

# **Planetary Control**

Revised Design Document

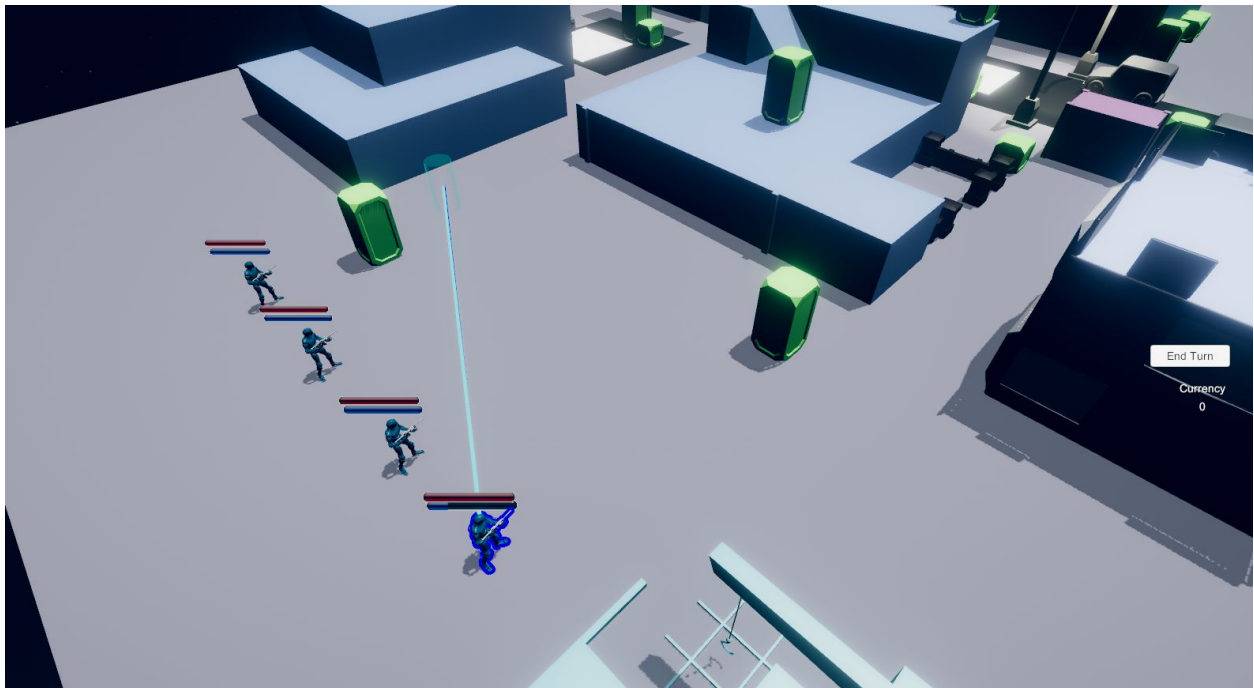
## **Team Saber**

Ishpreet, Alex, Ivan, Brandon, Claire

## Design Questions

- Does the level provide enough variety of moves to feel meaningful
- Are players able to create strategies and execute on them
- Is the level layout readable and is the player able to navigate it
- Does enemy AI behavior meaningfully challenge the player
- Do control points provide a significant enough reward for the player to want to use them
- Is the player able to complete fights in the third person and benefit from their upgrades
- Does the player know what they are supposed to do at all times
- Is the player able to engage in the core loop without issues

## Prototype Functionality



The prototype starts with a short guide as an overt teaching preamble to the gameplay. This sets up the player with the basic knowledge needed to learn the game. Then the gameplay does most of the teaching. The level is taught largely from teaching with experimentation as the sandbox nature of the game lends itself to this technique. Lots of feedback is provided to actions, like units being outlined on hover and paths being drawn to show how the unit will move and if they can move there, that helps the player learn what they can interact with. Then the actual level is structured to offer the player safe places to test mechanics, like a control point that is on their team's side so they

can safely see what it does. From there, the level design layout is structured to provide meaningful locations where players will collide with the enemy, and paths for them to take advantage of. The player wins by defeating all the enemy units on the map.

### **How prototype lets us answer the design questions**

The prototype allows us to answer most of our design questions for the level because the player is reliably able to complete the level. The player can learn the mechanics, engage in the core loop, progress through the level and complete it. The ramping points are testable due to the level offering the player lots of turns to experiment before the enemy AI will get close. There is some feedback missing in the form of UI elements showing each unit's stats (so the player knows when they get upgraded) which we have in a separate prototype but are waiting to integrate.

