1 concisely demonstrates the player's aligned goal with the player character to reach a destination as quickly as possible. When designing each new level, please note that we will have a unique starting point for the player, potential obstacles that may hinder their progress, alternative paths that may present secondary objectives or activities involving the buddy NPC, and the player's ultimate objective or goal. These are important basic components that act as a foundation for understanding the rest of the map design decisions, and the ways in which they prompt the player to move.

Figures 2, 3, and 4 show the maps created by our group members, which depict buildings as white boxes that function as static, impassable obstacles on the map, with only a few exceptions that we will examine later. They share the purpose and behavior of randomly generated obstacles (represented by drawn X's in Figure 4), which parallel real-life road blockages and detours when driving (road work, NPC crowds, traffic, etc.). That said, the buildings are different from the road obstacles in the sense that they are a part of the map; obstacles change with each playthrough and can be considered a game mechanic that the player must play around, but the buildings are simply a constant element of the map that the player intuitively knows cannot be passed due to real-life navigation experience.

All of the maps created by our group members feature adversary NPCs, represented by blue dots in Figure 3, a red letter "P" in Figure 4, and blue dotted lines to show their patrol routes in Figure 2. Figures 2 and 4 depict example paths that police NPCs may take, which are not immediately apparent to the player. Additionally, Figure 2 highlights the possibility of a variable number of police patrols that may exist on the map, depending on the player's current level within the game. While officers are not innately hostile towards the player, the player must make decisions about when and where to drive fast in order to reach the destination faster. However, driving above the speed limit within the adversary's detection range will result in a time deduction, and thus their presence will increasingly inform and influence the player's navigation behaviors as they learn how the adversary functions.

Returning to Figure 1's side mission or NPC encounter, we have the NPC buddy/friend character represented in various locations in Figures 2, 3, and 4 that has the potential to prompt a side activity or mission that might yield a time bonus or reward for the player, or may even just provide some kind of assistance to the player directly. Therefore, the buddy's very presence on the map influences the player's navigation by encouraging them to seek out the buddy. Figure 4 expands on this by introducing the idea of a dynamically moving NPC buddy in later, more difficult levels that moves on a path similar to the police NPC. If these paths were to cross, it would introduce a risk/reward factor that encourages the player to reach the buddy quickly, but with the risk of being caught by the police in the same area. Each figure also illustrates the gas stations, which, like real life, influence a potential path a driver might take.

Another key element represented in Figure 3 specifically is the restaurant, which the player must visit to retrieve the items that are to be delivered. This creates navigational decisions before the game even necessarily starts; there could be a single fastest way to get to the final destination, but it is meaningless if you don't pass the restaurant to pick up the goods on the way. There are several factors at work in our map design that both help the player understand the game's boundaries of navigation, but simultaneously challenges the player to try new things and explore new paths to go as fast as possible despite the reality of these boundaries.

Considering the map diagrams in unison, we have agreed upon player starting at a random position, adversary NPCs patrolling around specific buildings, buddy NPCs appearing at higher levels with a movement path similar to the police NPCs, gas stations for refueling, and a restaurant for retrieving deliveries.

II. Storyboards / In-Game Viewport

Our storyboards and viewport mockups effectively allow us to take a look at the visual tools and information outlets that players will utilize in order to navigate and play the game. Our agreed view as of now is a 3-dimensional, third person view that includes several heads-up display elements that the player receives information from and uses to navigate the cityscape. Figures 5 through 8 all showcase potential gameplay views, showing the player and their vehicle centered on the view, with several locations, destinations, and collectibles visible. Figure 5 showcases an obstacle in-view, as well as a speed-boosting collectible. Similarly, Figure 8 shows a destination indicator (a star pointing to a circle on the road) conveying to the player where they should go to progress the game. Figure 8 also shows some alternate game states, such as the player stopping at a specific gas station location for gas, and a potential cutscene where an NPC sprite representing a happy customer thanks the player for delivering their goods.

Figures 6, 7 and 12 explore the aforementioned information sources that constitute the player's heads-up display, or HUD. The timer, fuel gauge, speedometer, and minimap are of

paramount importance, serving as the primary sources of information for the player's decision-making during gameplay. The timer, represented as a digital clock, informs the player of how much time they have used so far, informing the player's sense of urgency. The fuel gauge appears as a dial as one might expect to find in an average vehicle, and indicates how far the player can drive before a time deduction. The speedometer helps the player accurately control their speed when near police NPC adversaries, and is represented by a speedometer you might find in any vehicle's dashboard cluster. The minimap appears in an upper corner and is a mockup of a phone-based GPS navigation app, and it has several of its own indicators: the speed limit, which is relevant when near police officers, distance from final destination, and mini-indicators for police officers, gas stations, and the NPC buddy once it is in range. Figure 7 includes more mechanics on top of these that we will explore the implementation of as we move forward with development, specifically the mechanic of in-game order selection (another layer of decision making) and the connection of the scoring system to a monetary-based one, where different, higher-difficulty orders yield higher pay. Both of these mechanics are represented in Figure 7's HUD elements, and also potentially could introduce outside-of-game customization and decision options, such as vehicle cosmetics, speed upgrades, or possibly even alternative vehicles.

That said, our current gameplay viewport and HUD follows a third-person, three dimensional model that incorporates the primary elements of vehicle information, time/score information, and map/hazard information for communicating the game state to the player and prompting them to respond and play accordingly. As we move forward, we will experiment with the amount of information shared with the player, the type of information shared with the player, and potentially new mechanics to fit into these categories of communication.

