

University of Calgary
CPSC 585 — Winter 2019 — Games Programming

Hover Wars
High-Concept Design Document

Team A — Pressurized Studios
Austin Easton, Evan Quan, James Coté, Jianan Ding
January 21, 2019

Contents

1	Introduction	1
2	Gameplay	1
2.1	Player Count	1
2.2	Game Modes	2
2.2.1	Free-for-all	2
2.2.2	Secure the Intel Variant 1	2
2.2.3	Secure the Intel Variant 2	2
2.2.4	Secure the Intel Variant 3	2
2.2.5	Free-for-all Survival	2
2.2.6	Co-op Survival	2
2.3	Arenas/Maps	3
2.4	Hit Points, Lives, and Damage	4
2.4.1	Hit and Continue	4
2.4.2	Lives with Hit Points	4
2.4.3	One Hit One Kill	5
2.5	Players	5
2.5.1	Movement	6
2.5.2	Abilities	6
2.6	Bots	6
2.6.1	Movement	6
2.6.2	Abilities	7
2.6.3	Friendly fire	7
2.7	Power-Ups	7
2.7.1	Spawning	8
2.8	Difficulty	9
2.9	Menu	10
3	Game Design	10
3.1	Aesthetic	10
3.2	Inspiration	12
3.3	Designer Insight/Goals	12
3.3.1	Vibe	12
3.3.2	Role of AI	12
3.3.3	Driving System	13
3.3.4	Learning Curve	13
3.3.5	Getting Hit	13
3.3.6	Abilities	14
3.3.7	Blue Shell Effect	14
3.3.8	Performance	15
3.4	Market Competition	15
3.5	Game Genre	15
3.6	Branding	15
3.7	Target Market	15
4	Concept Art	15

Multiplayer

Similar to single player, difficulty arises from the skills of opposing players. As other players improve, so does oneself need to do 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.

2.9 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.

