## 1a: Pre-Battle Card Selection

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 1a | Pre-battle card selection | Player selects their battle card set before the match | High | Medium | How many cards? Will require lots of UI/UX |

# Brief

Tenant from Vision Canvas: *“Player is challenged in strategy when composing their own set of cards for battle (strategy preparation before battle)”*

Core Pillar: *“Player creates a battle strategy from a set of cards, and can only use those cards in a match”*

The purpose of this feature is to allow the player to create a set of cards to use in their battle against another player. This feature should be simple and easy to understand. The player should not feel frustrated or overwhelmed when in this feature. The player should receive guidance but also be free to make their own decisions.

This feature should allow the player to view information in both as little and as much detail as they would like. For example, if the player already has a deep understanding of the strategic trade-offs of certain units and buildings, then they should be able to select a card without having to review the card’s attributes. However, for players who are interested in number-crunching their strategic advantages, they should be able to view the card’s attributes.

The player’s emotional track during this feature should be neutral. The player should not feel frustrated or angry during this feature. This feature is a building block for the actual battle.

# Spec

Player will have a global card deck that stores all their cards (limited to some number X). From this global deck, the player will select 10 cards to use during battle.

* This global deck is stored by the system or tied to the player’s account
* This global deck is the only cards that the player can select from
* The global deck cannot be reset all at once
  + Available cards can be modified by interacting with the AI shop, see (Feature 10)

Player will have 3 slots for “quick decks”, or decks that the player has already built and is fine with using. This is to help the player get to a match quickly if they want to play with a certain deck.

* A quick deck consists of 10 cards
* These quick decks are stored by the system
* A player can replace a quick deck by building a new one

Requirements

* Loading player’s card deck into UI
* Allowing the player to navigate and select cards
* Allowing the player to view the attributes of the cards
* Allowing the player to remove and change the selected cards
* Allowing the player to save their new deck to one of their 3 “quick deck” slots
* Allowing the player to remove a deck from their “quick deck” slots

# Beat by Beat

1. Player enters match mode
2. Player starts the deck selection process
   1. Player selects a card and views its attributes
      1. Player chooses this card
         1. Player has reached the 10-card limit
            1. Player is notified that they have selected all their cards

Player wants to change a card

Player selects a card to remove

Go back to (3.a.i)

Player wants to keep their deck and go to match

Player leaves deck selection

Player has not reached the 10-card limit

Go back to (3.a.i)

* + 1. Player does not choose this card
       1. Go back to (3.a.i)

Player chooses a “quick deck”

Player reviews that this deck is the one they want to use

Player leaves deck selection

# Asset List

* Standard card model
* Resource icon/drawing
* Resource text description
* Resource attribute values
* Background music
* Card selection sound effects

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 1b | In-battle card selection | Player selects their card in game | High | Medium | How many cards?  +Requires animation system |

# Brief

Tenant from Vision Canvas: “*Player is challenged to make quick decision and react to their enemy's actions in real-time\* (quick reaction during battle)”*

Core Pillar: *“Player creates a battle strategy from a set of cards, and can only use those cards in a match”*

The purpose of this feature is to allow the player to select unit and building cards to play during a battle. This will be the main way that the player wages battle. Once selected and placed, the cards will act automatically. There will be four card slots (and one special slot) that rotate out the player’s 10-card battle deck each time a card is used. This will allow the player to not be overwhelmed by information, and to have a variety of selection for each turn.

Card selection should be easy to understand, fast, and responsive to the user’s actions. The player should feel that card selection is fluid and be able to quickly select a card and place it in their intended lane well within the turn time limit.

The player’s emotional track/flow should be focused on the outcome of selecting a card, not on the act of selecting a card itself.

# Spec

Player will have their battle card deck (around 10 cards), four card slots, and one slot holding the special attack card (if one was selected). The 9-10 cards are randomly placed in the four card slots. The card ordering can be implemented with a queue: every time a card is used, the next item in the queue is placed in the open spot, and the used card is placed at the end of the queue. This is to ensure that the player has a small “cool down” period when using resources.

Once a card is selected, the player will drag it into the lane that they wish the resource to attack in. After the player releases the card, the card will transform into the resource via the animation system (Feature 12).

Requirements

* Loading player’s card deck into the four card slots
* Allowing the player to select a card to play
* Rotating cards into the slots after a card is used
* Allowing the player to select the lane to place the card in

# Beat by Beat

1. Player begins the match
2. Player selects a card
3. Player drags the card to the desired lane
4. Player sees the card transform into the unit/building/special
5. The used card slot is emptied, and a new card is placed in that slot
6. Go back to (2)

# Asset List

* Card slot models
* Card models
  + Icons
  + Attributes
* Card->Resource animations
* Card selection sound effects

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 2 | Winning conditions based on health | Determining which player plays the game | High | Medium | Balancing |

# Brief

Tenant from Vision Canvas: *“Player experiences a thrilling adrenaline rush from the time-limited and fast-paced gameplay”*

Core Pillar: “*Player can win by destroying the enemy base or losing less resources than their enemy in the given time period”*

The purpose of this feature is to describe the rules of the game and how a player can win. The player should feel that the winning conditions are easily understood and visible.

