



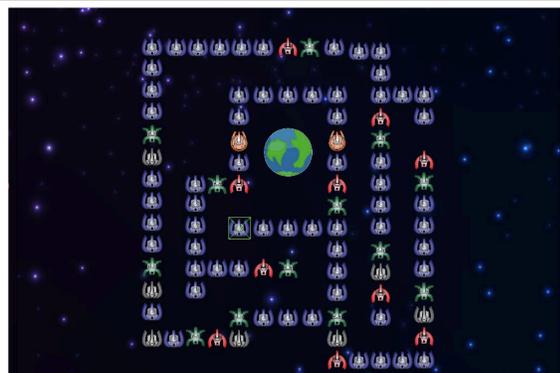
# iOS Game: Planet Tower Defense

Daniel Grewal • Kevin Scrivnor • Comp 499-03

## Introduction

Planet Tower Defense is a mobile game targeted for the iPhone and iPad. Planet Tower Defense combines the strategy of a tower defense game with the action of a twin stick shooter. The idea behind this game is to take the classic tower defense format and expand it to create a more complete and interactive mobile experience. Planet tower defense takes a classic tower defense game and expands the players creativity by allowing them to place towers anywhere on the map to create their own mazes!

## Create Unique Defense Strategies



## Process and Goals

The overall goal of this project was to learn a new programming language and implement an agile development approach to replicate a professional project.

Software development skills used:

- A\* / Uniform Search Pathfinding algorithms
- Graphics algorithms
- Object Oriented Programming
- Swift 2.0
- Apple's SpriteKit
- Github Source Control
- 2D Sprite creation / Photoshop



Gameplay screenshot

## Gameplay

Planet Tower Defense is at its core a tower defense game:

- Home planet: The player has a home planet at the center of the world that it must defend.
- Towers: Player can place stationary towers, that kill attackers, anywhere around the planet to create a maze.
- Attackers: Alien attackers spawn from all 4 sides and find the shortest path around the towers to the home planet.
- Defense Ship: Player gets an upgradeable ship that is twin joystick controlled.
- Items: Special items drop from the attackers (freeze, planet HP, etc)
- Points: Used to buy towers and ship upgrades.

Goal: Survive the most waves

Start: Player starts with some points to place towers. Attackers spawn in waves and take the shortest path available to the planet. The player must use towers and their ship to survive as many waves as possible.

End: If the players planet reaches zero hp the player loses.



Game over screen

## Conclusions

This project overall was an interesting undertaking. Learning a new language on the fly and implementing it was enlightening to some gaps in my knowledge. It was also good to see all the parts that go into the actual development process; the artwork, the game logic, algorithms, reworking code, adding features, etc. It definitely would have been nice to have someone else do the artwork since I don't particularly enjoy drawing. Certain game logic was harder than expected, such as figuring out the ratios between attackers and damage and figuring out how each wave should scale. The agile approach, as I knew it would, had some interesting downsides. Adding in some features later in the project caused some code rework that could have been avoided with some preplanning. However not having a real idea of what I wanted the game to be when I started made the agile approach very effective.

Overall the project was a success. I learned a new language and used many programming practices that can be used in a professional setting.

## Future Additions

- Different game modes: Could be a timed feature, or a challenge mode that restricts tower placement.
- Difficulties: Give the player difficulty choices to scale to player ability
- Map variations: Give the player some background options or different home planet positions and attacker spawns to vary the game a little bit.
- Expand towers, upgrades, items and attackers: Create more images and types for creeps, items, ship upgrades and towers.
- Tutorial notices: It isn't the most difficult game but maybe some tutorial or help screen for new players.