III. Graphical User Interface

Our user-end interfaces prioritize simple, linear, yet immersive design in accordance with *Beep! Beep!*'s premise and gameplay. Let us examine our prototypes for an open game start menu. Figures 9 and 10 introduce fundamental start menu options that prompt the earliest stages of user interaction: an option for playing the game from the beginning, an option for some kind of continuation (saved game/level select), Settings (graphical, audio, etc.), about/help menu (gameplay instructions as seen in Figure 10, developer information), and an option to quit the game. Figure 11 effectively expands on these ideas by introducing a dynamic and immersive background to the game's start menu, and visually including potential orders for the player to take throughout any of the available levels. The immersive background shows the player character in their half-underground apartment at night during stormy weather with their vehicle outside, grounding the player in the character's reality and making clear their motivations throughout the game.

Figure 10 explores the next level of player navigation beyond the start menu, showcasing a potential model for both an about/help screen as well as a level selection screen. Again, these

elements are streamlined and simple, allowing the player to easily get into the action: the about screen shares the input layout and game premise with the player in a visually welcoming tab-based menu, and the level selection screen concisely lists the levels and allows the player to choose one with a single input. These screens could also expand on the immersive background of Figure 11, causing the background to dynamically shift between different sights in the player's life, such as other rooms in the apartment building, when moving to different menu screens.

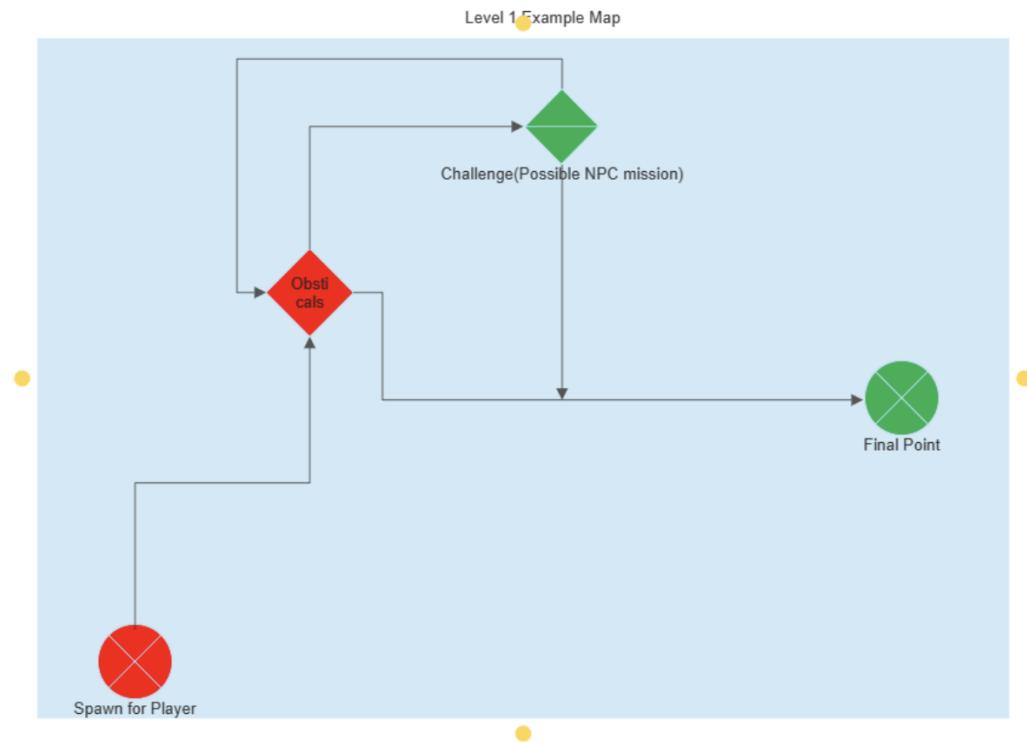
The primary components on which our interface is focused are immersion and streamlined simplicity; the goal is to lead the player through a menu. The start screen represents the first impression the game has on players, and therefore our combination of simple UI design as well as an immersive background and appearance is crucial to ensuring players have an accurate read on the tone and expectations of our game.

IV. Input Layout

Our game's inputs are relatively straightforward—in Figures 13-16, the primary controls can be identified as the arrow keys and mouse. The left mouse button is utilized for selection and menu navigation during menus and cutscenes, while the “WASD” keys are used in-game for acceleration and steering. Holding W is used to accelerate, S is used for brakes and moving backwards, while A and D are used for steering left and right, respectively. We also have the spacebar representing some potential alternative interactions, such as a handbrake (Figure 16) or secondary accelerator (Figure 14) for sliding or drifting, adding some extra mechanics and depth to our game's driving. The spacebar could also be used as a button to specifically interact with any collectibles a player has picked up (Figure 13), or potentially as a secondary accelerator (Figure 14). If we choose to include both of these mechanics, one could alternatively be mapped to the shift key. Generally, our controls remain simple and intuitive, and play into expectations many players will already have about games that include driving mechanics.

V. Reference Figures

Figure 1 - Haotian's Map Prototype:



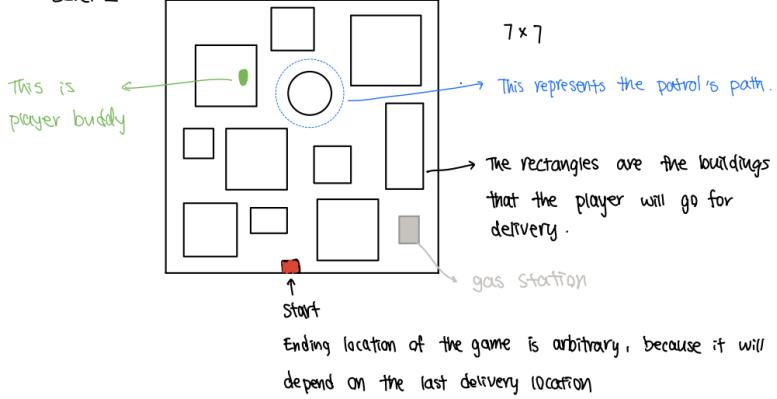
In the above diagram, the player spawns at the spawn point; then he follows the path to the first obstacle; there is two paths after this obstacle, one is straight to the finishing line, and one is to another challenge. This challenge will have an NPC with a mission or question to ask; if you pass, you keep driving to the final point; if you fail, you go back to the previous obstacles.

