**Wave requirements**

• The game has at least 5 waves of enemies, each wave is more difficult than the previous one.

• Properties of a wave can be configured in the Unity Editor without changing the code, e.g. one or multiple of: enemy types, enemy amounts, enemy combinations, delay between enemies, percentage of each enemy type, chance of spawning, etc.

• In between waves the players have a short building phase to sell/destroy, build and upgrade the towers.

**Enemy requirements**

• Enemies are spawned from spawn point(s)  
• Enemies follow a path, or multiple paths to the end goal  
• When x enemies reach the end goal it is game over (tweakable level property)  
• Enemies have at least three properties (health, speed and carried money)  
• There are at least two types of different enemies  
• When an enemy unit is destroyed, player gets the cash  
• Attributes and visuals of enemies can be set up in the Unity Editor without hard coding.

**Tower requirements**

• Towers can be built on grid cell along the enemy path  
• One tower-gridcell can only hold one tower (towers can’t overlap with each other).  
• It should be clear to the player what towers can be bought with the current amount of money.  
• There are at least three different types of towers:  
\* Single target attack tower

\* AOE attack tower

\* Debuff attack tower (slows downs enemies, or another effect, the same effect never stacks)

• Towers can be upgraded with money; the upgrade system can be adjusted in the Unity Editor without changing the code.  
• It should be clear which towers can be upgraded with the current amount of money

**GUI requirements:**

The following information should be displayed in clear ways in the game:

• Wave number.   
• Total money.  
• Time left for the player to build/upgrade towers before next wave begins.  
• Where can the towers be built.  
• Cost for building/upgrading each tower.  
• Available and unavailable towers and upgrades.   
• Health bar of each enemy.  
• Amount of money dropped by each enemy when killed.  
• How many enemies can enter the end point before game over.  
• Game over.  
• Player wins the game.