# Spec

The winning conditions are based on whether one player destroys the other player’s base within the given time period, which is calculated based on player health. Destroying the enemy’s base (reducing the enemy’s health to 0) results in an automatic win. If the match timer (Feature 8b) expires when both players are still alive, then the player with the higher overall health will be declared the winner.

Health will be indicated by a UI element somewhere on the screen that is easily visible to the player but not distracting from the battle. The health bar will change color based on value, and show the score in the bar. Health is calculated and updated every second.

|  |  |
| --- | --- |
| **Base Health** | **UI Color** |
| > 50% | Green |
| 21% - 50% | Yellow |
| < 20% | Red |

# Beat by Beat

1. Start the match
2. As players make their moves, calculate health in real-time
3. As the base takes damage, update the health score
4. If either player’s base health reaches 0, end the game

# Asset List

* Health bar drawing (rectangle)
  + Background color
    - Green, yellow, red
  + Health score value
* Score update animations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 3 | Multiple lanes | Player can select which lane to place a unit or building | High | High | Requires animation system and selection system |

# Brief

Tenant from Vision Canvas: “*Player can place units in three lanes to focus on what they should destroy, but the units act independently”*

Core Pillar: “*Player can place units in three lanes to focus on what they should destroy, but the units act independently”*

The three-lane feature is meant to give players a variable game experience. Players are allowed to place units in any of the three lanes, with only one lane leading directly to the base with the other two leading to building plots before the base. By placing troops in side- lanes, players can take a methodical approach to victory by destroying buildings that improve their opponents’ stats (attack, defense, higher energy) before moving onto the base. By attacking the main lane, players can rush their opponents base, taking advantage of times when their opponent is at low energy.

Players will undergo a lot of stress during the game due to the fast-paced nature, putting pressure on them to place troops in certain lanes to counter or take advantage of opponents. Certain decks will be more geared towards different lane strategies, forcing players focus on openings and opportunities to maximize their success.

**Spec**

The lanes of attack dictate the combat paths of resources. There are two types of lanes: unit lanes (3 in number) and building lanes (2 in number). A unit can be placed in any of the three-unit lanes, and it will stay in that lane (except when the building lanes are empty, then that unit will move into the center lane once it hits the end of its current lane). A building can only be placed in the left or right lane (the center lane is occupied by the base).

Units move automatically until they reach the end of the lane or are destroyed. Units can only move in the vertical direction, except in the case mentioned above where the side- building slots are empty. Deciding which lane a unit is placed in is determined by calculating the position of the player’s mouse/finger when released, and “snapping” to the closest lane (done by calculating the distance to the center of the lane). Here are the approximate coordinates of the lane locations.

Lane Coordinate Ranges

|  |  |  |
| --- | --- | --- |
| **Lane Name** | **X Range** | **Y Range** |
| Left lane (lower player) | (-120, -30) | Building to building, (-100, 0) |
| Center lane (lower player) | (-30, 30) | Base to base, (-120, 0) |
| Right lane (lower player) | (30, 120) | Building to building, (-100, 0) |
| Left lane (higher player) | (-120, -30) | Building to building, (0, 100) |
| Center lane (higher player) | (-30, 30) | Base to base, 0, 120) |
| Right lane (higher player) | (30, 120) | Building to building, (0, 100) |

In the case that the user releases at the exact center between two lanes, the card will snap back to its slot and the user will drag it to the desired position again.

Requirements

* Able to pick lanes to place units or buildings
* Seeing units move and fight within lanes
* Units move from side lanes toward base after destroying (or in the absence of) a building

**Beat by Beat**

1. Players enter the game after selecting their decks
2. Players decide what buildings they want to put in their side-lanes
   1. Player selects a building
   2. Player drags card to the desired side lane
3. As the player selects a card to place
   1. Players drags the card to one of the lanes
   2. Card transforms into the resource from the card (Feature 12)
   3. Go back to (3)

**Asset List**

* Unit and building models
* Unit and building animations
* Map appearance/texture

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 4a | Combat system: units | Determining unit strengths and weaknesses versus enemy units | High | High | What are the attributes?  Requires animation system |

# Brief

Tenant from Vision Canvas: *“Player is challenged to make quick decisions and react to their enemy’s actions in real-time\* (quick reaction during battle)”*

Core Pillar: *“Each unit and special (the cards) has strengths and weaknesses (eg. strength, move, health, etc.)”*

The purpose of the unit combat system is to develop a system of strengths and weaknesses that the player uses to make in-game decisions. Ultimately, all three combat systems specify the rules of the game and help in determining the winner and loser.

The unit combat system should be deterministic (outcomes are predictable), easily available and understandable (not hidden behind a “black box” algorithm), but relatively complex as to not be boring. The player should want to discover and understand the relative strengths and weaknesses and having an understanding should provide some reward (i.e, the player performs better in a battle when following the combat system).

The combat system for units should give the player a feeling of control, a sense of thrill in battle, and contribute to the overall emotions caused by winning or losing.

# Spec

The unit combat system will encode the basic rules of the game and will be responsible for calculating winners/losers.