Figure 2 - Anqi's Map Prototype:

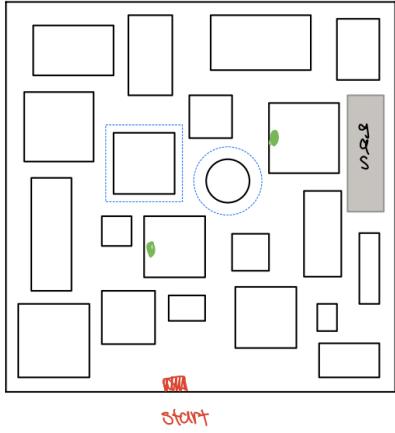
A. Mapping :

① Elevation view of our game view:

Level 1:



Level 2:



Level 3:

larger map
14x14
+
3 patrol
path
+
3 player
buddies

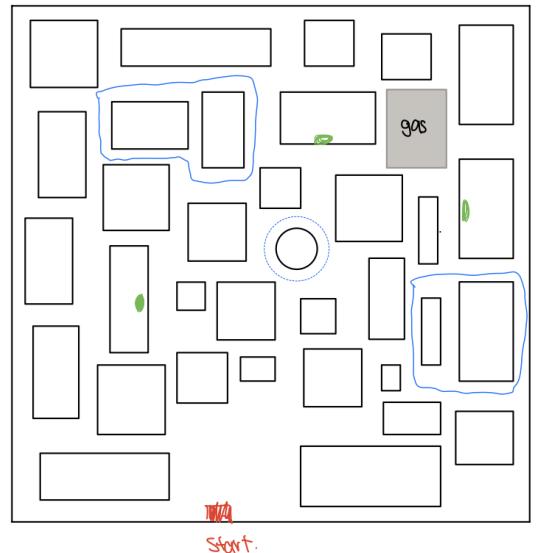
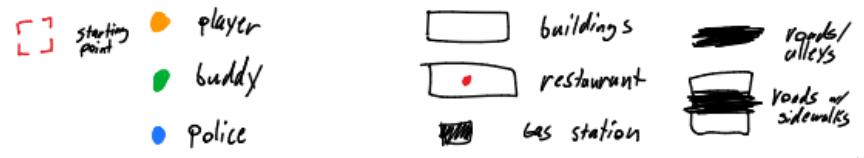
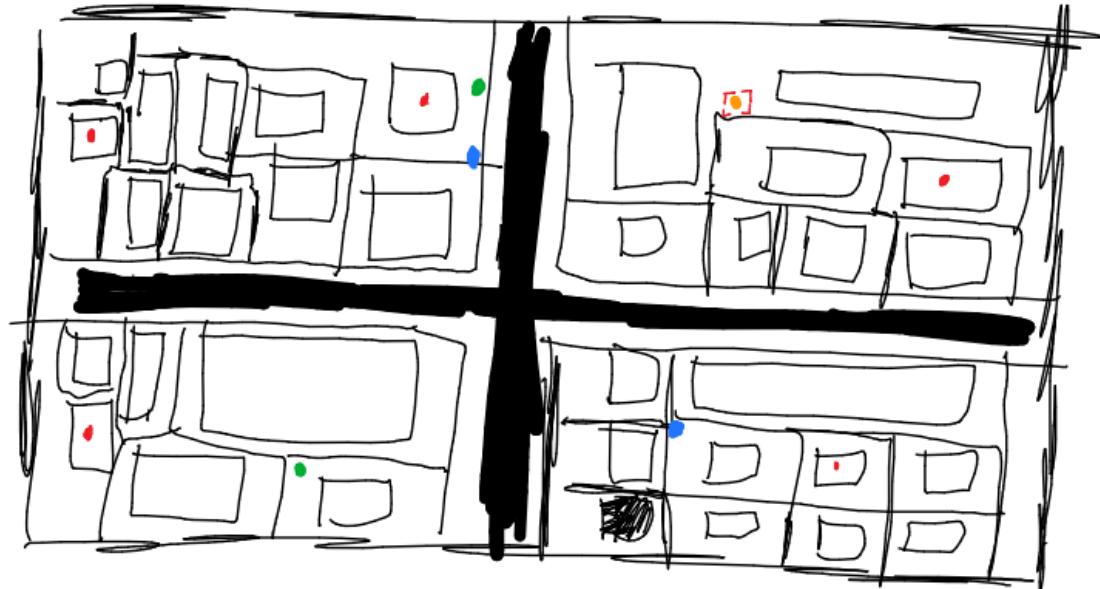


Figure 3 - Hunter's Map Prototypes:

