Michael Petitclerc

March 7th, 2017

Rilla Khaled

CART 353

Artist Statement

Sonar Defense is a tower defense inspired game where enemies funnel through pre-determined paths\* towards the players main base. Towers are placed along said paths to attack oncoming enemies and destroy them before they reach their destination. The kick in Sonar Defense is that vision is limited to a small radius around towers. To determine enemy type, weakness, and location, the player must use sonar and sound acquisiton. Sonar establishes enemy location and possible direction while sound acquisition further establishes exact position as well as sound hints to the kind of enemy found.

There will be multiple entry points into the players base, pushing the use of previous mechanics to appropriatly prepare. Also towers will be salvageable with little consequence to allow proper rotation of assets to suitable areas of defense.

Sonar Defense is a unique twist in an otherwise dead genre. With little pertinent releases in the past years (each being unique themselves), it is crucial that Tower Defense games offer something new to gain attention. With the elimination of vision (a given in most versions), emphasis on sound and new ways of placing towers (instead of the usual maze) will provide a refreshing take to the genre.

Sonar is rather simple, ellipses will spawn and grow from the click point and will spawn a visual cue directly over enemies that it collides with. Sound acquisition works with distance, the further the mouse from the enemy, the quieter the sound. Both these mechanics simply calculate the distance between two points and when small enough, the visual/sound cue starts.

Vision limitation through fog of war remains tricky, simply due to multiple objects interacting with a single image. At its core, a fog layer calculates and translates distance from x and y to a limit to opacity at the pixel level creating circles of vision for the player.

Minion and Tower interaction is a matter of fetching specific objects variable when close enough to each other. (reduction of hp, strengths versus weaknesses, etc...)

Libraries will aid with GUI impletation as well as pathing for the enemies to follow. (Sprites, G4P, Path Finder Library, Minim)