There are four-unit classes, with different attribute values, strengths, and weaknesses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Unit class** | **Starting Health (points)** | **Damage per second (points)** | **Energy cost (points)** | **Movement (points)** |
| Infantry | 100 | 5 | 5 | 10 |
| Ranged | 90 | 10 | 8 | 15 |
| Vehicle | 120 | 20 | 10 | 20 |
| Special Ops | 150 | 40 | 20 | 30 |

Each second, damage and health calculations will need to be performed for all units engaged in combat. Two units are said to be in combat if they are within a certain distance threshold of one another. The distance threshold is based on the size of the game map and the unit’s relative positions. For unit-unit combat, one only needs to consider the distance in the vertical (y) direction, since units are locked into the lane they were placed in.

|  |  |
| --- | --- |
| **Unit class** | **Distance threshold** |
| Infantry | <= 5 distance units |
| Ranged | <= 20 distance units |
| Vehicle | <= 10 distance units |
| Special Ops | <= 5 distance units |

Requirements

* Storage of unit attributes (health, damage per second, etc.)
* Real-time damage calculations between units
* Real-time health calculations between units
* Real-time movement calculations between units and the lanes they occupy

# Beat by Beat

*Player A plays an Infantry class unit, Player B plays an Infantry class unit*

1. Without any modifiers and with equal health, both units are evenly matched and will end in mutual destruction
2. With unequal health, the unit with the higher starting health will win

*Player A plays an Infantry class unit, Player B plays a Ranged class unit*

1. Without any modifier, the Ranged unit has an advantage
2. If the Ranged unit’s health is lower than the Infantry’s damage per second, the Ranged unit will be destroyed
3. If the Infantry unit’s health is lower than the Ranged unit’s damage per second, the Infantry unit will be destroyed

*Player A plays an Infantry class unit, Player B plays a Vehicle class unit*

1. Without any modifiers, the Vehicle unit has an advantage
2. If the Vehicle unit’s health is lower than the Infantry’s damage per second, the Vehicle unit will be destroyed
3. If the Infantry unit’s health is lower than the Vehicle’s damage per second, the Infantry unit will be destroyed

*Player A plays an Infantry class unit, Player B plays a Special Ops unit*

1. Without any modifiers, the Special Ops unit has the advantage
2. If the Special Ops unit’s health is lower than the Infantry’s damage per second, the Special Ops unit will be destroyed
3. If the Infantry unit’s health is lower than the Special Ops unit’s damage per second, the Infantry unit will be destroyed

*Player A plays a Ranged class unit, Player B plays a Ranged class unit*

1. Without any modifiers and with equal health, both units are evenly matched and will end in mutual destruction
2. With unequal health, the unit with the higher starting health will win

*Player A plays a Ranged class unit, Player B plays a Vehicle class unit*

1. Without any modifiers, the Vehicle unit will have an advantage
2. If the Ranged unit’s health is lower than the Vehicle’s damage per second, the Ranged unit will be destroyed
3. If the Vehicle unit’s health is lower than the Ranged unit’s damage per second, the Vehicle unit will be destroyed

*Player A plays a Ranged class unit, Player B plays a Special Ops unit*

1. Without any modifiers, the Special Ops unit have the advantage
2. If the Ranged unit’s health is lower than the Special Ops unit’s damage per second, the Ranged unit will be destroyed
3. If the Special Ops unit’s health is lower than the Ranged unit’s damage per second, the Special Ops unit will be destroyed

*Player A plays a Vehicle class unit, Player B plays a Vehicle class unit*

1. Without any modifiers and equal health, both units are evenly matched and will end in mutual destruction
2. With unequal health, the unit with the higher starting health will win

*Player A plays a Vehicle class unit, Player B plays a Special Ops unit*

1. Without any modifiers, the Special Ops unit will always have an advantage over the Vehicle unit
2. If the Vehicle unit’s health is lower than the Special Ops unit’s damage per second, the Vehicle unit will be destroyed
3. If the Special Ops unit’s health is lower than the Vehicle unit’s damage per second, the Special Ops unit will be destroyed

*Player A plays a Special Ops unit, Player B plays a Special Ops unit*

1. Without any modifiers and with equal health, both units are evenly matched and will end in mutual destruction

# Asset List

* Unit character model
* Unit character animation
* Unit character audio
* Combat sound effects
* Unit card models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 4b | Combat system: buildings | Determining building strengths and weaknesses against enemy units and buildings | High | High | Modifier values, requires animation system |

# Brief

Tenant from Vision Canvas: *“Player is able to ‘choose their own adventure’ by customizing a variety of aspects in gameplay”*

Core Pillar: *“Each unit and special (the cards) has strengths and weaknesses (eg. strength, move, health, etc.)”*

By giving players a variety of buildings to choose from, players can select certain buildings that amplify the strength or cover the weaknesses of their decks.

As with units, the building combat system should be deterministic (outcomes are predictable), easily available and understandable (not hidden behind a “black box” algorithm), but relatively complex as to not be boring. The player should want to discover and understand the relative strengths and weaknesses, and having an understanding should provide some reward (i.e, the player performs better in a battle when following the combat system).