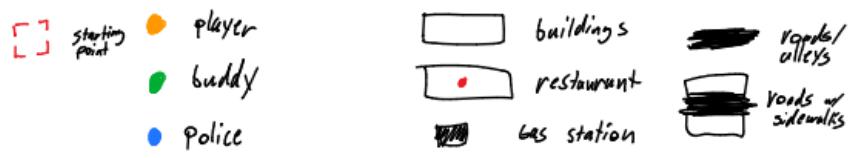




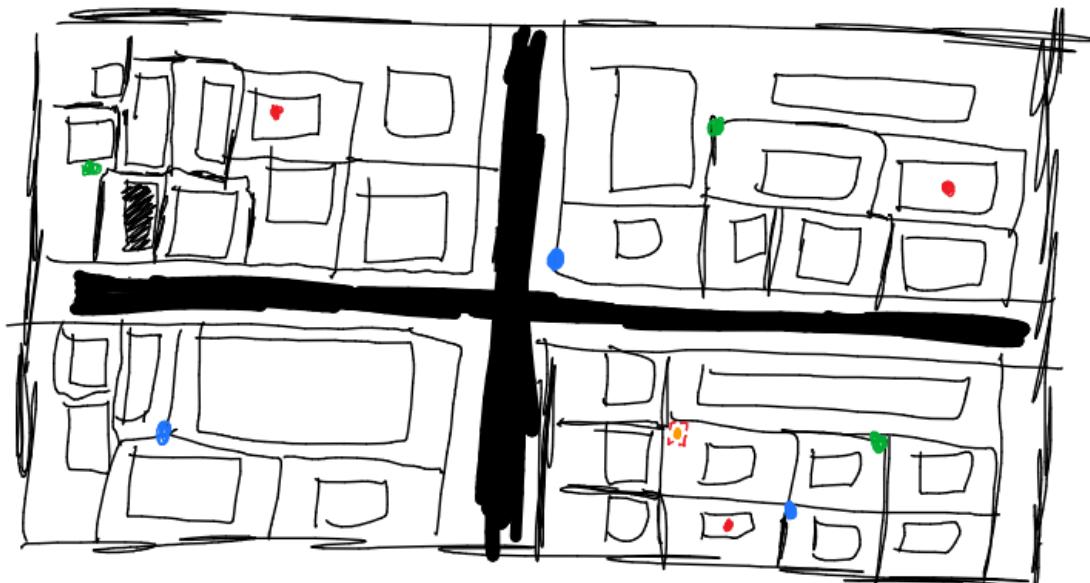
Level 2



9x16



Level 3

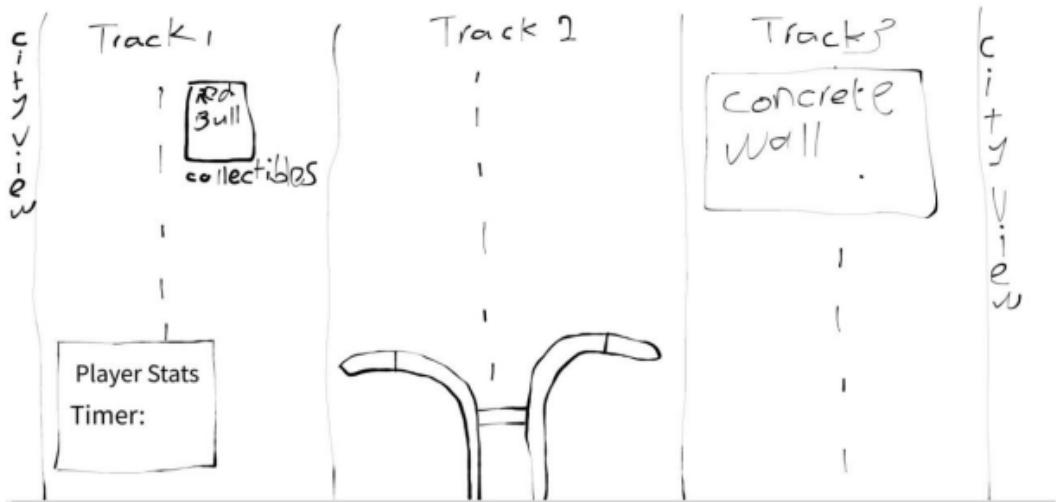


9x16

Figure 4 - Cor's Map Prototype



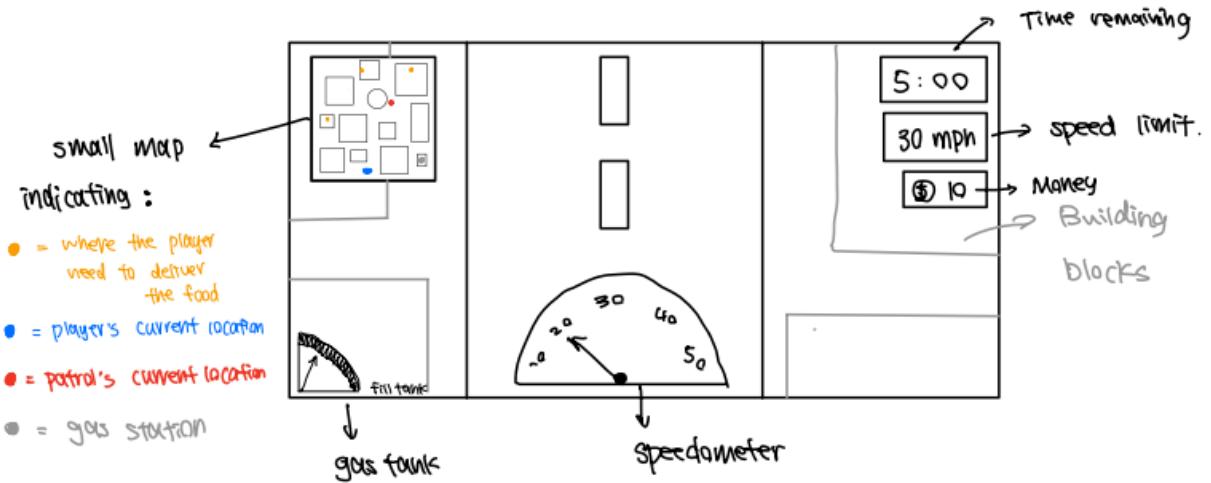
Figure 5 - Haotian's Storyboard / Viewport:



This is a first-person viewport in 2d. The player is centered with his ride in the middle at all times; the player can observe their statuses such as their name, timer, and level info. There are three tracks; you can use the arrow keys on the keyboard to move the bike left or right and fast and slow. There will be obstacles that block one or two tracks. There will also be a collectible that boosts speed or cancel times.