### **Prototype testing results**

- Does the level provide enough variety of moves to feel meaningful
  - Somewhat true. The level layout creates meaningful choices for the player but the value of height advantage is not stressed enough in the current prototype.
- Are players able to create strategies and execute on them
  - Yes. Every player played the game in a different way. Most just sent out all their units in the first playthrough. In the second, some tried keeping a unit near their base while sending out the other three. Some tried going along the edges of the map rather than the center to flank.
- Is the level layout readable and is the player able to navigate it
  - Somewhat true. The player is able to navigate the level but some confusion occurred in areas that the player thought they could go to but the level did not have the appropriate pathing implemented.
- Does enemy AI behavior meaningfully challenge the player
  - Mostly true. AI targets player units too heavily, which creates challenge but can limit some strategies for the player (like rushing the enemy base).
- Do control points provide a significant enough reward for the player to want to use them
  - Yes, but the player cannot tell if they have them. They do notice their gameplay changes afterward but making it more explicit through UI will help the player's perception of the reward.
- Is the player able to complete fights in the third person and benefit from their upgrades

- Yes, players are benefiting from the upgrades. Some may be too powerful, like the current damage upgrade giving the guns double damage.
- Does the player know what they are supposed to do at all times
  - Not really. Players can get confused about what their actual objective is as it is not explicitly stated anywhere. They eventually realize they have to destroy all the enemies, but some also try to attack the enemy base which currently doesn't work.
- Is the player able to engage in the core loop without issues
  - Mostly true. There are some bugs with interface with unit highlighting that create hiccups, but not major.

### **Unexpected Outcomes**

Some players did not even notice they were in control of a control point, so making that clearer to the player as well as their effects will help cement this as a mechanic. Once the players knew what it did they saw it as a meaningful addition to the strategy element.

### **Further Questions for Next Prototype**

- Which gameplay mode is most engaging, third-person or isometric.
- Is there any exploits that the player can use to win.

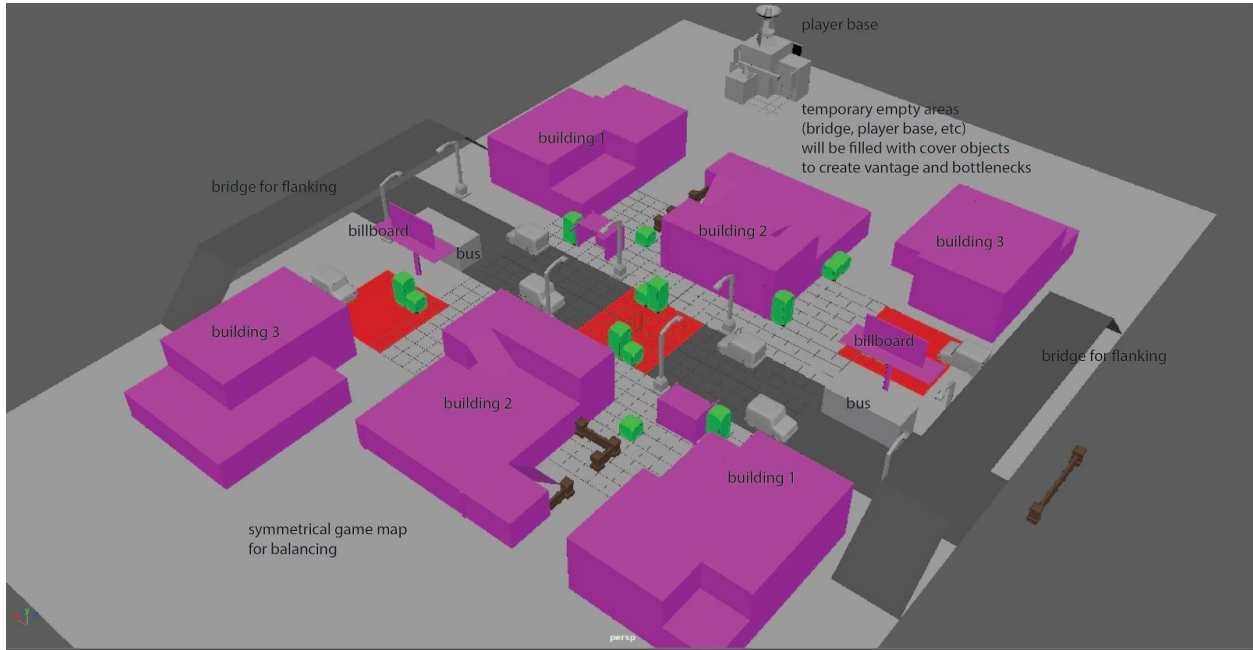
### **Design Refinements from Internal Review**

- Persistent UI unit windows with stats will help the player with strategizing
- Pathing should be consistent so any place that players can jump to should be clearly communicated.
- Enemy AI is a bit too obvious at times, create more variation in its decisions like searching in third-person and rushing the base.

