Game Development Proposal: Battle Heroes Defense

(Inspired by Line Rangers)

1. Project Name

Battle Heroes

2. Project Overview

Battle Heroes Defense is a 2D side-scrolling tower defense game where players deploy different hero units to attack the enemy base while defending their own. The game integrates real-time strategy mechanics, AI-controlled enemies, and resource management.

Key Mechanics:

- Hero Deployment – Players click on hero buttons to deploy different types of heroes with unique stats.
- Cooldown System – Each hero summon button has an individual cooldown time.
- Energy System – Summoning heroes costs energy (minerals), which regenerates over time and can be upgraded.
- AI Enemies – Enemies spawn automatically and try to destroy your base.
- Base Defense – Defend your base while trying to destroy the opponent's.
- Win/Loss Condition – Game ends when either base reaches 0 HP.
- Hero Skills – Certain heroes have special abilities (skills) with individual effects and cooldowns.
- Projectile System – Skills like AOE create visible projectiles that hit enemies.
- Pause & Analytics – A pause menu allows players to view battle analytics in a Tkinter window.
- Stage Progression – Players can advance to the next stage after winning.

3. Project Review

Existing Game Analysis: Line Rangers

- Character-based battles with different attack types.
- Progressive difficulty where enemies get stronger over time.
- Resource management for hero deployment.

Proposed Improvements in Battle Heroes Defense

- Smarter character behaviour: Hero with varied behaviors and attack strategies.

- Improved Graphics: Animated sprites instead of static blocks.
- More Hero Types: Tanks, ranged units, and special abilities.
- Hero Summon UI: Individual hero buttons with cooldowns and energy requirements.
- Data Tracking: Player stats for performance analysis.
- Skill System: Some heroes have unique skills (e.g. power shots, buffs) that activate during combat.
- Stage Progression System with dynamic difficulty.
- Analytics Visualization: Tkinter-based graphs and tables showing real-time data.

4. Programming Development

4.1 Game Concept

Game Objective:
Destroy the opponent's base while defending your own.

Gameplay:

- Deploy heroes strategically by clicking on buttons.
- Manage your energy resources and cooldowns.
- Use different hero types based on the situation.

Controls:

- Click UI buttons to summon heroes.
- Each button has a cooldown and costs energy.
- Enemies spawn automatically.