Figure 6 - Anqi's Storyboard / Viewport:

Player's view in game :



* not going to show where's the buddy
because it is for the player to find it.

Figure 7 - Hunter's Storyboard / Viewport:

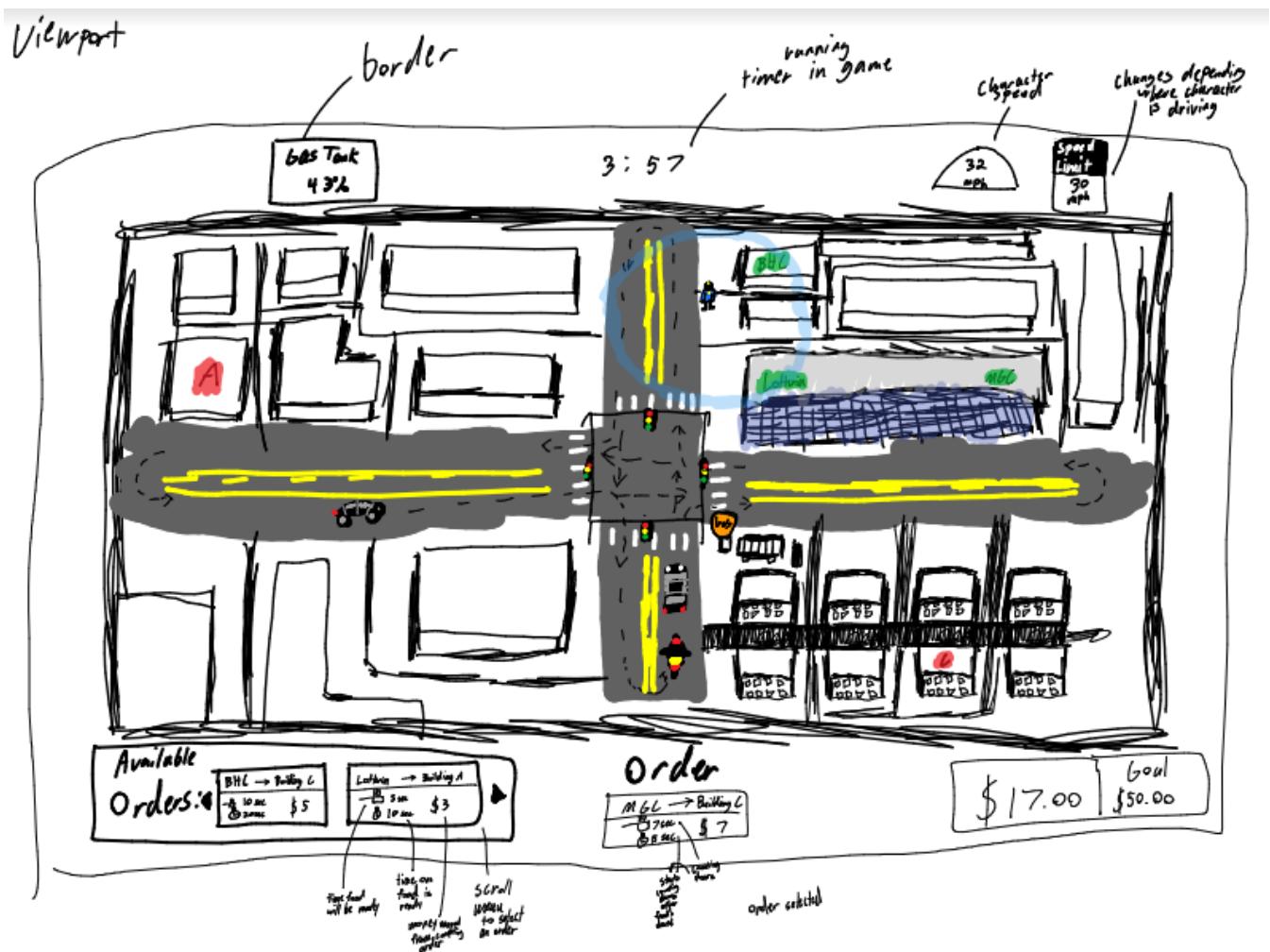


Figure 8 - Cor's Storyboard / Viewport:

