University of Calgary
CPSC 585 — Winter 2019 — Games Programming

Hover Wars

High-Concept Design Document

Team A — Light Theme is for Heretics
Austin Easton, Evan Quan, James Cot, Jianan Ding
January 21, 2019

Contents

1 Introduction

Hover Wars is a combat-based driving game aimed to test your skill to fight against your enemies. Whether playing alone against AI or with friends, each player find themselves driving a hovercraft in an arena pitted against each other. Utilizing abilities, picking up power-ups, and the navigating the map, everyone must destroy each other in a chaotic battle of strategy and wits before the round is over.

The central constraint of the proposed features is development time available to implement them. As a result, the following game features are subject to change as development progresses.

2 Gameplay

2.1 Player Count

The game supports single and local multiplayer, allowing for 1 to 4 players. Local multiplayer would be split-screen multiplayer and would require multiple input devices (XBOX controllers).

2.2 Game Modes

The main game mode proposed is **free-for-all**. At a low priority, other modes may be developed if enough time is available.

Free-for-all

The game is composed of a single round that lasts for the duration of a timer. Each player is placed in an arena to control a hovercraft with various abilities. They must fight each other in a free-for-all battle with those abilities, amidst a number of neutral bots roaming the arena to hunt down players.

The central goal of the game is to gain the highest score possible before the round is over. Players have the following means to gain score:

- Damaging other player hovercrafts, which can only be done in multiplayer. This can be through the use of abilities accessible to every player.
- Destroying other player hovercrafts, which can only be done in multiplayer. If damage is done to a hovercraft's final hit point and it is destroyed, extra points are awarded.
- Destroying bots, which can be done in both single and multiplayer. Bot hovercrafts do not have all the abilities of player hovercrafts and so reward less points.
- Picking up power-ups. These help the player gain points by improving their abilities, but also innately give points when they are picked up.

Secure the Intel

Similar to free-for-all, except a single player is designated to have the "intel". Only this player is able to score points, meaning that players are incentivized to hold the intel for as long as possible to gain score. If another player destroys the player holding the intel, that player then takes the intel and the game continues.

Co-op Survival

Players must team up against a never-ending swarm of bots. Players only have one life before they are out of the game. When all the players are destroyed, the game ends.

2.3 Arenas/Maps

The game will feature a single map. Given the development time frame, a single fun and polished map is preferable to multiple mediocre maps. It should be "small" to "medium" in size, to maintain a high player density to ensure that players are always in the middle of the action, and do not get lost.

If time allows for additional maps, they may be added at a lower priority, but the focus will still be on a singular main map. These additional maps should provide a different purpose than the main map. For example, there could be a small map designed for 1–2 players. This will lower the downtime as the action with be closer and more compact.

Primary Environmental Features

These are basic features that are core to the map's design, and so play a high priority in their implementation.

- **Barriers/obstacles** — These can block movement and projectiles. These can take the form of various things: street lamps, walls, buildings, tunnels, or other structures.
- **Ramps** — Ramps lead to higher and lower platforms, or can be jumped off of over obstacles. This can provide better vision of certain parts of the map, give opportunities to jump between arenas.

Tentative Environmental Features

These features are open to be cut, either due to time constraints to implement or potential game design problems they introduce.

- **Speed pads** — Driving over these will give a momentary boost of speed. Designed for getting power-ups quicker, and chasing or escaping other players.