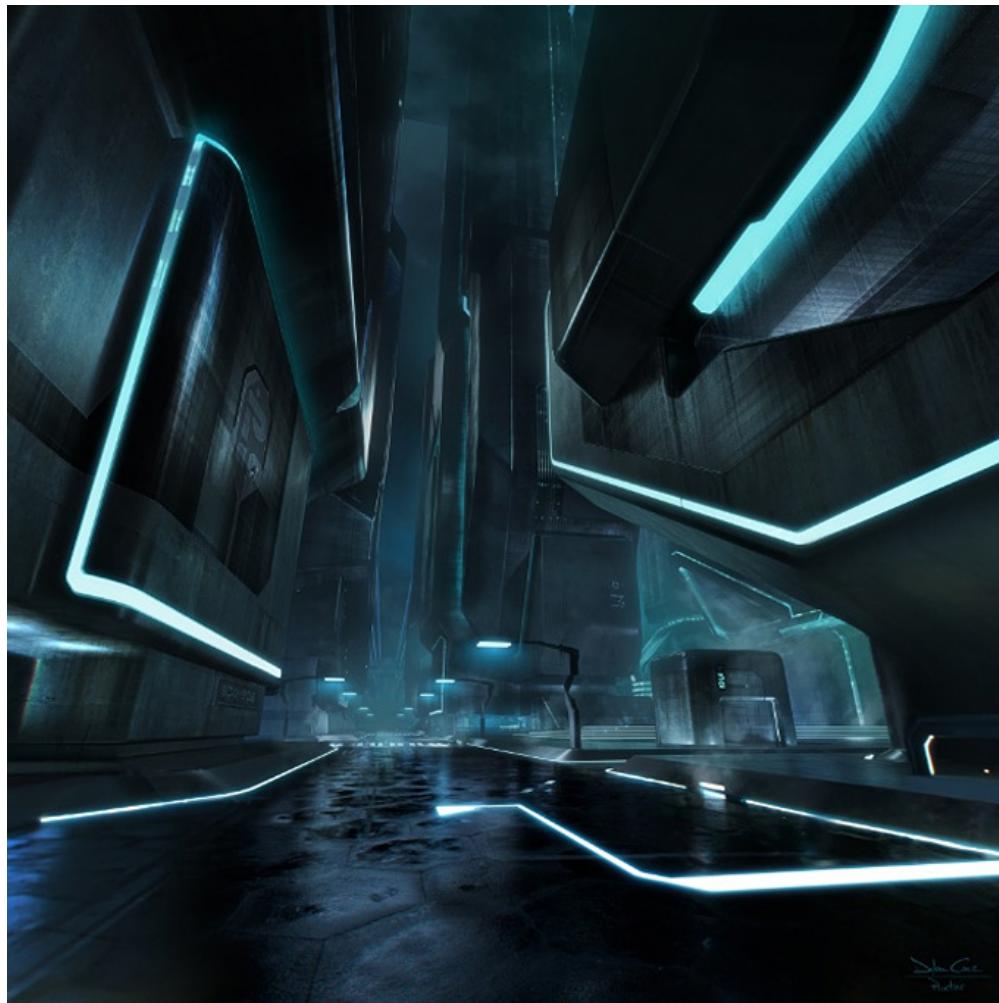


Figure 6: Taking place at night, there will be a focus on artificial lights from buildings to illuminate the area.



3.2 Inspiration

Hover Wars is inspired from a number of other games:

- **Akane** (2018) — aesthetic
- **Counter-strike** (2000) — point system
- **Crawl** (2017) — secure the intel game mode
- **Cyberpunk 2020** (1990) — aesthetic
- **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 planned, 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 the 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 their own skills to use. Players can learn which abilities fit well for what situations and adapt to different playstyles. Various strategies on map control can be learned to gain power-ups easier or deny them to enemies. While **Hover Wars** is not primarily geared towards highly competitive play, we think it's important to consider how the game could play at a competitive level when designing and balancing elements of the game.

3.3.5 Getting Hit

Independent of which damage system we implement (hit and continue, lives with hit points, or one hit one kill), we want to ensure that it is in line with number of our design goals.

- **Emphasize the impact of getting hit** — Getting hit should feel bad. It should be immediately noticeable to the player and have a sizeable in-game impact more than just altering points on the scoreboard or decrementing hit points. A temporary loss of vehicle control is enough to get the player's attention (along with any accompanying visual and audio effects), and punishing enough for them to want to avoid getting hit.

This system would be contrasted with damage systems found in some first or third-person shooters, in which players may receive a number of hits with either no or minimal in-game effects, allowing them to “tank” damage.

- **Ease 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 to either getting hit with an ability or not, players simply need to know a single general rule of the consequence of receiving damage.
- **Easy to Interpret** — It is easier to understand how much damage the player can take if hit points are described in small values (ex. 3), or if there is no hit point system at all. As the player's immediate attention is on the action in front of them, they should perfectly understand the state of their hovercraft at a moment's glance, without needing to read large numbers or perform rough calculations.

For example, with a health bar, a percent gauge, or larger hit points values (ex. 100), players may have some rough idea how many hits they can take, but it can still be somewhat fuzzy if abilities deal different amounts of damage, or if players need to figure out how many more hits they can take from how much damage different abilities do.

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 obsolete.
- **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 dodged 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 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