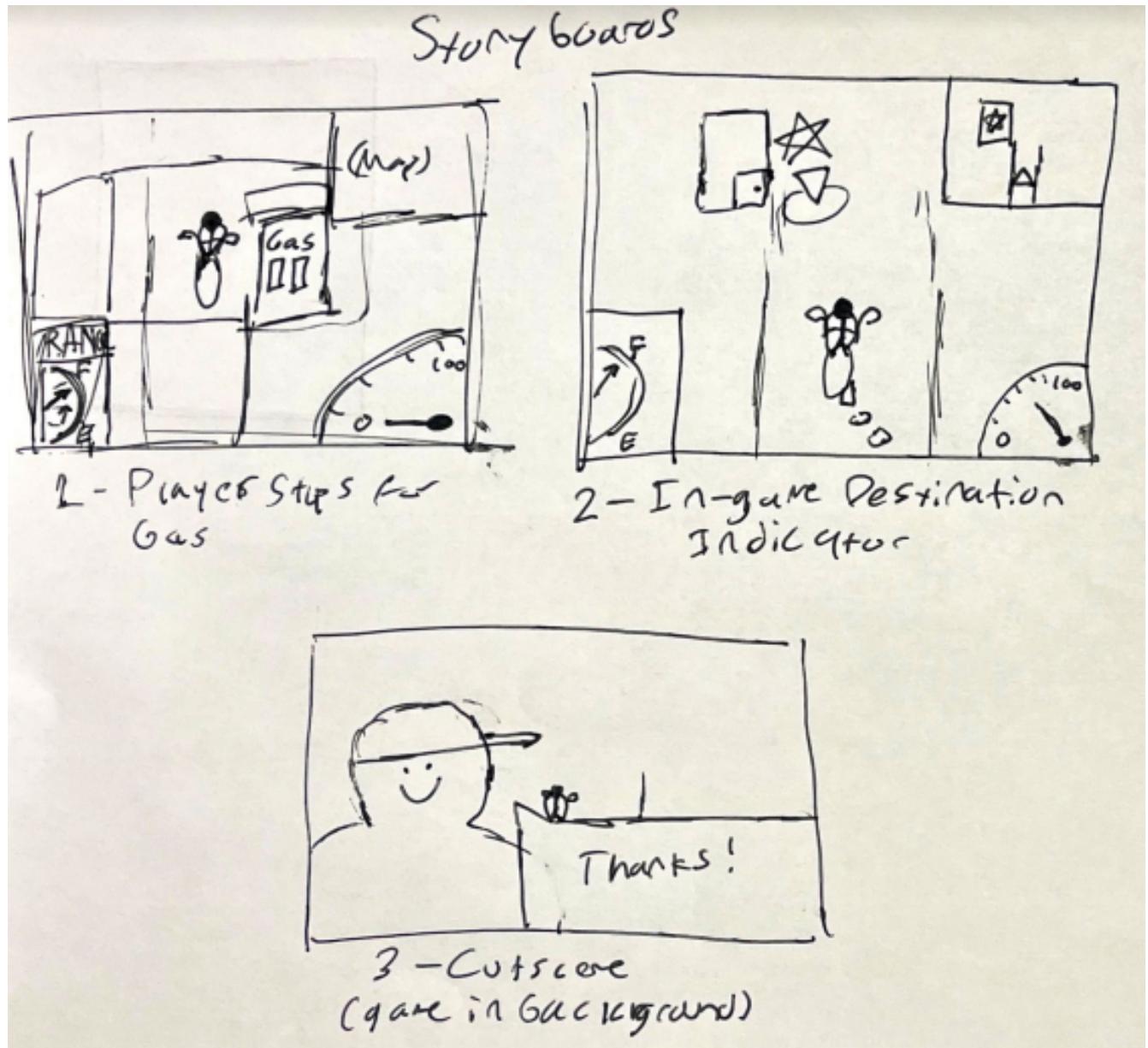
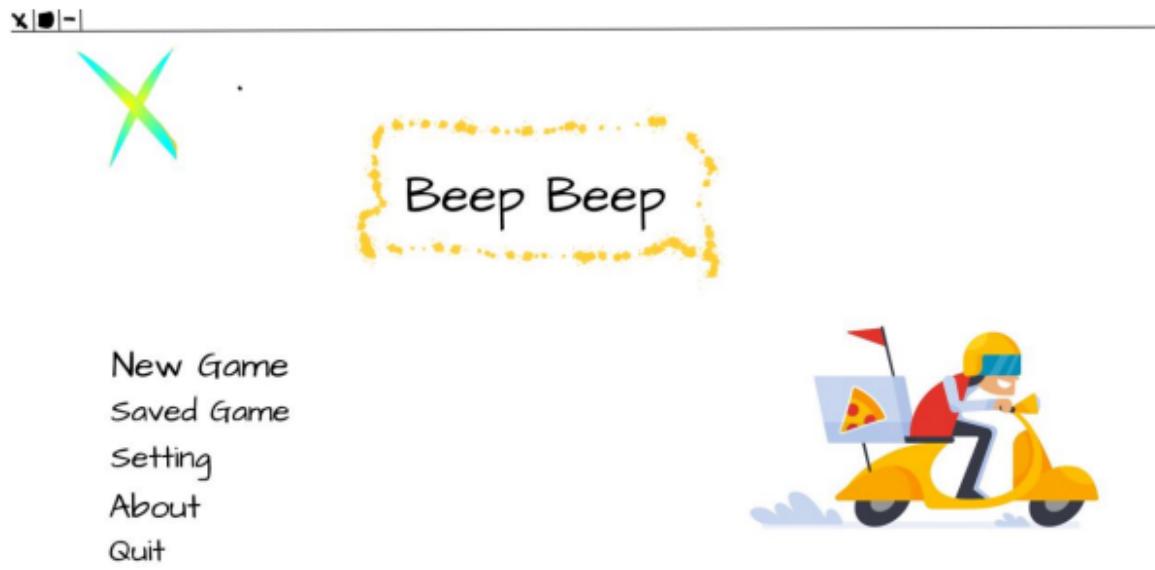


Figure 9 - Haotian's HUD/GUI



This is the open interface of the game.

Figure 10 - Anqi's HUD/GUI:

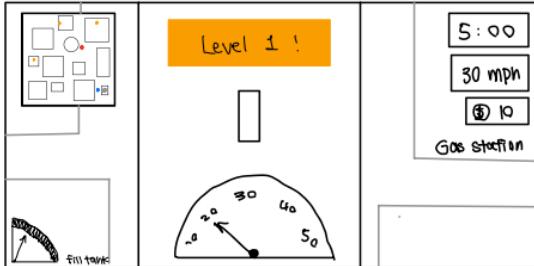
C. Interface

MENU PAGE

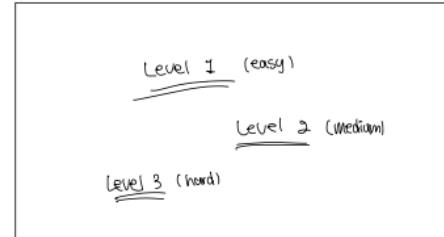


If the player click "Start new game", it will automatically start at level 1.

This will lead to :



If the player click "select level", he/she will be directed to:



If the player click 'help', they can read about the overview of the game's objectives, and learn the keys that will be using in the game.