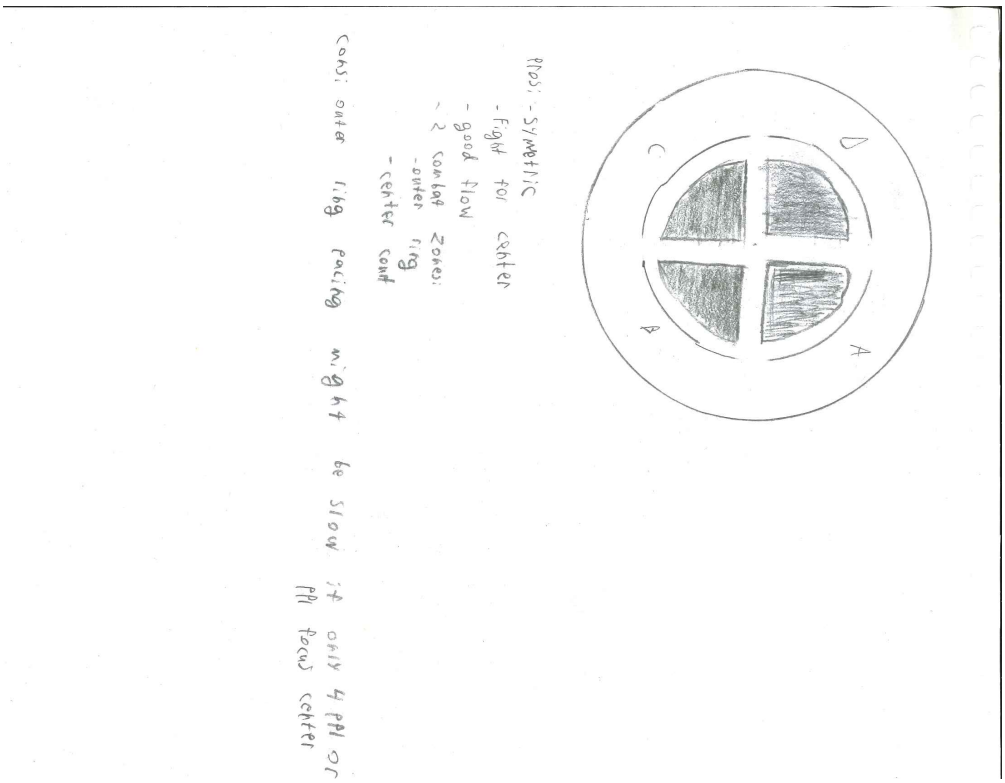

Figure 1: A sketch of the map.

Potential problem: While these make sense for a racing game, they may detract from the gameplay for a combat arena game. Players will typically want full control over their vehicle and driving over a speed bump in locations where they don't want it can be annoying. If a speed boost mechanic is to be implemented, it may be better to have speed boost power-ups instead of speed pads.

- **Pits** — Falling into the pit will instantly destroy the player hovercraft. Avoid at all costs or try to bump enemies into it.

  Potential problem: Depending on the placement, size and frequency of the pit, players can be easily frustrated if they are constantly falling in it. Instant death mechanics such as this can also break the flow of gameplay.

- **Jump pads** — Driving over these, will give a vertical boost while maintaining horizontal momentum. Designed for jumping over obstacles or reaching areas of higher elevation.

  Potential problem: Similar to speed pads, players may not actually want to use them when traversing across the map. It can potentially be very disorienting to players to use jump pads, especially if they lose control of their movement while in the air.

It is important to note that more doesn't always mean better. Even if the idea of certain map features are cool or interesting on paper, if they detract from the overall gameplay experience in practice, it is better to leave them out. There is value in a good but simple map design.

2.4 Players

Players drive a hovercraft. They innately have full access to a number of movement options and abilities from the start of the game.

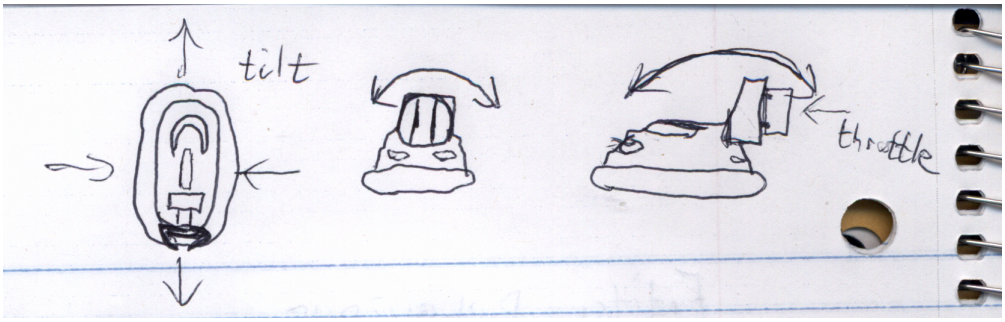

Figure 2: Sketch of the player hovercraft with a more mechanical, used-future design. The hovercraft should tilt based on its movement.

