

Here is the logbook for the tests, please let me know if you have any questions.

DateTimes

Thursday 22nd:

- Research
 - Trello configuration: 30m
 - Tower defense gameplay: 2h
 - Unity navigation AI: 30m
 - Unity UI: 45m
 - Setup project
 - Unity Tests
- Production
 - Project creation and setup: 1h
 - Failed Unit tests creation: 2h

Friday 23rd:

- Camera, navmesh, and first creep: 2h

Saturday 24th

- First turret: 2h
- UI for Main Menu and HUD: 2h

Sunday 25th

- Economy: 2h
- UI for economy and tweaks: 2h

Monday 26th

- Performance UI for health bar: 1h
- Waves: 2h
- Art tweaks: 2h

Tuesday 27th

- Different turrets: 1h
- Difficulty tweaks: 1h
- Prepare deliverables: 2h

Problems and decisions

- I was unable to configure Unit testing in a decent amount of time, giving the schedule I planned for the test. The problem was related to Assembly creation and some missing symbols from DOTween. I decided to continue without Unit Testing.
- Some of the tower defense I investigated used a grid to place the turrets, but it felt a bit constrained, so I decided to go for a more organic placement system.
- I decided to go with navmesh since I think it is the most scalable option for this type of game.
- Also I used an advanced UI system, called DoozyUI, it makes the configuration of the UI flow and animations easy and powerful.
- I investigated DI systems but the time for a proper investigation felt not enough. So I did a "Systems" class with the system as static classes, with this I can use a simple dependency injection.
- Decided to go for a simple economy for purchasing turrets and rewarding when creeps are killed. I was tempted to go for a "rogue like" approach and make the game easier with some boosters, but I didn't have time to do it properly.
- I decided to use waves to have a better customization for the difficulty, and have a repeatable last wave with increasing difficulty.

Possible refactors and improvements

- Investigate why DOTween assembly is incomplete, and create some Unit Tests
- Automation Tests
- It should be relatively easy to refactor the code a bit to create different creeps, I will need to separate some behaviours like I did with the turrets.
- Disengage some of the system interactions using Actions
- For different scenes, I will change the placeable components, since now it is not scalable enough
- Using the navmesh and some triggers it should be easy to attack the turrets if they are blocking the path
- General improvements in the UI layouts
- Debug/stats panel for easier testing
- Investigate if a pool of enemies will increase the performance vs using directly the prefab
- Remove some of the code from GameController
- Click and drag for turrets to place them, it may be more intuitive?

- Cancel turret when placing
- Sell turret