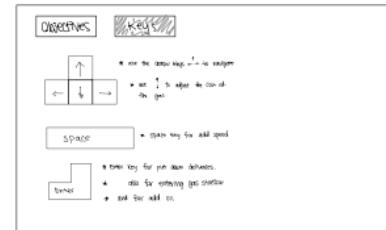
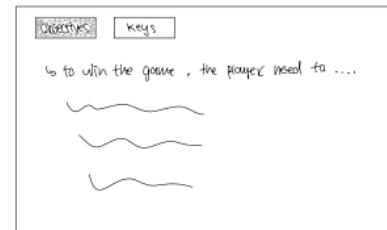


Figure 11 - Hunter's HUD/GUI:

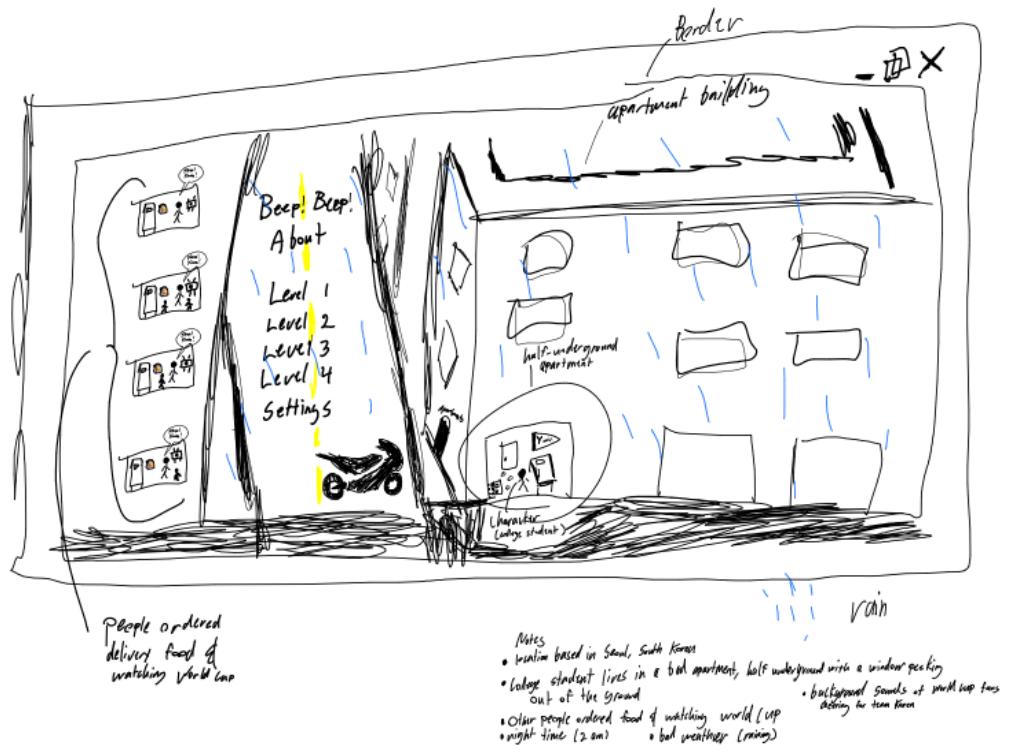


Figure 12 - Cor's HUD/GUI:

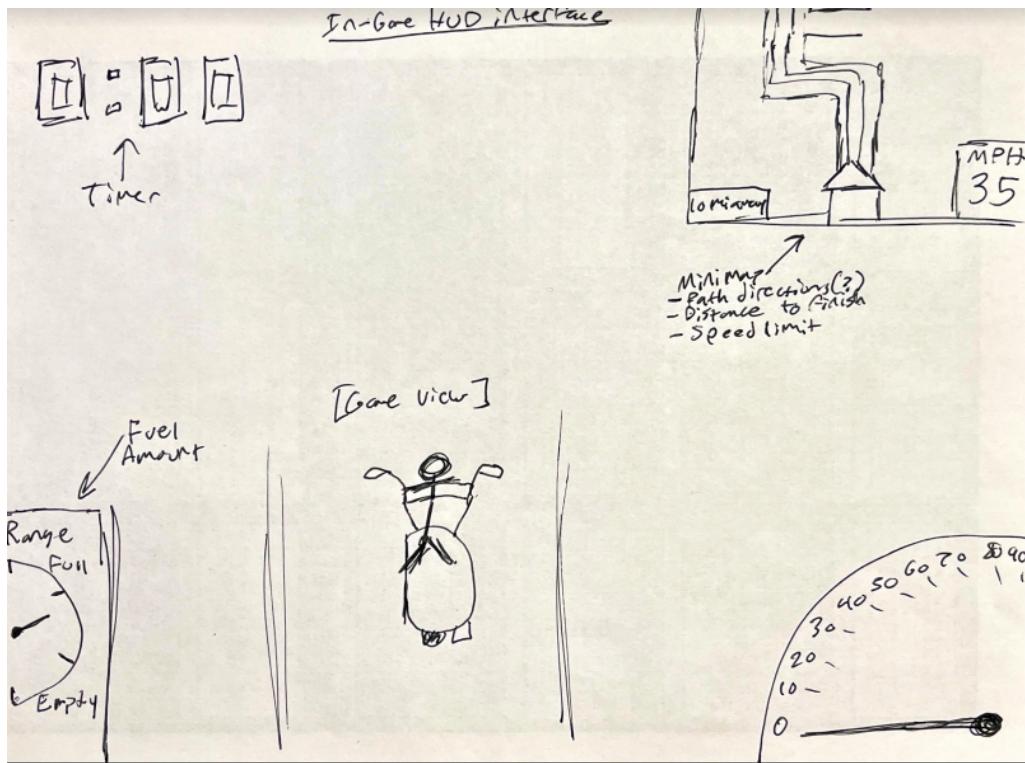


Figure 13 - Haotian's Input Layout:

Input part of the game
The game will need a mouse and a keyboard to play
Left mouse click – to receive input from the user at the begging user GUI, or quit the game
Keyboard
W – speed up
A – left turn
S – right turn
D – slowdown (beak)
Space – Use Collectibles.