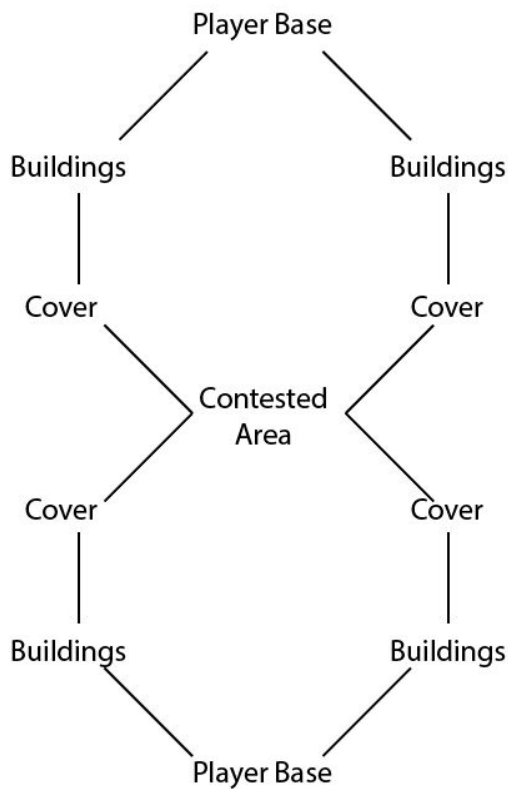
### **Design Refinements based on Playtesting Feedback**

- Always give the player direction, even if the sandbox game. Player needs to know what their next goal is. Show some messages at key points in the game to say what the player can do next.
- Give more reason to use height as a gameplay mechanic. This may fix itself with more units in the game, like a sniper with a really long attack range perching on high buildings.

# Level Design Document



Early rendition of our map but carry fundamental



When the map was designed, we had a few key principles in mind: balance, safety areas, battle areas, flanking areas, and contestable areas. In terms of scale for the map, we designed the size considering the unit sizes. Due to the characters height, jump, and move range, the world follows in scale. Ex: a character is 1.4 units tall, which makes 1 story buildings 2 units tall, and a 2 lane street 6 tiles wide, etc. To facilitate combat, 3 contestable areas are placed along the center of the map. These areas reward the team that controls the point with stat bonuses for their characters. To provide adequate combat environment, we surround combat areas with cover objects that matched the futuristic urban setting, like lamp posts, crates, vehicles, etc. On both sides of control areas,