The combat system for buildings should give the player a feeling of control, a sense of thrill in battle, and contribute to the overall emotions caused by winning or losing.

# Spec

The building combat system will encode the basic rules of the game.

Requirements

* Storage of building attributes (health, damage per second, etc.)
* Real-time damage calculations between the building and enemy units
* Real-time health calculations between the building and enemy buildings

There are several different classes of buildings, each with different attribute values, modifiers, and energy costs.

|  |  |  |  |
| --- | --- | --- | --- |
| **Building class** | **Health (points)** | **Modifier (points)** | **Energy cost (points)** |
| Hospital | 300 | +1 health/second for player’s units on the battlefield (no effect if at max health) | 30 |
| Fort/Barracks | 400 | +10 damage/second for player’s units on the battlefield | 30 |
| Energy Generator | 300 | +1 energy/second generated by the building | 30 |

# Beat by Beat

*Building setup*

1. Player selects a building card
2. Player selects the lane to place the building

*Building-unit interaction*

1. A unit will attack a building until either the unit is destroyed, or the building is destroyed
   1. The unit will be alive until its health is lower than the building’s damage per second
   2. The building will be alive until its health is lower than the unit’s damage per second

# Asset List

* Building model
* Building animation
* Building sound effects
* Building card models

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 4c | Combat system: special attack | Special attack that has some global effect | Medium | High | Requires interaction with unit and building combat systems (modifier) |

# Brief

Tenant from Vision Canvas: *“Player is able to ‘choose their own adventure’ by customizing a variety of aspects in gameplay”*

Core Pillar: *“Each unit and special (the cards) has strengths and weaknesses (eg. strength, move, health, etc.)”*

The purpose of the special attack combat system is to develop a system of strengths and weaknesses that the player uses to make in-game decisions. Ultimately, all three combat systems specify the rules of the game and help in determining the winner and loser.

The special attack combat system should be deterministic (outcomes are predictable), easily available and understandable (not hidden behind a “black box” algorithm), but relatively complex as to not be boring. The player should want to discover and understand the relative strengths and weaknesses, and having an understanding should provide some reward (i.e, the player performs better in a battle when following the combat system).

The combat system for special attacks should give the player a feeling of control, a sense of thrill in battle, and contribute to the overall emotions caused by winning or losing. Specials will also add some variety to battles and temporarily update the combat system for units and buildings by boosting or reducing certain attributes.

# Spec

The special attack combat system will encode the basic rules of the game.

Requirements

* Real-time attribute calculations and combat system updates of all units on the battlefield
* Real-time attribute calculations and combat system updates of all buildings on the battlefield
* Effects are applied to multiple turns

There are several different classes of special attack cards. Each has different effects that last for some specified number of turns.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Special attack class** | **Modifier** | **Energy cost (points)** | **Color** | **Cooldown** |
| Morale | Unit damage per second is increased by 10 points for 3 turns | 50 | Deep red | 90 sec |
| Reinforcement s | Unit energy costs are cut in half for 10 turns | 50 | Navy blue | 90 sec |
| Reconnaissan ce | Unit distance threshold is decreased (units do not need to be as close to enemy units before they start doing damage) for 5 turns | 50 | Grey | 90 sec |
| Energy boost | Building energy generation is increased by 3 energy/second for 5 turns | 50 | Yellow | 90 sec |
| Cyberwarfare | Enemy energy generation is decreased by 5 energy/second for 5 turns | 50 | Dark green | 180 sec |
| Fireball | Cast a fireball from base, dealing 50 damage in a small area at target location | 20 | orange | 30 sec |
| Ultimate | Kill all the enemy target in a target area | 100 | black | 180 sec |
| Pause | Time-stop an enemy building for 10 sec | 50 | Gold | 90 sec |

# Beat by Beat

1. Player A selects a special attack card
   1. The attributes of that card are applied to the battlefield
2. Player A must wait some amount of time before they are able to use the special attack card again

# Asset List

* Special card model
* Special attack visual effects
* Special attack sound effects

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 5 | Energy generation | Energy is generated by hurting the enemy and passively every second | High | Medium | What rate to generate? How much to start with? |

# Brief

Tenant from Vision Canvas: *“Player is challenged to make quick decisions and react to their enemy’s actions in real-time\* (quick reaction during battle)”*

Core Pillars: “*Player manages their resources through energy, which is generated by hurting enemy units and buildings (also passively generated every second)”*

The purpose of this feature is to provide the player with a limitation within battles. Energy is used to use cards and is earned by destroying enemies and is passively generated during each turn.

# Spec

The energy generating system will encode the basic rules of the game, and will be responsible for calculating winners/losers, energy is generated by units in the battleground:

|  |  |  |
| --- | --- | --- |
| **Unit class** | **Max Energy Pool** | **Energy supplying** |
| Base | 100 | 3 energy per second |
| Energy generating building | 30 | 1 energy per second |
| Destroy each enemy unit | - | 25% of the enemy unit’s cost |
| Destroy each enemy buildings (not base) | - | 20 energy per building |

When the game begins, there should be an energy bar beside the HP bar on the UI system. The basic max energy should be 100, and it comes from the base. Besides that, if the player chooses to build extra energy generating buildings, the max energy pool is increased by each energy generating building.