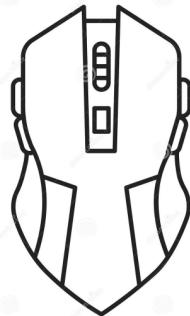


Figure 14 - Anqi's Input Layout:

4. Inputs

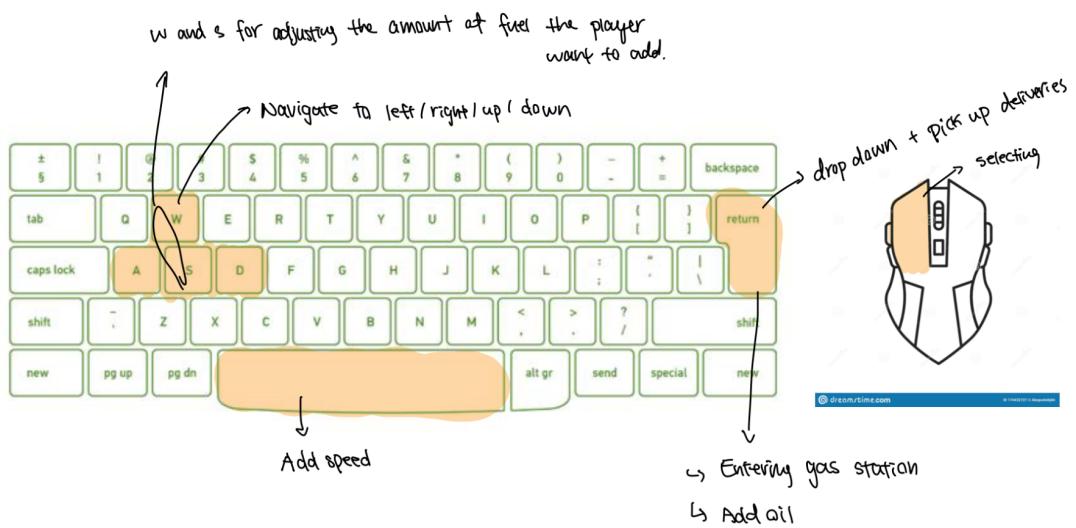
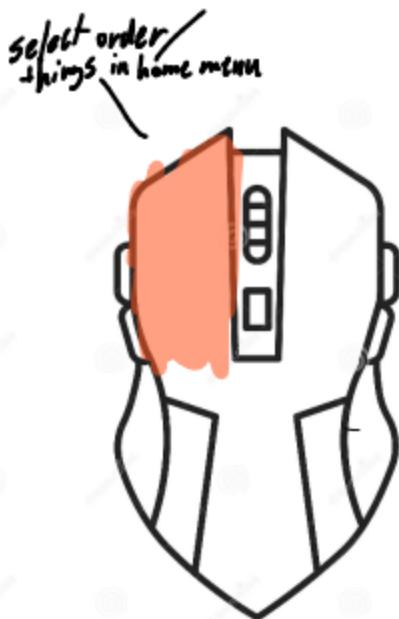


Figure 15 - Hunter's Input Layout:



© dreamstime.com

ID 174432127 © Alexpoloskih

Go North/Up

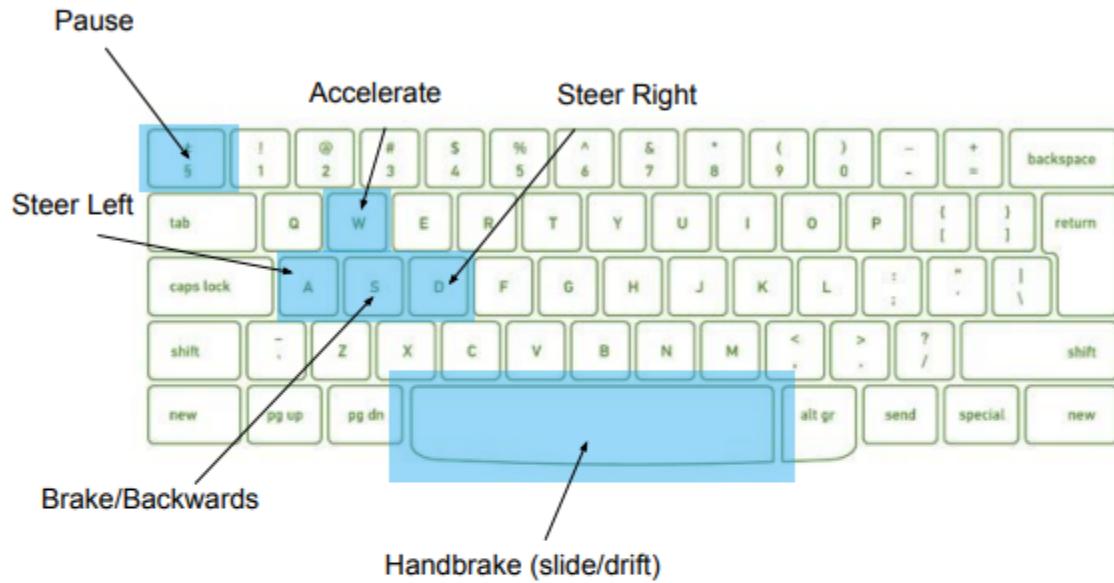


- holding down button = accelerate
- to brake = hold button of opposite direction

note
- character will decelerate slowly if it is in motion and no buttons are held down

Figure 16 - Cor's Input Layout:

Input Diagram



Confirm>Select