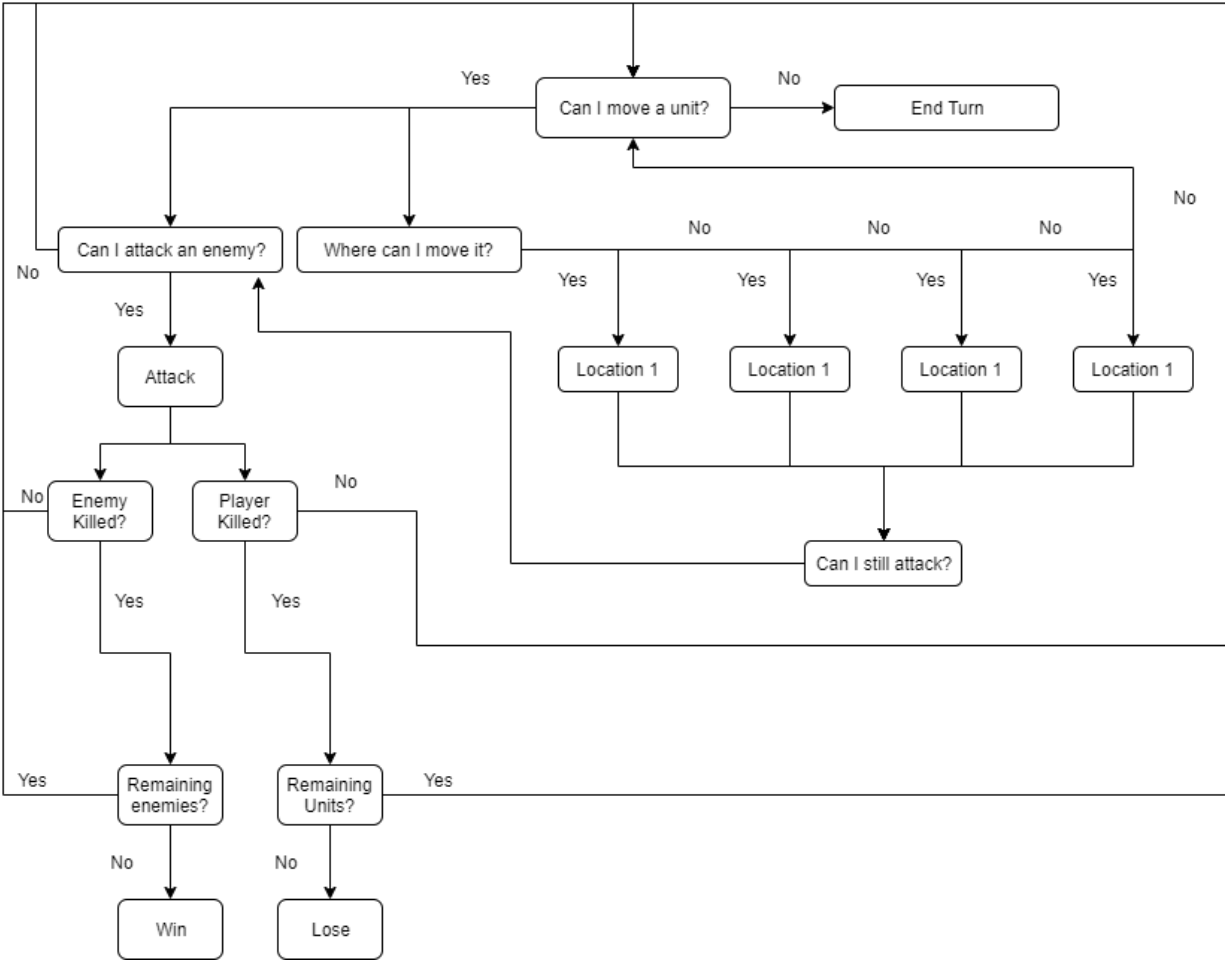
buildings are set to provide vantage points as well as cover for both teams to counterbalance the advantages provided by the covers around the control points. Essentially, buildings give attackers a fighting chance against those defending the control point. To provide counterplay to camping on buildings, we added flanking routes in the form of bridges on the sides of the map. These bridges are 2 stories tall like the buildings, which means that players on rooftops still have vulnerabilities. This may sound like the flanking bridges have no weakness, but bridges are vulnerable to buildings in the same way, with the added weakness that they are far off on the sides of the map therefore limiting their utility. Since player bases are win conditions for each team are player bases, we kept them far apart at the ends of the map so the other team has to gain major map control before being able to target the player base.

Our map has the 4 terrain archetypes as well as a combination of them. Flat terrain is most dominant making up most of the map (street, neighbourhood). Ledge terrain exist in the form of buildings, bridges, and billboards. These areas give player vantage points like ledge terrains do. Gap terrain exist in the form of space between buildings. Cover terrain exist all over the map in the form of objects like crates and walls. A combination of these exist in our map mostly in the form of buildings which are flat, ledge type with cover and gap between other buildings.

