

- **GAME BOARD** chess board-like, 2d grid, 3d space, being able to see all your “units” at once, strategy involves taking up as much space as possible so darkness can’t take up that space
- each side moves all their units, then the other side, repeat
- max movement distance per unit
- *emit* is an optional ability that each unit has that reduces the strength of one opposing unit
- some units will be ranged, some melee (ranged will have shorter max movement distance)
- menu at end of turn, optional for player to bring up
  - info panel to describe different types of light and darkness
  - panel describing different types of space
  - panel giving current strength of players
  - option to quit
  - option to save
- when units’ strengths (light or darkness) are reduced, they are permanently reduced (unless we create a building that could regenerate that strength)
- energy for each player
  - spend that energy to create light and darkness at generators
  - energy is generated by controlling spaces called “harvester lights/darknesses”
- harvester lights/darknesses harvest energy from certain neutral energy mines
- we will have light points that are created in the generators
  - these light points will be used to create units of whatever type the player wants
  - we will have factory-like buildings that list all possible units that the player can create
  - player selects which type of unit that they want to create
- four types of buildings: energy center(base), generators(creates light or darkness points), energy resource location(harvest energy to produce light or darkness), and factory buildings(create units of player’s choice)
- generators will have a maximum amount of light/darkness points they can create at the end of each turn or day, if we decide to implement that
- the general strategy will be to obtain as much energy as you can from the energy resource buildings, while also balancing that between captured generators;
- each player starts with a set of units, all buildings except energy centers are neutral to start the game

[Advance Wars Example 1](#)

[Advance Wars Example 2](#)

### Iteration 1:

- One level
- One unit type
- A story element
- Units have a light/darkness strength
- Units demonstrate emission

### User Stories:

1. **Gameboard** - Upon the starting the game, the players see the gameboard in its entirety and all of the units and buildings are already laid out.
  - a. **Priority: 10**
  - b. **Time: 50 minutes**
  - c. **TASKS:**
    - i. Develop gameboard by setting the game mode to 3D and creating a 2D grid.
      1. Time: 45 minutes
    - ii. Set camera at desired position.
      1. Time: 5 minutes
2. **Indication Text** - The player who goes first sees text indicating that it is their turn
  - a. **Priority: 10**
  - b. **Time: 1 hour 30 minutes**
  - c. **TASKS:**
    - i. Implement a GUI in the main scene by using GUI GameObjects.
      1. Time: 1 hour
    - ii. Write scripts to display text in the GUI upon the player's turn.
      1. Time: 30 minutes
3. **Starting Units** - Each player starts with some basic units, including harvester units, which start positioned on the game board.
  - a. **Priority: 10**
  - b. **Time: 1 hour 30 minutes**
  - c. **TASKS:**
    - i. Create the unit class with all basic attributes.
      1. Time: 15 minutes
    - ii. Create prefab for basic emission unit
      1. Time: 15 minutes
    - iii. Create prefab for harvester unit
      1. Time: 15 minutes
    - iv. Apply unit class to each prefab
      1. Time: 15 minutes
    - v. Use decorator pattern to extend basic unit classes. [Iteration 2]
      1. Time: 30 minutes

4. **Interact With Units** - The player who is taking their turn can now click on their own units. If a unit has been clicked, the grid positions where that unit can be moved and where it can emit are highlighted in different colors.

a. **Priority: 10**

b. **Time: 15 hours**

c. **TASKS:**

- i. Write scripts to calculate which squares are available to be moved on to, and then update the corresponding GUI.
  1. Time: 2 hours
- ii. Write script that moves a unit when a highlighted square is clicked on. Use Dijkstra's Algorithm to calculate the shortest path, and have the unit continuously move across that path.
  1. Time: 4 hours
- iii. Create an emission type for a dark unit and light unit [**Iteration 2**]
  1. Time: 2 hours
- iv. Write script for when player clicks on an enemy unit that is within range, the selected unit will emit to that enemy unit.
  1. Time: 2 hours
- v. Ensure that if a player clicks on a unit that has already moved in the turn, no squares are highlighted, indicating that the unit can not be moved again.
  1. Time: 2 hours
- vi. If the player clicks on a building within range, the unit will move to that location and start to "occupy" that building. [**Iteration 2**]
  1. Time: 3 hours

5. **Factory** - A player can click on a factory in order to see options in an information panel depending on whether they own the building.

a. **Priority: 30**

b. **Time: 1 hour**

c. **TASKS:**

- i. Develop a building class which will be extended for energy centers, factories, resource nodes, and generators.
  1. Time: 1 hour

6. **Factory Builds Units** - If the player owns the factory, different types of units that can be built are displayed in the information panel. [**Iteration 2**]

a. **Priority: 40**

b. **Time: 1 hour**

c. **TASKS:**

- i. Write script to create different types of units; factors in "light" costs per unit. (The unit pops up next to the factory on the game board, but can not be moved until the next turn.)
  1. Time: 1 hour

7. **Generators** - A player can click on a generator in order to see options depending on whether they own the building.
  - a. **Priority: 30**
  - b. **Time: 1 hour**
  - c. **TASKS:**
    - i. Develop a generator child object of the building parent class [Iteration 2]
      1. Time: 1 hour
8. **Generator Energy Conversion** - If the player owns the generator, they are given the option to convert energy to light points.
  - a. **Priority: 40**
  - b. **Time: 1 hour**
  - c. **TASKS:**
    - i. Update GUI to reflect option to convert energy to light points.
      1. Time: 30 minutes
    - ii. Write script to check player's energy point amounts, and assign new values to energy and light points based on number of times conversion button is clicked.
      1. Time: 30 minutes
9. **Resource Node** - A player can click on a resource (energy) node in order to see options depending on whether they own the resource.
  - a. **Priority: 30**
  - b. **Time: 1 hour**
  - c. **TASKS:**
    - i. Develop different energy nodes as a child of the building parent class (maybe some nodes deliver more energy than others, giving players incentive to choose certain nodes over others).
      1. Time: 1 hour
10. **Player Receives Resources** - If the player is controlling the resource then after their turn has passed, they will receive X amount of resources which they can use on their next turn.
  - a. **Priority: 30**
  - b. **Time: 1 hour**
  - c. **TASKS:**
    - i. Write script to deliver energy to player if they have a harvesting unit on that particular energy node.
      1. Time: 30 minutes
    - ii. Write a script ensuring that only Harvester Units can get energy for a player.
      1. Time: 30 minutes
11. **Player Ends Turn** - The player has an option to end their turn, which they can see as a button on the information panel. This can be done at any point during the turn.
  - a. **Priority: 20**
  - b. **Time: 2 hours 15 minutes**

**c. TASKS:**

- i. Create a GUI button within the information panel which is clickable.
  1. Time: 15 minutes
- ii. Write script to display a GUI indication that it is the next player's turn, and then change which objects are available to be moved.
  1. Time: 2 hours

**12. Player Command Centers - Each player has one energy center.**

**a. Priority: 10**

**b. Time: 10 minutes**

**c. TASKS:**

- i. Create energy center prefab.
  1. Time: 5 minutes
- ii. Place energy center in correct locations for the level (probably opposite sides of the game board).
  1. Time: 5 minutes

**13. Player Win / Lose - If this center is occupied by an enemy then the player loses the game.**

**a. Priority: 10**

**b. Time: 45 minutes**

**c. TASKS:**

- i. Write script to check if the center has been occupied every turn.
  1. Time: 15 minutes
- ii. Write script to change GUI to reflect the end of the game.
  1. Time: 30 minutes