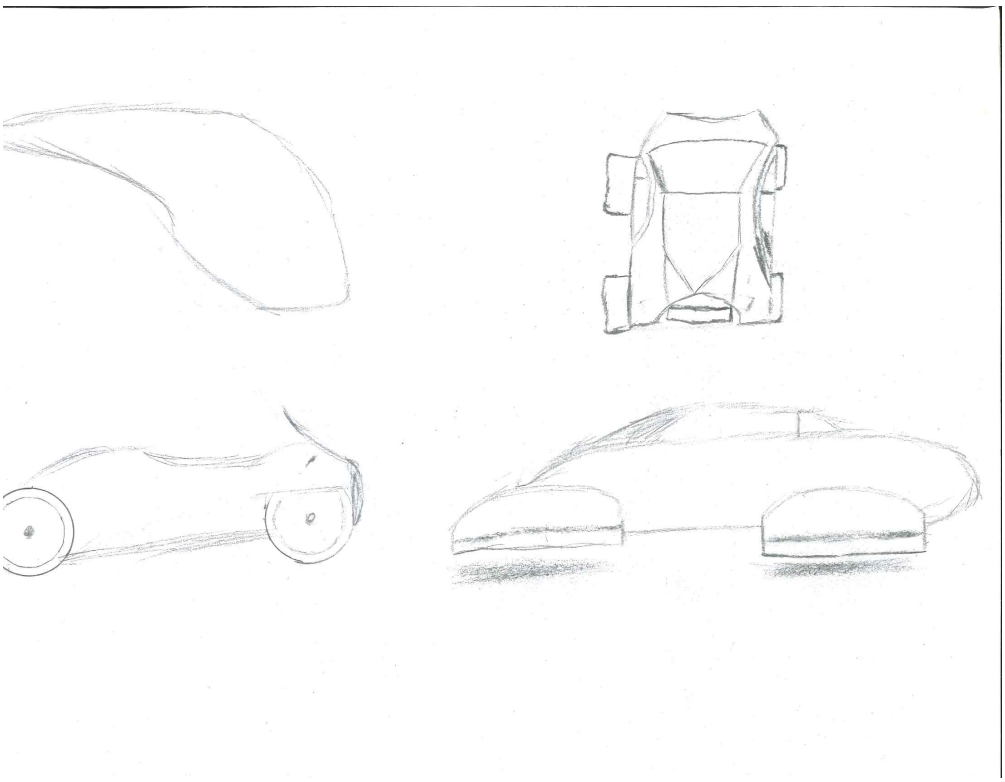

Figure 3: Sketch of the player hovercraft. The hovercraft should tilt based on its movement.

2.4.1 Health, Lives, and Damage

There are a few health and damage systems we have considered. We are currently committed to the hit and continue system, but will be open to change as we play-test.

Hit and Continue

Modelled after **Mario Kart**'s damage system, vehicles are blown away or spun around when they are hit by abilities, temporarily making the player hit lose control of their vehicle. Shortly afterwards, the player regains control and is given a few seconds of invincibility before continuing on playing. This minimizes the downtime during play since there is no respawning, while still punishing players that get hit.

Lives with Hitpoints

Players start the game alive with a small set amount of hit points, which will be explicitly displayed to the player. They can be damaged by the abilities of other hovercrafts, lowering their current hit points. The same temporary loss of control will occur as the hit and continue system. When all hit points are removed, the player's hovercraft is destroyed.

When a player's hovercraft is destroyed, they are momentarily out of the game before respawning randomly at one of the respawn points on the map. If another player destroyed said hovercraft, that player is awarded points for the kill.

One Hit One Kill

Similar to the lives and hitpoints system, but players are instantly destroyed upon impact of an ability. This makes getting hit more punishing and further promotes players to dodge abilities.

2.4.2 Movement

- **Standard movement** — A hovercraft does not rely on wheels to move and so can traverse in any lateral direction without needing to turn, meaning that strafing is possible.
- **Acceleration/braking** — A hovercraft can accelerate an brake in any direction is it currently moving. It will often drift if a turn is made, even at relatively slow speeds, which can be both advantageous and disadvantageous. This is open to change as movement is play-tested to figure out what is fun and feels right.
- **Dashing** — A hovercraft can dash in any direction. From a mobility standpoint, dashing can be used to catch up to other hovercrafts, reach power-ups faster, or lose others when being chased. From a defensive standpoint, it can be used to dodge attacks.

2.4.3 Abilities

Every hovercraft has 3 attack abilities that are available from the start of the game.

- **Rocket** — A rocket launches forward straight out from the direction the hovercraft is facing until it hits a surface. Upon impact, it explodes, damaging everything in a radius around it. Being the only ranged attack, it is great for attacking distant enemies if aimed well, or when chasing other vehicles. The splash damage can be utilized with parts of the arena environment to hit enemies near walls easier, or to hit multiple enemies that are grouped together.
- **Spikes** — Spikes temporarily extend in all directions from the hovercraft, damaging other vehicles that come in contact with it. Can be used both aggressively and defensively when other vehicles are nearby. It can also be used in combination with dashing to crash into enemies.

- **Trail** — A trail is created behind the player that follows the players path. Any hovercraft that contacts it is damaged. Great to use when being chased. Visually, this trail may be composed of flaming napalm, light energy, or some toxic substance, which is still up for discussion as development continues. Players can dash through the trail unharmed.

2.5 Bots

In all modes, a number of bots will be present on the map. With a visually distinct design from players, bots will only target player hovercrafts. They are weaker than players in that they have fewer hit points and fewer abilities, but also award less points on kill. As a result, it is up to players to decide much they want to focus on destroying bots versus other players in their strategy.

The capabilities of bots is open-ended as development progresses. Here is a tentative priority list of bot functionality, from highest to lowest priority.

2.5.1 Movement

It will be a requirement for bots to be able to target players to chase them. They must be smart enough to recognize the presence of obstacles and general map geometry to reach players in a reasonably direct path if possible.

At a lower priority, they may be given movement abilities or have more advance movement decision-making such as:

1. Speed boost. Bots will speed up towards players when a path is clear between the two. This will help them use their spikes to crash into the player for damage.
2. Guard power-ups. Bots will recognize when power-ups are on the map and priotize attacking players who they believe are going for them.