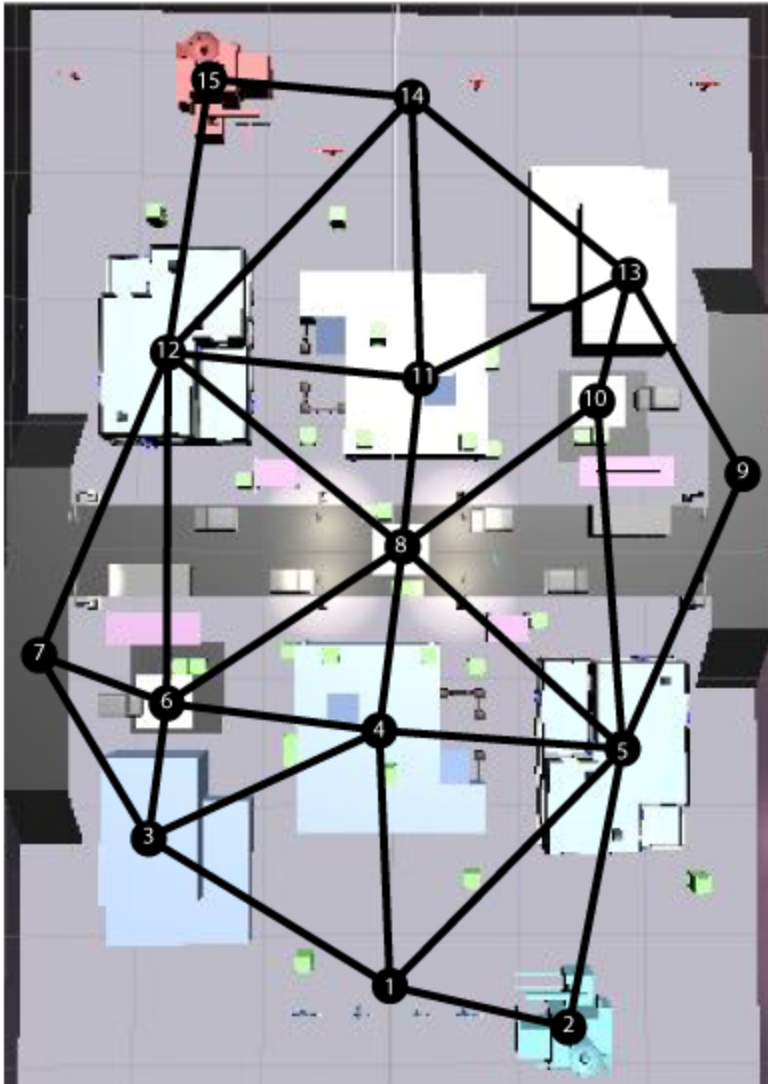
Since our game was initially multiplayer, the characters were designed to be relatively equal in power. There are no categorical swarmers, rangers, heavies, etc, however there are strengths and weaknesses to each character. Snipers have furthest range and high damage, but is weak in health and movement. Paladins have high health and defence but low range and medium attack. Rewards are given to players in the form of currency as they eliminate enemy units. Due to our initial multiplayer setting, the ramping aspect of our game doesn't exist in the form of harder enemies, but in the form of action, ex: no initial combat, but combat happens more often as encounters occur.

### **How we lead the player**

The player's goal at the start is just to move around their units. Once they can attack then they realize they can take damage and defeat enemy units or lose their own. Then their sub-goal becomes to defeat the enemy players units. Achieving these results in the overall goal being completed of winning the game. Losing all their units results in them losing the game.