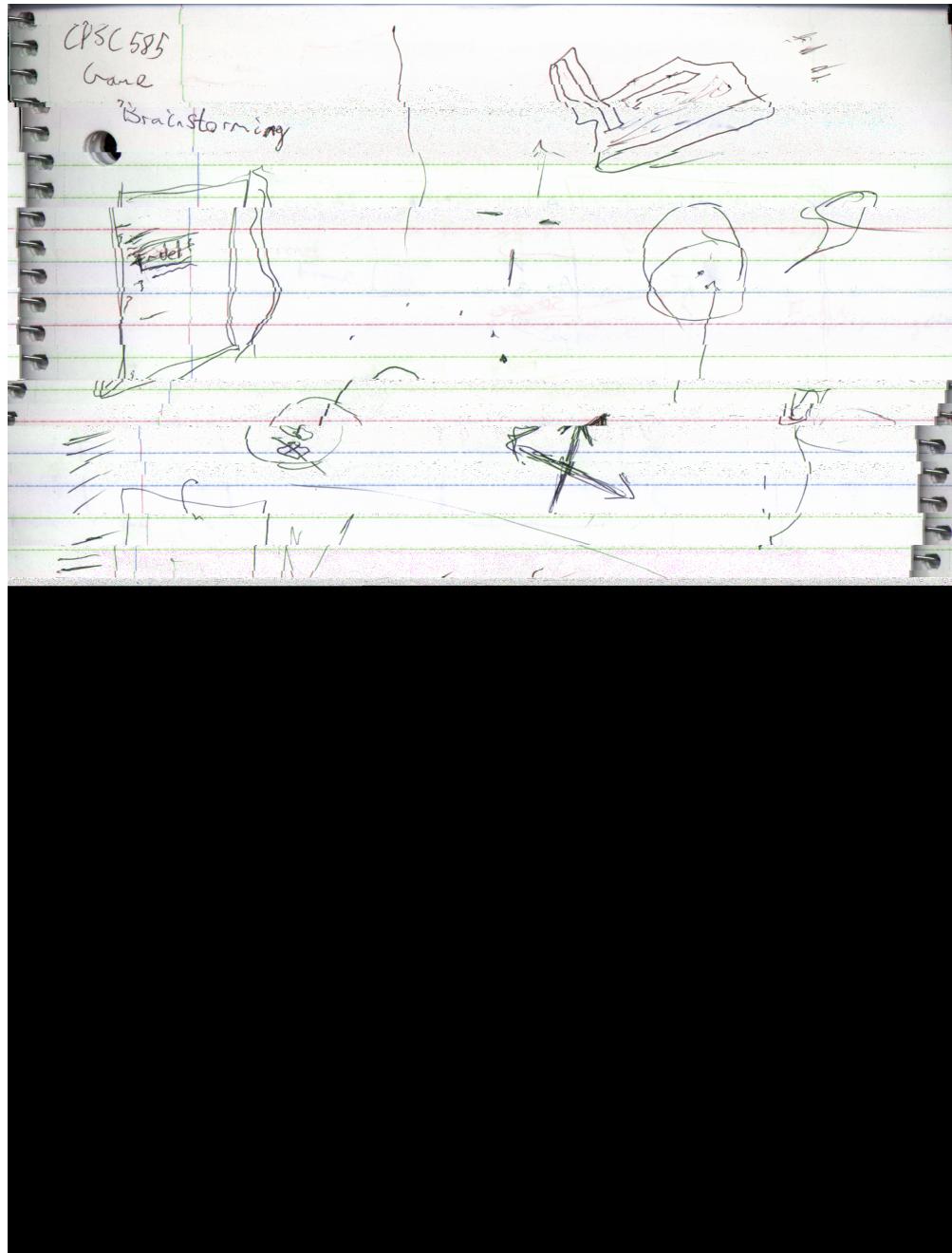


Figure 9: Rough sketches of the map, UI, and hovercraft

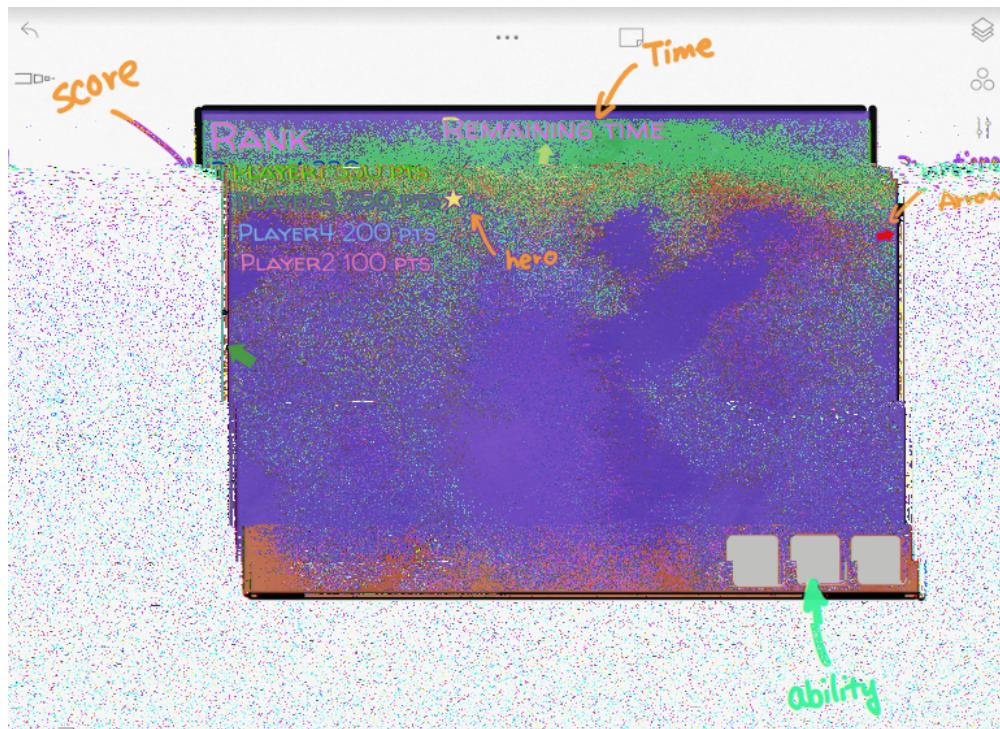


Figure 10: Early UI design. Directional arrows are less intrusive and easier to interpret than