2.5.2 Abilities

To work with their basic chasing behaviour, the bots will, at minimum, be equipped with the spikes ability, allowing them to ram into the player for damage. Whether the spikes need to be activated or are always enabled will be up to play-testing.

At a lower priority, they may given other abilities like the players:

1. **Rockets** — At its simplest, bots would be able to aim at players to fire rockets at them. More sophisticated, bots would be able to lead their shots based on the player's predicted trajectory and the rocket trajectory. They may also utilize walls to hit players with the rocket's explosion radius. Bot rockets may potentially travel slower than player rockets.
2. **Trail** — Bots will try to intercept the player's path to get them caught in the trail.

2.5.3 Friendly fire

Since the bots team up against the players, any damage they would deal to each other would be purely accidental. At first glance, it would seem reasonable to disable friendly fire so that bots would not destroy each other en masse. However, friendly fire amongst bots may create in-game situations that would add extra chaos, excitement or even humour to the game's atmosphere. If need be, the bot count (or respawn timing if hovercraft destruction is implemented) would be adjusted for balance to account for these friendly kills.

Ultimately, the decision for allow for bot friendly fire will be up to play-testing, and we will be open for it to work either way.

2.6 Power-Ups

Throughout the map, there are set power-up spawn locations that will randomly spawn one of several power-ups. Players can pick them up by contacting them. Upon contact, the player that picked it up will temporarily receive passive bonus for a set duration.

At minimum, we want there to be 3 total power-up types. While there is no particular priority in what power-ups to implement, we would ideally have one power-up for each player ability. Here are some ideas:

Rocket

- **Rapid-fire rockets** — Rockets can be launched at a faster rate of fire. Good for spamming at a distance.
- **Homing rockets** — Rockets will slightly home towards the closest target. Good for chasing.
- **Explosive rockets** — Rockets will have a much larger impact radius. Good for taking out groups of enemies in small corridors.
- **Multirockets** — Shoot multiple rockets at once.

Spikes

- **Large spikes** — Spike size increases, hitting enemies are further distances.
- **Longer spike timing** — Spike duration is increased. of the power-up.
- **Bumper spikes** — When enabled, the player is immune to other players' spikes.

Trail

- **Wide trail** — The trail is much wider, easing the ability to catch opponents in their tracks.
- **Passive trail** — The trail continuously lasts for the duration of the power-up.
- **Slowing trail** — The trail slows the movement of enemies crossing it, making them vulnerable to attack.
- **Blocking trail** — The trail cannot be bypassed, blocking enemies trying to cross it. Great for defense when being chased, or aggressively to trap enemies in an area.

Other

- **Repair** — The player gains an extra hit point.
- **Speed boost** — The player gains a speed boost.
- **Shield** — The player is invulnerable to abilities.
- **Invisibility** — The player is invisible to other players and will not be targeted by bots.

2.6.1 Spawning

There are three power-up spawning systems we believe are workable, each with their pros and cons:

1. **Random Unknown** — When a power-up spawns, it is unknown to the players what power-up type it is until it is picked up.

Pros: This can add a sense of surprise to the players as they find out what power up they have. It can also change up the gameplay as players adapt their strategy to best utilize whatever power-up they have gained. This prevents players from just taking a single type of power-up.

Cons: May not be as interesting as the other systems.

2. **Random Known** — When a power-up spawns, it is randomly chosen from the list of possible power-ups and is known to the player what power-up type it is by its appearance.

   Pros: This can add an extra layer of strategy as players can scout the map for power-ups they they want, if certain ones work better their for their playstyle or current strategy.

   Cons: If certain power-ups are deemed useless or underpowered, then no players may pick them up. Over the duration of the round, all power-up locations may eventually be filled up with those power-up types as players ignore them.

3. **Deterministic by location** — Certain parts of the map deterministically spawn certain power-up types.

   Pros: This can add an extra layer of strategy as players can gravitate towards certain areas to get certain power-ups they want.

   Cons: Certain areas of the map may be neglected if they have bad power-ups.

As it stands, we are committed towards the **Random Unknown** spawning system, but will be open to change as we play-test.

2.7 Difficulty

Single Player

In a single player experience, the difficulty can arise from a competitive approach in achieving a high-score. Whether one is attempting to outperform their previous high scores, or compete with others' high scores, players can improve their skills and learn new strategies to improve. This self-imposed motivation to improve and compete can create new levels of difficulty at a meta game level.

Multiplayer

Similar to single player, difficulty arises from the skills of opposing players. As other players improve, so does oneself need to do to so to compete. New strategies can arise in using abilities, power-ups, and parts of the map to maximize points, as well as strategies to counter other players' play styles.

2.8 Menu

3 Game Design

3.1 Aesthetic

Visually, the game mainly follows a cyberpunk and TRON aesthetic. We are going for a futuristic city at night appeal.