The game start at max energy so the player can choose whichever card they want to start the match.

The base and extra generating building unit also generate energy per second during the match automatically. And the player can grab more energy through destroying enemy units. Some energy from destroying enemy units but plenty energy from destroying enemy building. Usually, enemy gained from destroying unit should be 25% of the destroyed enemy unit’s cost, and the energy grab from destroying building should be a constant number.

When the player’s energy generating building is destroyed by the enemy player, player won’t lose the extra energy pool they gain from the building but lose the energy generated per second by the building.

Requirements

* Storage of unit attributes (health, damage per second, etc.)
* Real-time damage calculations between units
* Real-time health calculations between units

# Beat by Beat

1. Player enter the game at full energy
   1. Full energy would be 100 if player don’t have any extra energy building
   2. Full energy would be 130/160 depends on how many energy buildings player has
2. Match Begin and Energy start to generate
   1. Energy generate whether player consume it or not
   2. Energy generating rate depends on buildings
      1. Base generate 3 points of energy per second
      2. Base with one energy building generate 4 points of energy per second
      3. Base with two energy buildings generate 5 points of energy per second
3. Player gain extra energy when they kill enemy units
   1. Player gain energy through killing enemy units
      1. Player gain 25% energy of the enemy units’ cost
      2. Killing special cards won’t generate energy
   2. Player gain energy through destroying enemy buildings
      1. Player gain 20 energy each building they destroyed
      2. 20 energy is a constant and won’t change no matter what building the player destroyed
      3. Base is not included since the destruction of base will directly end the match.

# Asset List

1. The energy visual bar
2. The energy generating animation
3. Visual effect of gaining energy through killing enemy units
4. Visual effect of gaining energy when destroyed enemy building

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 6 | Faction selection | Player selects a faction that determines card names and animations | Medium | Medium | Will need to update animations and card names to match the selected faction |

# Brief

Tenant from Vision Canvas: “*Player can customize their experience - whether it's playing as a different type, on a different map, picking a new battle set, or reacting to the enemy's attacks differently”*

Core Pillars: “*Player can customize their experience - whether it's playing as a different type, on a different map, picking a new battle set, or reacting to the enemy's attacks differently”*

The purpose of this feature is to give players a more variable experience. Ultimately, each faction’s cards are just variations of the same cards from the other two factions, yet the difference in animations and appearances creates a concrete conflict for players to get behind. Players should choose the faction that interests them the most at the time. Because this is a simple choice before the game, this decision should not be a defining characteristic of the player the way that Alliance vs Horde is in WoW.

**Spec**

Player will be prompted to choose what team they want to play as before the game starts and are then put in a matchmaking waiting screen.

Requirements

* Generate a screen where the three factions are pictured and selectable
* Allow player to choose which faction they want
* Reskin the player’s deck to fit that faction’s card cosmetic variations

**Beat by Beat**

1. After selecting their wager, players are taken to faction selection screen
2. Player selects faction
3. Players wait to be paired with an opponent

**Asset List**

* Faction UI icons
* Unit card skins
* Building card skins
* Special attack card skins

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 7 | Wager system | Player selects a wager to bet on the match | High | High | What to do if no one matches wager  Are there limitations? |

# Brief

Tenant from Vision Canvas: “*Players are incentivized to wager currency in order to earn more points to move up in rank, and also earn more currency to buy new cards*”

Core Pillars: “*Players are incentivized to wager currency in order to earn more points to move up in rank, and also earn more currency to buy new cards”*

*“Wager system drives key elements of the game: matchmaking and wagering the outcome of the battle”*

The purpose of this feature is to give more gravity to winning and losing while allowing players to customize their own experience. When players win, they get a return on their wager, making that win more satisfying than if they were to not wager (vice versa if they lose). By having a customizable wager amount, we also allow players to decide how seriously they want to play their opponents. For example, the best players who want an extremely competitive game would wager a lot of money, matching their confidence in their own abilities. Players who want to try out a new deck or are still learning the game would gravitate to the no-low wager matches. This system will be outlined to the players before their first games, giving them an idea of how to use it.

This feature will have the option to choose between 5 wager amounts (None, Low, Medium, High, and High Roller) which are set to a certain in-game currency value. The player should think about just what kind of game they want to play before going in since the competition between two people wagering nothing will be very different than people playing in high roller.

**Spec**

Player will be prompted with a wager screen before the game starts with the five choices of wager amounts. Players can then choose which amount they want. Players are then matched with someone who chose the same wager amount as them.

Behind the scenes, the players are organized in a structure and wait until another player with the same wager is also waiting to be matched.