a mini-map. Important information such as ability cooldowns, score and time should be clearly visible to the player.

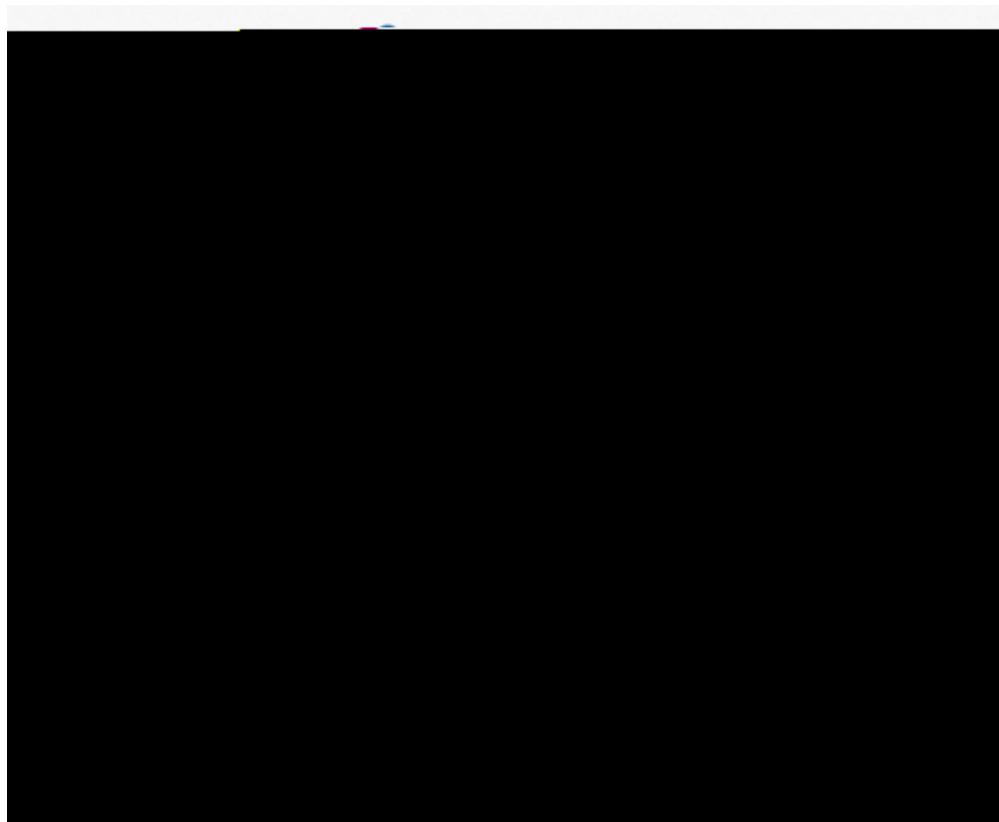


Figure 11: Possible design for the main menu with a Tron-like aesthetic.