3.2 Inspiration

Hover Wars is inspired from a number of other games:

- **Akane** (2018) — aesthetic

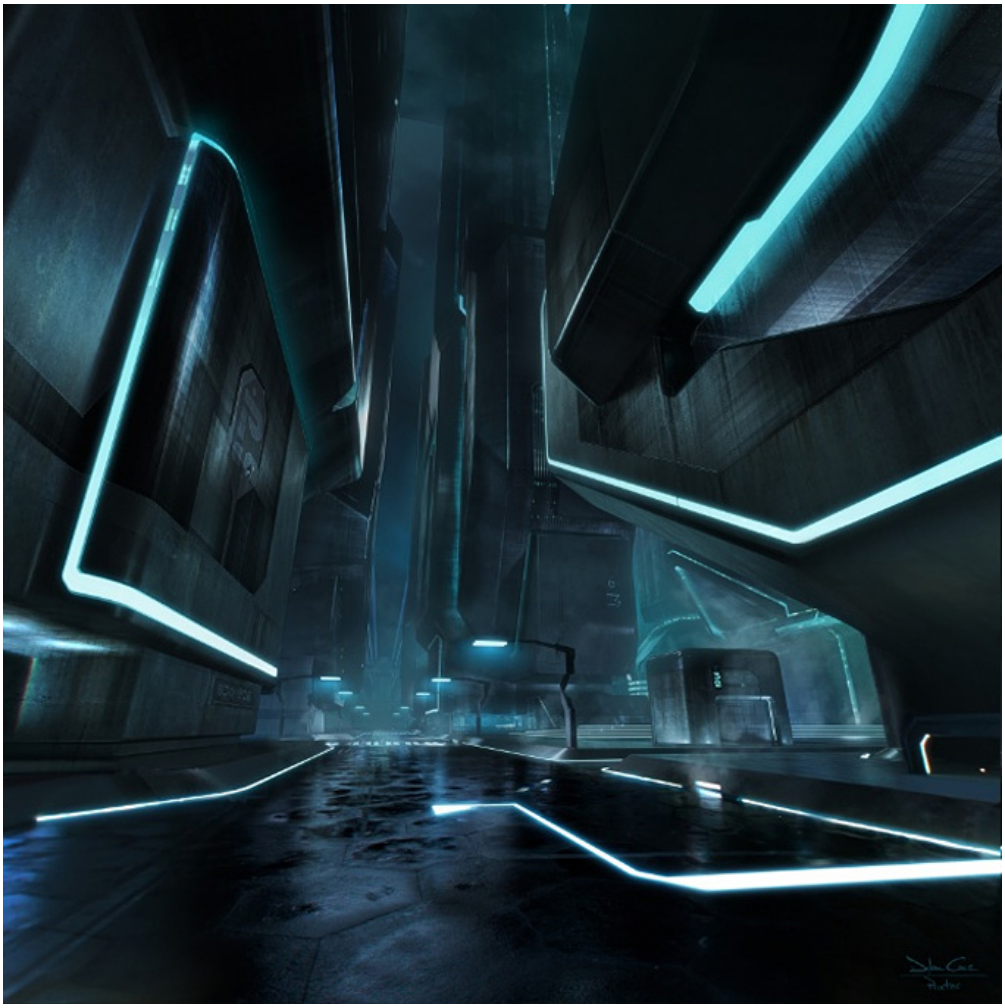

Figure 4: Taking place at night, there will be a focus on artificial lights from buildings to illuminate the area.

Figure 5: Colourful neon lights will play an important role in creating a sense of warmth.

Figure 6: There may be a contrast of lighter and darker arenas in the map, which may be used to highlight important areas such as power-up locations, or ramps leading to different areas.

- **Counter-strike** (2000) — point system
- **Crawl** (2017) — Secure the Intel game mode
- **DOTA 2** (2010) — point system
- **Mario Kart** (1992) — free-for-all game mode and general combat design
- **Pac-Man** (1980) — bot versus player design
- **ThinkTanks** (2003) — Speed and jump pads
- **Tron** (1982) — aesthetic and trail ability

3.3 Designer Insight/Goals

Here are some goals we have in mind for the project, as well as some insight behind our design decisions from our team discussions.

3.3.1 Vibe

Playing **Hover Wars** should feel exciting and bring a sense of hype and energy, similar to combat games like **Super Smash Bros** or **Street Fighter**. This can be brought about through fast-paced gameplay, coupled with action-packed sound effects and music.

3.3.2 Role of AI

The introduction of AI-controlled hovercrafts (bots) adds an interesting element to the game, but also a few problems.

First, given our past experience and the time-frame creating the AI, we don't believe the bots will be equally competent to a skilled human player. If bots are given a hovercraft with equal capability to that of a player, it is unlikely they will be able to utilize their abilities and movement sufficiently to compete with players, or have sufficient game sense to outplay and counter different play styles. This poses a problem for single-player, as competing against a group of underperforming bots will not particularly fun or challenging.