Requirements

* Loading Wager Screen
* Allow player to choose which value they want to choose
* Match players with other players from that

**Beat by Beat**

1. Player is prompted with the wager screen after deck selection
2. Player chooses which amount they want to wager
3. Players are then matched with someone of relative skill and same wager.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 8a | Turn timer | Player has 3 seconds to make a move | High | Medium | What happens if player does nothing? Is there a default action? |

# Brief:

Tenant from Vision Canvas:” *Player is challenged to make quick decision and react to their enemy's actions in real-time\* (quick reaction during battle)”*

Core Pillar: “*Player is limited by time during turns and also by the overall match length*”

The turn timer’s parameters are set to three seconds and the opportunity to play one card. If players do not play a card in that set time, they defer their turn to their opponent. The turn timer exists to keep the game moving. If there was no turn timer, the player who has an advantage could just sit back and let his troops defeat his opponent with them having no opportunity to counter. In the opposite sense, it also keeps players from playing multiple cards in one round to give them an advantage. The game is supposed to be very fast paced and having the timer set to three seconds keeps gameplay quick and stressful, improving a player’s focus and ability to think swiftly.

The timer will begin after the short building period at the start of the game and will continue until the game is over. The player will be locked out of making another move until after the timer has expired.

# Spec

The game will need to have a single time turner for both players. This will be kept by the system. If both players make a move before the timer expires, the timer is automatically reset and restarts, so not to penalize quick-thinking. The timer checks if both players have made a move, and then resets. This could be implemented using event/callback functions instead of looping and checking.

# Beat by Beat

1. Players A and B enter a match
   1. The timer begins
   2. Players A and B each make a move within the given time
   3. The timer automatically expires and restarts
2. Player A makes a move, but B does not
   1. The timer expires
   2. Player A’s move updates the game, Player B has no game update
   3. The timer restarts

# Asset List

* UI timer in corner of screen
* Visual feedback for user when timer is running out

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 8b | Match timer | Battle is limited to 3 minutes | High | Low | Is time given enough/too long? |

# Brief

Tenant from Vision Canvas: *“Player experiences a thrilling adrenaline rush from the time-limited and fast-paced gameplay.”*

Core Pillar: “*Player is limited by time during turns and also by the overall match length.”*

The purpose of this feature is to make battle fast-paced and prevent players from wasting time or wasting a match. Currently, the match timer will give both players 3 minutes to do as much damage to their opponents as possible. If one player manages to destroy their enemy’s base before the match timer expires, the game automatically ends and that player is declared the winner. The match timer helps keeps players focused and allows players to play in “bursts” if they don’t have a lot of time to commit to the game.

# Spec

The game will need to have a single match turner for both players. This will be kept by the system. The timer checks if a player has won, which could be implemented using event/callback functions instead of looping and checking.

# Beat by Beat

1. Players A and B enter the battle
2. There is a short countdown, and the match timer begins
3. Players A and B make moves until the timer expires
   1. The timer expires
   2. The player with more health wins
4. Player A destroys Player B’s base
   1. The timer automatically expires
   2. Player A is declared the winner

# Asset List

* UI timer in corner of screen (near or opposite the turn timer)
* Visual feedback when there are 30 and 10 seconds left

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 10a | AI Shop - Buy | Player can visit the AI shop and buy cards | High | Medium | How to store data? When to update with new cards? |

# Brief

Tenant from Vision Canvas: *“Player is able to “choose their own adventure” by customizing a variety of aspects in gameplay.”*

Core Pillar: *“Players can buy cards from their game wager earnings, and sell unwanted cards to an AI shop”*

The purpose of this feature is to allow the player to customize their global card deck. The goal of the AI Shop is to keep the player engaged in the game by releasing or unlocking new cards periodically that the player can purchase. Since the player is limited to a certain number of cards in their global deck, there must also be an option for the user to get rid of unwanted cards by selling them to the AI shop for in-game currency. The AI shop will be styled to fit a military science-fiction theme. The player should feel excited to discover new cards to purchase.

# Spec

The ability to withdraw currency from the player’s account in exchange for the card they are buying

# Beat by Beat

1. Player enters the AI shop
2. Player browses the selection of cards available for purchase
   1. Player has a full deck - they cannot purchase the card without first selling a card to free up a slot
3. Player selects a card
   1. Player views the attributes of the card
   2. Player purchases the card
   3. Player adds the card to their global deck
4. Player can leave the AI shop or return to (2)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 10b | AI Shop - Sell | Player can visit the AI shop and sell cards | High | Medium | Should you keep the cards in storage to buy again? |

# Brief

Tenant from Vision Canvas: *“Player is able to “choose their own adventure” by customizing a variety of aspects in gameplay.”*

Core Pillar: *“Players can buy cards from their game wager earnings, and sell unwanted cards to an AI shop”*

The sell feature exists for the sole reason of repurposing. Throughout their experience in collecting cards through levels, players are bound to collect cards that do not fit their current playstyle. The sell feature allows players to take these unwanted cards and turn them into currency that players can then use to buy cards they find more useful from the shop. Player should be relieved to not be stuck with bad cards and excited to buy the cards they want afterwards.

# Spec

The ability to assign currency to the player’s account in exchange for the selected card from their inventory.

# Beat by Beat:

1. Player enters AI shop
2. Player selects the sell option
3. Player is presented with the prices for each card they own
4. Player selects which card(s) they want to sell
5. Player confirms their sale and receives corresponding currency

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 11 | Deck storage | Player has a global deck that they pull cards from to form their battle sets | Medium | Low | How to tie deck to player? Via account or just stored to local data? |

# Brief

We want the players have a place to view all of their cards to inspire a feeling of collections. Also, it is the place to compare all the cards and do the thinking to come up with different kinds of card sets, which may fit different types of battles based on players’ decisions.

