**Tower Wars**by Andrés Herrera

**Target Audience:** Competitive players from all ages  
**Genre:** Strategy, Tower Defense  
**Target Platforms:** Android, PC  
**Number of Players:** 1-2

**High Concept**

A multiplayer tower defense game in which you send minions to defeat your enemy’s castle while building towers to defend yours.

**Features**

* Challenge your friends in a real time 1v1 battle
* Play in a map with a fixed path or build your own maze with towers
* Top-down isometric view
* Multiple towers with multiple different upgrade paths
* **Game Phases**: For each wave
  + **Building phase**: Use your gold to build and upgrade your towers
  + **Deployment phase**: Use your gold to send your minions to attack your enemy castle
  + **Strategy phase**: Choose the best attack patterns for your towers for the upcoming wave
  + **Fighting phase**: Watch as the action unfolds
* Get gold by killing enemy minions with your towers, and get an extra income per wave that can be increased by sending minions
* The first player to kill the enemy castle wins, so protect yours at all cost

**Player Motivation**

Challenge your friends and prove them who is the best. Choose the best minions to destroy your enemy castle and use the best towers to defend yours.

**Design Goals**

**Competitive**: Battle with a friend to prove who is the best

**Strategic:** Strategize for the best combination of minions and towers to best your opponent

**Replayable:** Each game is different, with many maps, towers, minions and upgrades to choose from

**Team Roles**

* **Gameplay developer**: To code the core functionality of the game
* **3D modeler**: To model the towers and minions
* **3D animator**: To create minion’s movement, attack and death animations. Also, to create tower building and attack animations
* **Level designer**: To balance the towers, minions, maps and waves.
* **Networking engineer**: To design the networking for the real time gameplay
* **Algorithm designer**: To design a neural network able to learn from each player