To address the challenge issue, it is possible to give bots point bonuses when they score to improve their chances to beat players. However, this does not necessarily address the fun issue, as the player will still experience fighting against simple bots.

Instead, bots can be given an alternative role rather than replacing a player. By explicitly giving them less capabilities than the player, and having them exist in-game independent from the player count, they can add an extra depth to the gameplay without heavily relying on the depth of their capabilities. The benefit is that if the bots end up less capable than we initially planed, the combat can still be just as fun since bots will be in greater numbers and team up against the players. If they are more capable than we initially planned then all the better, since this will simply make the gameplay more engaging.

Overall, the focus of multiplayer will be on the interactions between players for the multiplayer experience, where the bots will provide a background or secondary element to the gameplay. For single player, the bots will be core to the gameplay, as they are the only enemy against the player.

3.3.3 Driving System

Since players are driving hovercrafts, the driving experience should model that of a hovercraft. As a result, the driving model should feel somewhat floaty, allowing for easier drifting. Without the constraint of

wheels, the players should be able to move in any lateral direction without needing to turn, allowing for strafing.

However, if the hovercrafts are too floaty, they may be frustrating to control. Driving needs to feel responsive, especially if sudden turns or braking are done. Our decision to add the dashing mechanic partially solves this problem, allowing for players to make sudden movements to combat times when they may feel out of control.

Our goal is for there to be a balance in the driving system for it to feel somewhat floaty to imitate a hovercraft, and yet grounded enough to feel fun and responsive. As we play-test the movement throughout the development, we will iterate on this balance, and may even radically change how driving feels if we believe it is best for the game.

3.3.4 Learning Curve

Easy to Learn

A core goal for game is for it to be easy to pick up and start playing. Part of this involves controls that are intuitive to new players. While there are a fair number of abilities, they are the same for everyone, meaning players do not need to know the ins-and-outs of different vehicle abilities that they themselves do not have access to.

Power-ups should feel intuitive to understand and use. They should not introduce new mechanics or keybindings. It is frustrating for new players to "waste" power-ups in order to understand what they do, especially if they are single-use. Instead, power-ups should augment already existing abilities and clearly display in the UI which ability is improved.

Hard to Master

Players should be given opportunities to improve and apply their skill. Each ability is distinct and requires its own skills to use.

3.3.5 Health and Damage

We designed our health and damage system similar to **Mario Kart**'s. There are a few goals we had when choosing this approach for our system:

- It is easier to understand how much health the player has in "real terms." With a health bar or a percent gauge, while players have some rough idea how many hits can can take, it can still be somewhat fuzzy if abilities deal different amounts of damage, or if players need to roughly calculate how many more hits they can take from their past experience of how much damage abilities do. By keeping the hit points to a small discrete value, players immediately know how many more hits they can take a moment's glance.
- It eases user memory load. Players do not need to concern themselves with the stats of various abilities with how much damage they each do. By keeping it simple, players simply need to know that getting hit with an ability will remove one of their hitpoints.

3.3.6 Abilities

In designing our abilities, we have several goals in mind. For each one included, we want to make sure that they:

- **Feel distinct to use:** They should employ different strategies to use in order to feel unique. Abilities that are too similar can be boring.

- **Serve different purposes:** Multiple abilities that are best-used for the same situations creates redundancy. Either both abilities are equally useful in those situations, making having both unnecessary, or one is better than the other, making the other obselete.
- **Introduce some level of counterplay:** If skilled enough, players on the receiving end should be able to avoid getting damaged if they react appropriately. Not being able to react to abilities can feel cheap and frustrating, especially if they recognize what ability is about to be used and feel helpless to respond. As a result, we have avoided all hitscan weapons as they cannot be doged and can be difficult to anticipate.
- **Are deliberate in their use:** Due to the health and damage system we plan to implement, abilities are viewed more more from a "hit or miss" view rather than "receive various levels of damage" view. So just as abilities require attention to identify and avoid from the receiving end, they should also require a similar amount of attention to use from the aggressor's end. Players should have to identify their targets and employ some amount of skill to correctly hit their enemies, whether it involves aiming, colliding with or maneuvering around enemies. As a result, we have avoided abilities with high rates of fire that are easily spammable, or "fire and forget" weapons that do not require much thought to use.

These goals will always be in mind if we decide to create additional new abilities or tweak the existing ones later down the road.

3.3.7 Blue Shell Effect

In the context of this game, the blue shell effect can be described as *"a mechanism that evens the playing field between players that are performing well and those who are not"* in reference to the infamous blue shell item in **Mario Kart**. It gives players that are behind a chance to comeback in order to make games closer. Most multiplayer games implement some level of blue shell effect in their game mechanics, some through "better" means than others. There are a few common rationales for why this is desirable:

1. This can make the game more fun for less-skilled players by easing their ability to compete with higher-skilled players (hopefully not in an over-corrective manner if tweaked properly).
2. In games where players' abilities or stats can improve over time due to player actions/decisions (through items, power-ups, etc.), those who are ahead can easily snow-ball out of control, as their current power lead can further help them create an even bigger power lead. Blue shell mechanics can help combat snowballing.