# Spec

On the storage deck, the player can view all their collections and make upgrade decisions. The storage is unlimited, players can store as many kinds of cards as they want. On the interface, there are 5 card set options. Each card set has 10 slots. Players can pull the cards from the card pool to fill the slots. When the card set is selected, it is defaulted as the battle card set.

Cards can be sorted by arena, rarity or level.

# Beat by Beat

1. On the storage deck, players select a card in the card pool
2. Drag it to the slot to fill it in
3. Or drag it to the slot that already occupied with a card to do the replacement
4. Tap the set options on the top to switch from different card sets.
5. In the card pool, tap a card to view detailed information and do the upgrade action.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 12a | Animations  - Card selection | Selecting a card transforms it into the unit/building in the lane selected | High | High | Speed of animation, level of detail, is the animation helpful visual feedback or a distraction? |

# Brief

Tenant from Design Canvas: *“Player experiences a thrilling adrenaline rush from the time- limited and fast-paced gameplay”*

The purpose of this feature is to provide feedback and interaction for the user to select cards during battle. This feature will also enhance the style and cosmetic appearance of the game by providing nice visuals that fit into the universe and the player’s expectations.

The animations should be useful, and fluid. The player should not find the animations clunky or unrealistic. Since this game is fast-paced, it is also important that the player not find the animations distracting from making decisions.

Player should feel delighted when focusing on the animations. Player should notice animations when playing the game for the first time, but once the player is an expert, they should not feel bored, slowed, or inhibited by the animation system.

# Spec

In order to have a dynamic and full card selection animation system, the following assets will need to be designed

* Units
  + Icons for all classes and factions
  + Descriptions for all classes and factions
  + Attributes for all classes and factions
* Buildings
  + Icons for all classes and factions
  + Descriptions for all classes and factions
  + Attributes for all classes and factions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 12b | Animations  - Unit movement and combat | Units placed in lanes move towards their target until they are destroyed | High | Medium | Speed of animation, level of detail, is the animation helpful visual feedback or a distraction? |

# Brief

Tenant from Design Canvas: *“Player experiences a thrilling adrenaline rush from the time- limited and fast-paced gameplay”*

The purpose of this feature is to provide feedback for the user during unit combat. This feature will also enhance the style and cosmetic appearance of the game by providing nice visuals that fit into the universe and the player’s expectations. The animations should allow the player to assess their current level of battle success and provide the necessary feedback to adjust their strategy.

The animations should be useful, and fluid. The player should not find the animations clunky or unrealistic. Since this game is fast-paced, it is also important that the player not find the animations distracting from making decisions.

Player should feel delighted when focusing on the animations. Player should notice animations when playing the game for the first time, but once the player is an expert, they should not feel bored, slowed, or inhibited by the animation system.

# Spec

In order to have a dynamic and full unit combat animation system, the following assets will need to be designed

* Character models for all classes and factions
* Unit Animation sequences (create all of these for each unit class)
  + Spawn/creation
  + Walking and “jump over”
  + Combat
  + Death
  + Victory
* Character sound effects for all classes and factions

Card to Unit Sample Animations

* When the player releases the card, the card spins twice, transforms into a white sphere, and appears to “create” the unit out of thin air, surrounded by smoke
* The unit displays its “ready” position, and begins walking toward the enemy
* When the unit encounters an enemy unit, it starts the combat animation sequence
* When a unit is destroyed, the death animation sequence is shown
* When a unit successfully destroys its enemy unit, a small celebratory animation sequence is shown

# Beat by Beat

* Player selects and drags a card to the desired lane
* Player sees the card to unit animation
* The unit begins to move in the direction of its lane and begins attacking the enemy units or building
* If there is no building and the unit reaches the end of the lane, the unit “jumps over” to the center lane to begin attacking the enemy base or enemy units

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 12c | Animations  - Buildings and combat | Buildings with modifiers show animations and their effects | High | Medium | Speed of animation, level of detail, is the animation helpful visual feedback or a distraction? |

# Brief

Tenant from Design Canvas: *“Player experiences a thrilling adrenaline rush from the time- limited and fast-paced gameplay”*

The purpose of this feature is to provide feedback for the user during building combat. This feature will also enhance the style and cosmetic appearance of the game by providing nice visuals that fit into the universe and the player’s expectations. The animations should allow the player to assess their current level of battle success and provide the necessary feedback to adjust their strategy.

The animations should be useful, and fluid. The player should not find the animations clunky or unrealistic. Since this game is fast-paced, it is also important that the player not find the animations distracting from making decisions.

Player should feel delighted when focusing on the animations. Player should notice animations when playing the game for the first time, but once the player is an expert, they should not feel bored, slowed, or inhibited by the animation system.