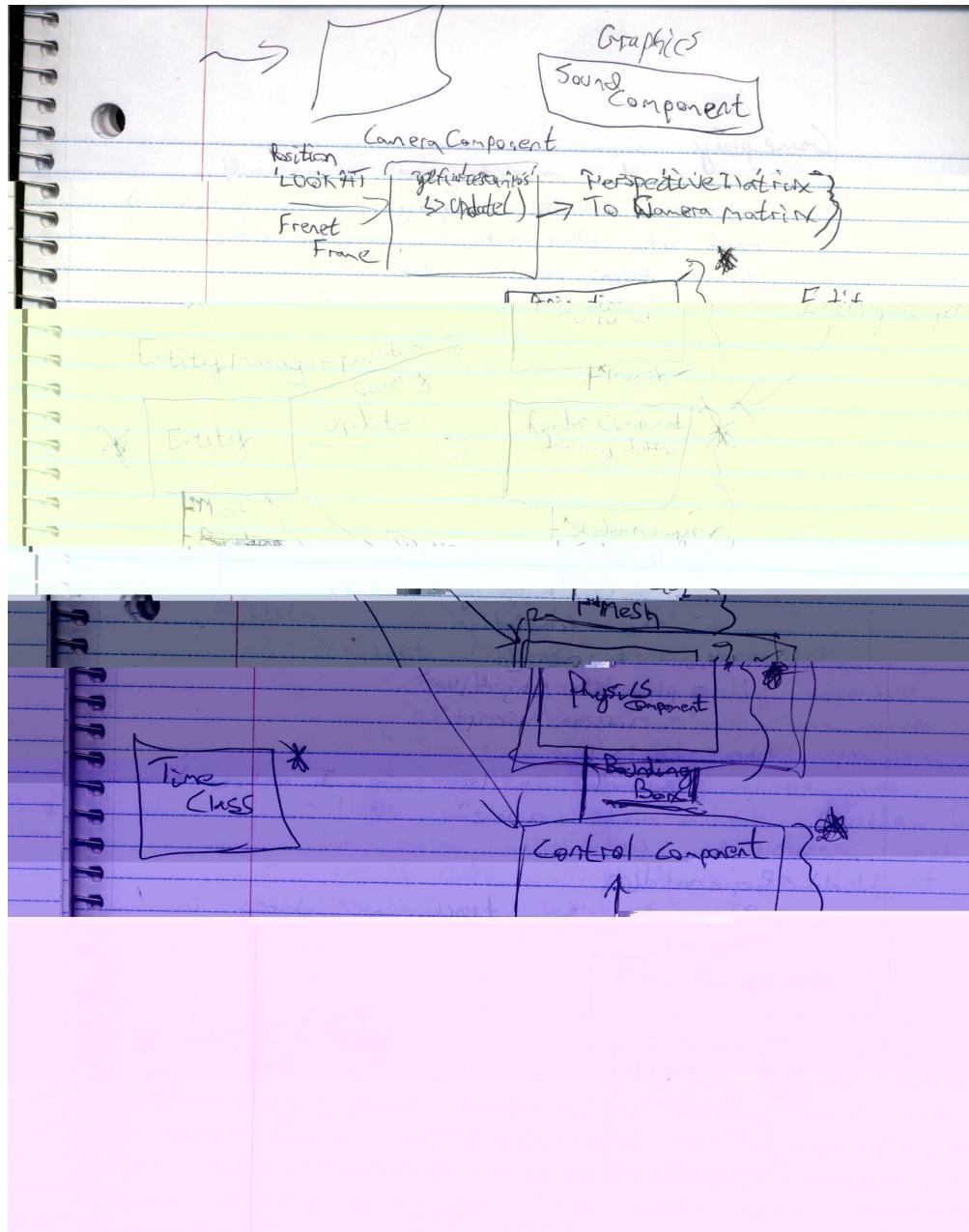


Figure 12: Early design of the game application framework