A blue shell mechanic we want for the game would be to track player kill-streaks, which is the number of kills (hits) they have done to other players in a row without getting hit themselves. As the kill-streak increases, so does a bounty on that player. If another player kills that player with a bounty, they are awarded extra points. This makes players that are performing well higher priority targets, while not giving them a direct gameplay disadvantage that seems unfair. We may experiment with other systems as we play-test if we feel like it could be improved or better balanced.

3.3.8 Performance

- **Hover Wars** should run at 60 fps for single player and 30 fps for 4-person multiplayer.

3.4 Market Competition

The main competition with **Hover Wars** in the market is Nintendo's **Mario Kart** franchise, which **Hover Wars** is heavily inspired from. As a fairly popular franchise, we hope to differentiate our game by emphasizing the combat aspect, as **Mario Kart** is primarily racing game first.

Another game that would compete with **Hover Wars** would be **ThinkTanks**, developed by GarageGames in 2003, which also plays as a vehicle-based battle arena game. Our edge over **ThinkTanks** would be the greater ability diversity and faster-paced gameplay, which will hopefully draw a greater appeal.

3.5 Game Genre

Hover Wars is a third-person combat-based driving game. It is designed to be a fun party game that is easy for new players to pick up and play, while giving the opportunity to those who want to master it the means to do so given.

It is developed for the PC, supporting Windows as a high priority and Mac and Linux with lower priorities, using mouse and keyboard controllers. It will also support XBOX 360 controller support, allowing for multiplayer modes.

3.6 Branding

Hover Wars is a new IP on its own.

3.7 Target Market

While violence is a core component of the gameplay, nothing is particularly graphic due to the use of vehicles and the lack of blood and gore. We do not intend there to be any mature themes in the game. We therefore believe that **Hover Wars** is appropriately targeted for all ages 10 and above years of age.