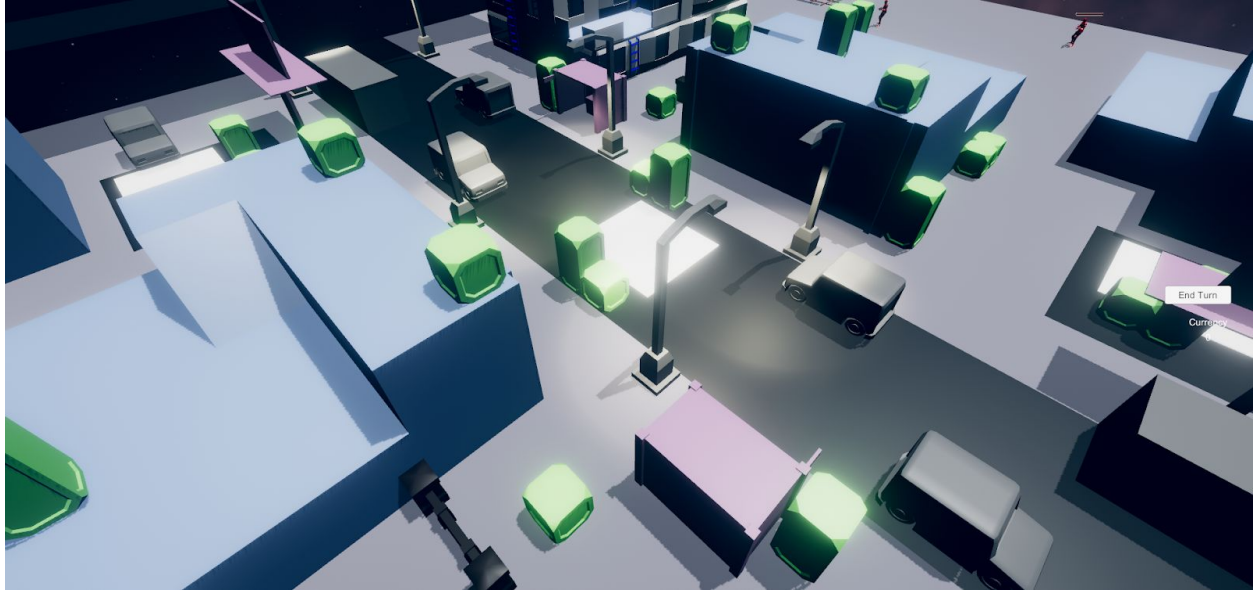


The level layout follows the spider-web layout. This fits our sandbox-style game best because it is up to the player where they want to go next. They are always heading towards the enemy base, but they control how they get there. The route they chose is part of the strategy of the level. Do they want to go straight to a control point, or do they want to prop up on a tall building? They have a lot of choices.



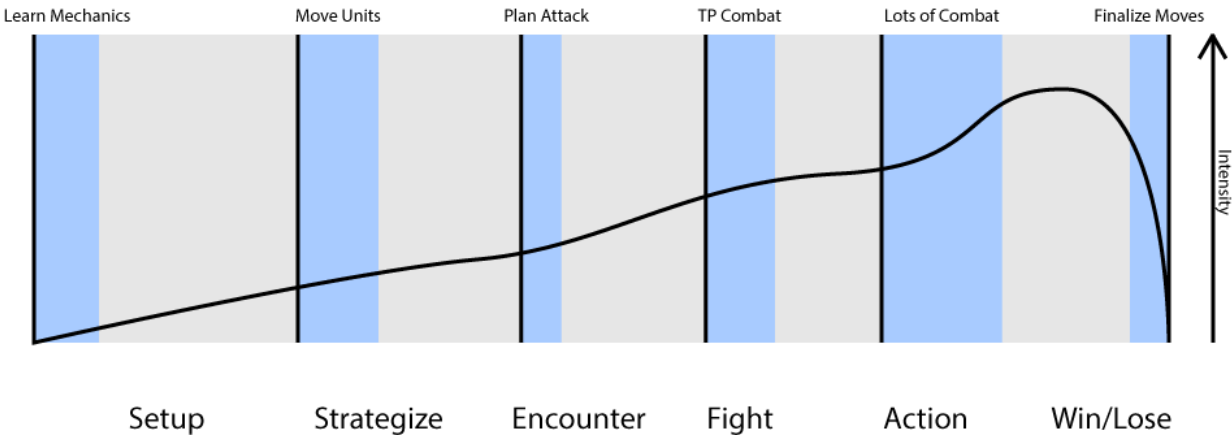
The game uses visual cues in the form of glowing objects. Specifically, we use this to highlight control points to make it obvious that they are something the player will want to target. They also adapt based on their state. If the player is controlling it, it will shift to the player color. If the enemy is controlling it, then it will change to the enemy color. This way the player always knows its state and that it is an important objective without the need for text next to it, pointing out that information.





**Major Beats and Pacing Rhythm**

The major beats of the game go from first setting up your units to eventually defeating the enemy and winning or losing all your units. The general structure is as follows, however, the game being a sandbox-style game the intensity ramping and learning may change. The levels are designed to be repeatable so each time this flow might look a little different. Maybe the player already knows the mechanics and can spend more time strategizing or running straight into fights.



**Reward Schedules**

The level uses three reward schedules. The first is a static per-interval reward of Currency on every turn. Later this will allow them to buy things like upgrades with that

currency. The second is static per-action where capturing certain control points will give a static amount of rewards. The reward types include weapon damage boost, unit health boost, strafe speed boost, run speed boost, jump height boost, and energy boost. Later on, the player will also be able to buy these with the currency they are earning per round. The final is variable per-action. Some control points might not grant anything when used, and their rewards are randomized from the possible reward types.

### **Gameplay teaching components**

The prototype uses an overt teaching preamble to start the gameplay. This sets up the player with the basic knowledge needed to learn the game. Then the gameplay does most of the teaching. The level is taught largely from teaching with experimentation as the sandbox nature of the game lends itself to this technique. Lots of feedback is provided to actions, like units being outlined on hover and paths being drawn to show how the unit will move and if they can move there, that helps the player learn what they can interact with. Walking over glowing control points shows the player capturing it with a change of color, indicating it is a good move to do. Paths are drawn out before players move so they know where they can and cannot go. As the player figures out the map, they will have more strategies and techniques they will have learned that they can use.