4.2 Object-Oriented Programming Implementation:

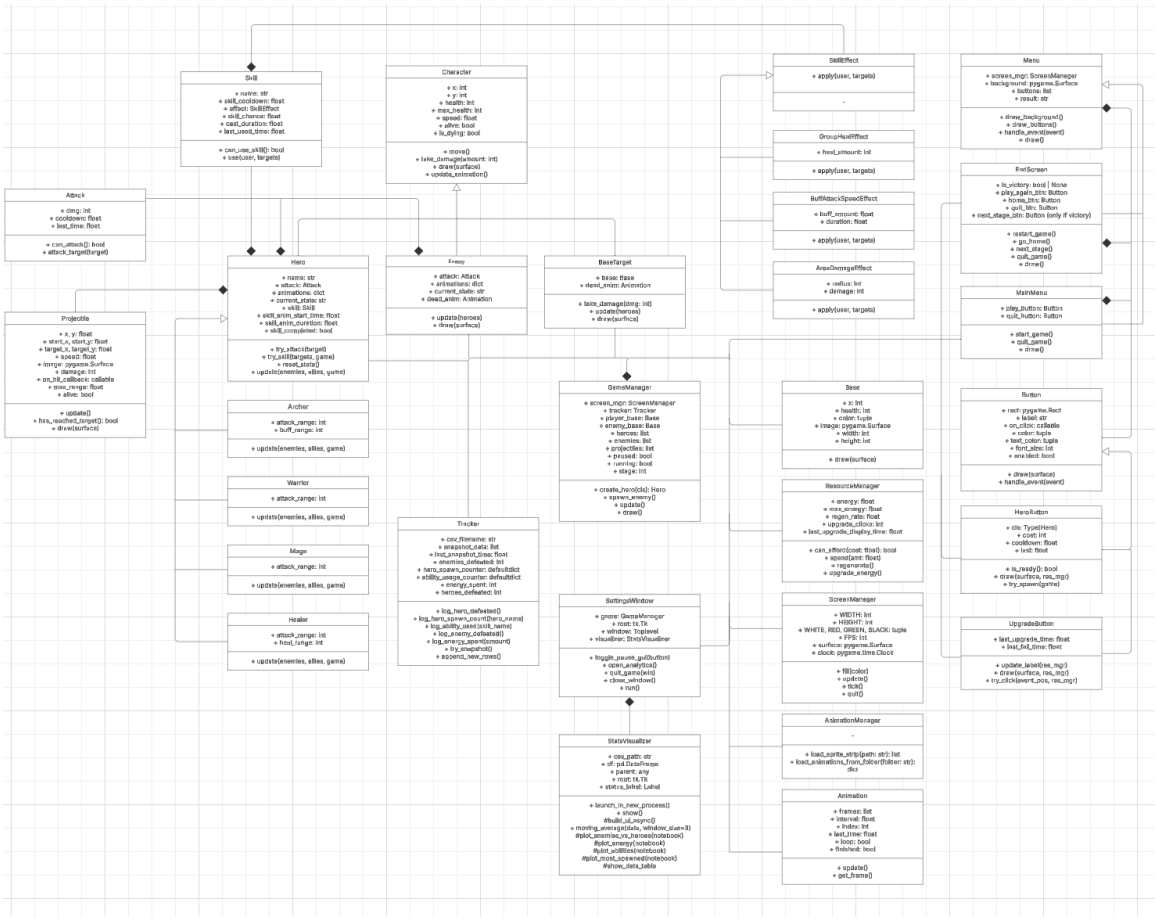
- Active skills (e.g., power attacks, heals, buffs)
- Cooldowns
- Energy or resource cost
- Skill targeting (self, ally, enemy, area)
- Projectile behavior (AOE, targeting enemies)
- Visual/sound effects (future improvement)
- Pause menu using Tkinter Toplevel
- BaseTarget class to allow base interaction

Update to Class Table with Skill

Class Name	Description
`GameManager`	Main controller for the game loop, states, unit updates, and stage logic.

`Character`	Base class for units with position, health, movement, and alive status.
`Hero`	Inherits `Character`; adds attack, skill, animation, and special behaviors.
`Enemy`	Inherits `Character`; implements attack logic and AI movement.
`Base`	Static base image with HP bar and scaling.
`BaseTarget`	Inherits `Character`; makes base attackable like a unit.
`Skill`	Contains logic for cooldown, skill chance, and effect application.
`SkillEffect`	Abstract class for different types of skill effects.
`BuffAttackSpeedEffect`	Applies temporary cooldown reduction to nearby allies.
`AreaDamageEffect`	Damages all enemies in range.
`GroupHealEffect`	Heals all allies in range.
`Projectile`	Moves from source to target with range check and damage on contact.

UML:



link
https://lucid.app/lucidchart/6a48afa7-1e3b-480f-ab15-1587b6981417/edit?viewport_loc=-2857%2C-2489%2C7516%2C3896%2C0_0&invitationId=inv_f42b5e9f-a717-4fcb-be57-2f106346f147

4.3 Algorithms Involved

- Pathfinding Algorithm: Basic linear movement logic for heroes and enemies.
- AI Behavior Algorithm: Enemy spawning at intervals and targeting based on proximity.
- Event Handling: Mouse click events trigger hero spawning and upgrade actions.
- Cooldown Management: Tracks individual button and skill cooldowns using timestamps.
- Damage Calculation: Skills and attacks subtract HP from targets; death is triggered at 0 HP.
- Energy Regeneration: ResourceManager adds energy over time and caps it at max value.
- Skill Activation Logic: Skills check cooldown and chance before applying effects.
- Effect System: Effects can deal AOE damage, heal allies, or buff attack speed.