# Spec

In order to have a dynamic and full building combat animation system, the following assets will need to be designed

* Models for all factions
* Background animation based on type
  + Energy generation, barracks, fort, etc.
* Stock animations for combat between each unit class
  + Infantry, ranged, vehicle, special ops
* Stock animations based on event
  + Creation, destruction
* Sound effects

Building Sample Animations

* When the player releases the card, the card spins twice, transforms into a white sphere, and shows a fast time-lapse of the building being built (think Civ IV wonder screen)
* The building has a general action animation based on its class
  + Forts show cannons firing
  + Barracks show combat training
  + Hospitals show doctors/labs
  + Generators show flashing lights
* When the building encounters an enemy unit, the unit starts the combat animation sequence, but the building does not change its animation
* When a building is destroyed, the death animation sequence is shown

# Beat by Beat

1. (Before battle) Player selects and drags a building card to the desired lane
2. (In battle) Player sees the card to building animation
3. The building begins its actions (combat for barracks, defense for fort, healing for hospital, energy for generators)

When the building is destroyed, the animation sequence shows the destruction and removes it from the map, as well as its effects

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 12d | Animations  - special attack and combat | Special attack shows animations and effects | High | Medium | Speed of animation, level of detail, is the animation helpful visual feedback or a distraction? |

# Brief

Tenant from Design Canvas: *“Player experiences a thrilling adrenaline rush from the time- limited and fast-paced gameplay”*

The purpose of this feature is to provide feedback for the user during special attacks. This feature will also enhance the style and cosmetic appearance of the game by providing nice visuals that fit into the universe and the player’s expectations. The animations should allow the player to assess their current level of battle success and provide the necessary feedback to adjust their strategy.

The animations should be useful, and fluid. The player should not find the animations clunky or unrealistic. Since this game is fast-paced, it is also important that the player not find the animations distracting from making decisions.

Player should feel delighted when focusing on the animations. Player should notice animations when playing the game for the first time, but once the player is an expert, they should not feel bored, slowed, or inhibited by the animation system.

# Spec

In order to have a dynamic and full building combat animation system, the following assets will need to be designed

* Visual effects for all special attacks
  + (Example) Show effects on units if special attack modifies units
* Sound effects

Special Attack Sample Animations

* When the player selects the card, the card spins twice and transforms into a sphere colored with that class’s/type’s color
* The animation plays for the duration of the special attack
* Class/type-based animations
  + *Morale*: all that player’s units have red swirls surrounding them that make them look like they’re “powered up”
  + *Reinforcements*: no extra battlefield animation
  + *Reconnaissance*: modifies the player’s units to start their combat animation sequences from farther away than the normal distance threshold
  + *Energy boost*: the player’s energy buildings have yellow swirls surrounding them, making them look “powered up”
  + *Cyberwarfare*: the enemy’s energy buildings have black swirls surrounding them, making them look like they’re not operating at peak efficiency

# Beat by Beat

1. Player selects a special attack card
2. Player sees the card to special attack animation
3. The effects of the special attack are shown on the battlefield, as well as outlining the edges of the player’s screen with the color of that special attack
4. When the effects of a special attack end, the battlefield colors return to normal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature Number** | **Name** | **Description** | **Priority** | **Difficulty** | **Risks** |
| 13 | Matchmaking System | Matches players with those close to them in skill level to give them a competitive experience | High | Medium | Speed of matchmaking Jeopardizing match quality based on number of people playing |

# Brief

Tenant from Vision Canvas: *“Wager system drives key elements of the game: matchmaking and wagering the outcome of the battle”*

The purpose of this feature is to assess the skill of players and match them with someone relative to their skill in the same wager category. This system is essentially invisible to the gamer as their matches should be fair and seamless enough for them not to question the algorithm involved in matching them with other players.

The feature will factor in a variety of set variables that are measured over the player’s last 20 games so that if they are rapidly improving their matches will quickly rise in skill. This “skill” match after the wager selection makes it so that players are experiencing competitive gameplay.

Although there is ranking system in the game, the player still matching basic on their skill points because the skill point can reflect player’s true skill-level.

The player’s emotional track during this should be neutral. As mentioned earlier, the player will not even be aware of the algorithm used to calculate the “skill” statistic that subsequently matches them with a near equal player.

**Spec**

Once a player makes and posts a wager, they will be placed in a structure with all other players. This structure will be organized into “bins” by wager - players with wagers within a certain range will be placed within the same bins to indicate that all those players are suitable matches.

An algorithm is used to calculate skill. Factors considered include (all within last 20 games): Actions per minute, wins, and total amount of destruction caused. The combination of these three factors should give an accurate gauge as to how good the player is. Once an algorithm is crafted to condense these values into one quantitative statistic called “skill”,

matching players of equal “skill” is easy. An example algorithm could follow the following template:

Skill = (actions per minute) \* 0.3 + (wins) \* 0.4 + (total destruction) \* 0.2

There will also need to be a lobby or waiting room that indicates to the player that the system is trying to find a match for them.

**Beat by Beat**

1. Player enters matchmaking mode
2. Player chose the game mode
   1. ‘Ranked’ or “Unranked”
3. Player specifies a wager
4. Player waits until another player with similar skill and wager is ready to play
5. Both players are locked into their match and move to battle