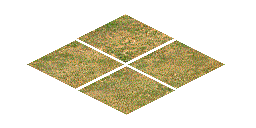
1. Xây dựng bản đồ vùng chiến sự

* Vấn đề :

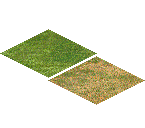
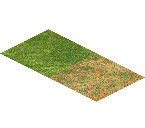
Bản đồ vùng chiến sự là nền để sắp đặt và bố trí các Sprite lên đó, bao gồm Terrain, Resource center, Unit và Structrure. Bản đồ có kích thước lớn hơn nhiều so với vùng nhìn của cửa sổ (Viewport) cho phép người dùng kéo vùng nhìn của cửa sổ trên khắp bản đồ để quan sát cục diện trận chiến trong Game, đồng thời bản đồ cần phải có tính linh động cao trong việc thay đổi kích thước. **Vậy, vấn đề ở đây là xây dựng một bản đồ chiến sự lớn, có khả năng tùy biến về kích thước**.Có hai giải pháp được nêu ra trong việc xây dựng bản đồ vùng chiến sự.

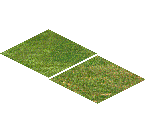
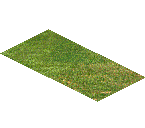
* Giải pháp
  + Load một ảnh thật lớn và vẽ ra như một bản đồ chiến sự trên màn hình Game. Nhưng khuyết điểm của giải pháp này là XNA framework chỉ cho phép load hình ảnh có kích thước tối đa là **2024 x 2024 px**, như vậy nếu load một ảnh thật lớn cũng chỉ có thể load được kích thước 2024px cho mỗi cạnh, điều này đã làm giảm tính tùy biến về kích thước của bản đồ chiến sự. Tuy vậy nếu chấp nhận giải pháp này có thể làm giảm ảnh khuyết điểm trên bằng cách vẽ hình đã load với kích thước khác nhau, tuy vậy cách này sẽ làm giảm chất lượng của ảnh được vẽ, thử tưởng tượng ta có một ảnh kích thước 128 x 128px thử resize nó thành 256 x 256 px, hiển nhiên hình ảnh sẽ không còn đẹp nữa. Cần thiết phải có một giải pháp tối ưu hơn.
  + Giải pháp thứ hai được nêu ra, thử xem bản đồ vùng chiến sự là một tập hợp nhiều ảnh nền nhỏ ghép lại, kích thước bản đồ tùy thuộc vào số lượng các mảnh ghép vì thế, khi tăng số lượng các mảnh ghép, ta được một bản đồ có kích thước to hơn. Việc vẽ bản đồ được thực hiện bằng vẽ nối tiếp các mãnh ghép trên vùng nhìn.



Như vậy, các bước thực hiện cho giải pháp này là

* + - Xác định tính đúng đắn của các mãnh ghép(sự chuyển tiếp giữa các vùng đất trên bản đồ).

* + - Từ đó xây dựng được ma trận các mảnh ghép.
    - Xác định vị trí của mỗi mãnh ghép bản đồ, sao cho khi vẽ ra, các mãnh ghép nối tiếp nhau.
    - Thực hiện thao tác vẽ chỉ cho các mãnh ghép nằm trong vùng Viewport và không thực hiện vẽ các mãnh ghép nằm ngoài cùng Viewport vì điều này là không cần thiết.

1. Quản lý Structure và Unit trên map sao cho truy vấn nhanh

* Vấn đề:

Các Sprite trên vùng chiến sự có số lượng rất nhiều, và chia thành nhiều loại bao gồm quân lính(Unit), công trình(Structure)… vấn đề đặt ra là làm cách nào để quản lý và truy xuất các Sprite theo từng loại nhanh và dễ.

* Giải pháp

Chia các Sprite thành các mảng con mà các phần tử là một loại Sprite muốn quản lý

* Mảng các Unit hiện có trong BattleField
* Mảng các Structure hiện có trên BattleField
* Thực hiện vòng lặp trên 2 mảng này

1. Xác định 1 unit hoặc 1 structure trong phạm vi tấn công

* Vấn đề
  + - Trừ health
    - Xác định đối tượng tấn công có còn trong phạm vi tấn công hay ko(đối phương có thể đã bỏ chạy)
    - Đối phương đã die hay chưa(Nếu còn health thì còn đánh, nếu hết health thì ko tấn công nữa)
* Giải quyết:
  + - Một Unit cần biết nó sẽ attack ai 🡪 Whom I Hit

🡪 thêm thuộc tính **\_whomIHit**

1. Structure hoặc Unit đã hết health 🡪 không còn tồn tại trên map 🡪 dispose
2. Producer Unit
   * Xác định vị trí resource center để lấy

**Giải quyết:**

* + - * Duyệt mảng các resource center trên bản đồ
        + Phát hiện ra resource center mà nó có thể khai thác
        + Khai thác resoure center này nếu

Resource mà nó đang mang trên người số lượng chưa max

Nó hiện ko có mang resource nào

Resource center mà nó muốn khai thác khác resource mà nó mang trong người

* + - * Kiểm tra khi nó di chuyển, nó đã chạy ra ngoài phạm vi của resource center để có thể khai thác resource chưa. Nếu chưa ra khỏi phạm vi 🡪 khai thác tiếp, nếu đã ra khỏi phạm vi 🡪 đứng yên
  + Xác định vị trí nhà chính để cung cấp resource
  + Tự động bỏ chạy khi bị tấn công hoặc đứng yên để cho đối phương tấn công(power = 0)
    - Chạy hướng nào, xa bao nhiêu

**Giải quyết:** chưa có

* + - (Hoặc tiếp tục khai thác cho đến khi bị đánh chết 🡪 máu phải thật nhiều để chịu đựng cho tới khi người chơi di chuyển nó bỏ chạy)

1. Hiệu ứng âm thanh (sound effect)
   * Có 2 cách tiếp cận
     + Dùng XACT(nói sau)
     + Không dùng XACT 🡪 sẽ dùng class SoundEffect hoặc SoundEffectInstance, cách này đơn giãn hơn dùng XACT

(XACT không hoạt động trên Zune, nên cách 2 sẽ là 1 tùy chọn tốt cho các game trên Zune)