5. Statistical Data (Prop Stats)

5.1 Data Features

- Total Energy Used– Measures player resource consumption for deploying heroes.
- Enemies Defeated– Tracks how many enemies were eliminated in each snapshot.
- Heroes Defeated – Tracks the number of fallen allied heroes per snapshot.
- Most Spawned Hero– Captures the most frequently summoned hero every 5 seconds.
- Attack Speed Buff Usage- Number of times the Archer's buff skill was activated.
- AOE Skill Usage- Number of times the Mage's area damage skill was activated.
- Group Heal Usage- Number of times the Healer's healing skill was used.

5.2 Data Recording Method

• Storage: Data is saved in CSV files using Python's `csv` module.

• Visualization: Tkinter GUI displays graphs and tables from data using Matplotlib and Pandas.

5.3 Data Analysis Report

- **Line Chart** – Displays the trend of total energy used over time (per 5-second snapshot).Bar Graph: Enemies defeated per session.
- Bar Graph – Shows the number of enemies defeated and heroes lost per snapshot session.
- **Strip Plot** – Tracks most spawned hero over time, showing player preference per snapshot.
- Data Table View – Shows raw data of each 5-second snapshot for deeper review and debugging.

5.4 Data Table (with X-axis and Y-axis)

Feature	Why is it good?	How to obtain 50 values?	Variable/Class	X-axis	Y-axis
Total Energy Used	Shows how much players spend over time.	Snapshot every 5 seconds in battle.	`energy_spent / Tracker`	Session	Energy Used
Enemies Defeated	Indicates combat performance and difficulty.	Count enemies killed in each snapshot.	`enemies_defeated / Tracker`	Session	Kill Count
Heroes Defeated	Tracks player losses and team survival.	Count fallen heroes per snapshot.	`heroes_defeated / Tracker`	Session	Fallen Hero Count

Feature	Why is it good?	How to obtain 50 values?	Variable/Class	X-axis	Y-axis
Most Spawned Hero	Reflects player unit preference.	Get most frequent hero in snapshot.	`hero_spawn_counter / Tracker`	Session	Hero Name
Attack Speed Buff Usage	Tracks how often Archer's skill is used.	Count Buff activations per snapshot.	`ability_usage_counter / Tracker`	Session	Buff Count
AOE Skill Usage	Tracks how often Mage uses AOE.	Count AOE activations per snapshot.	`ability_usage_counter / Tracker`	Session	AOE Count
Group Heal Usage	Tracks how often Healer uses group heal.	Count Heal activations per snapshot.	`ability_usage_counter / Tracker`	Session	Heal Count

6. Conclusion

Battle Heroes Defense provides an engaging tower defense experience that emphasizes tactical hero deployment, resource management, and real-time combat. With analytics integration and a variety of hero abilities, players receive both strategic depth and visual feedback. This foundation opens paths for future expansion, balancing, and advanced features.

- Future additions could include:
- • Boss waves and elite enemies
- • New hero types and stages
- • Save/load progress system
- • Online scoreboards and challenges

7. Project Timeline

- Week 1 (10 March): Proposal submission / Project initiation
- Week 2 (17 March): Full proposal submission
- Week 3 (24 March): Hero buttons, skill classes, energy system
- Week 4 (31 March): Data tracking and enemy balancing
- Week 5 (7 April): UI polish and game analytics
- Week 6 (14 April): Final testing and submission
- Milestones:
- 50% by 16 April – Core mechanics, tracking started
- 75% by 23 April – Skills and UI complete, analytics working

- 100% by 11 May – Final polish, graphs and submission

8. Version

Version: 5.0

Date: 11 may 2025