4 Concept Art

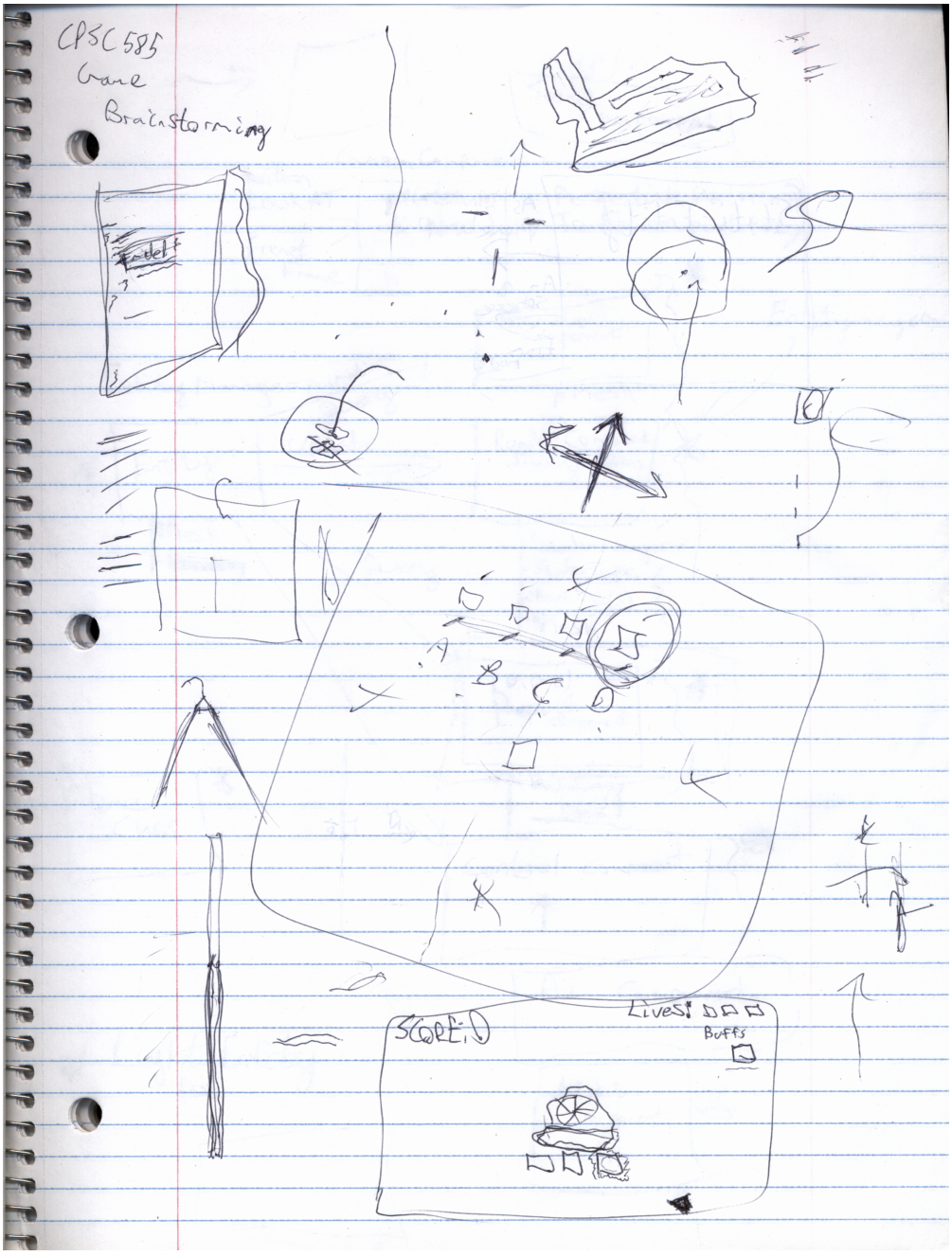

Figure 7: Rough sketches of the map, UI, and hovercraft

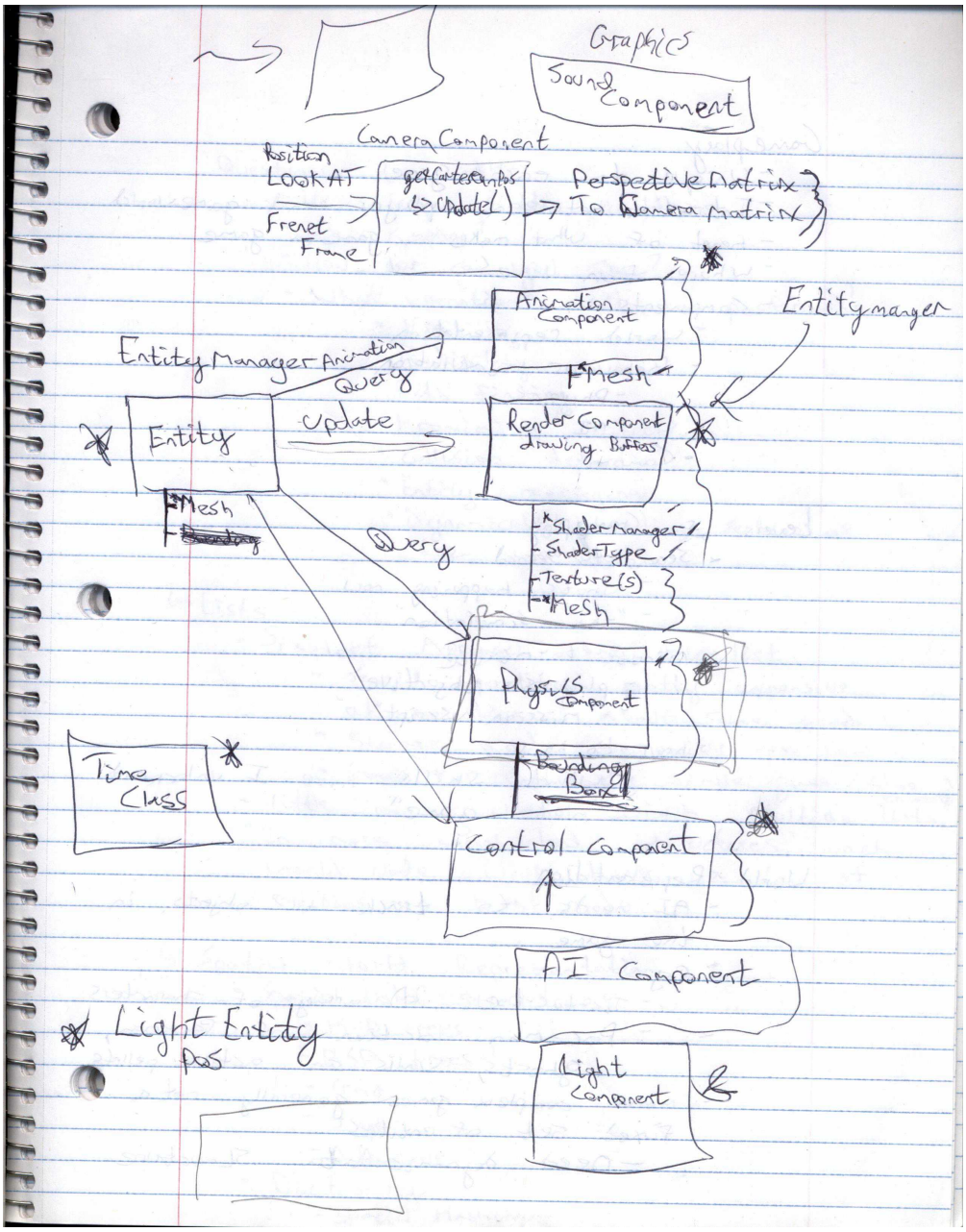

Figure 8: Early design of the game application framework