



Prep week project





/TABLE OF CONTENTS



/01 Task intro



/02 Js task



/03 Html css tasks



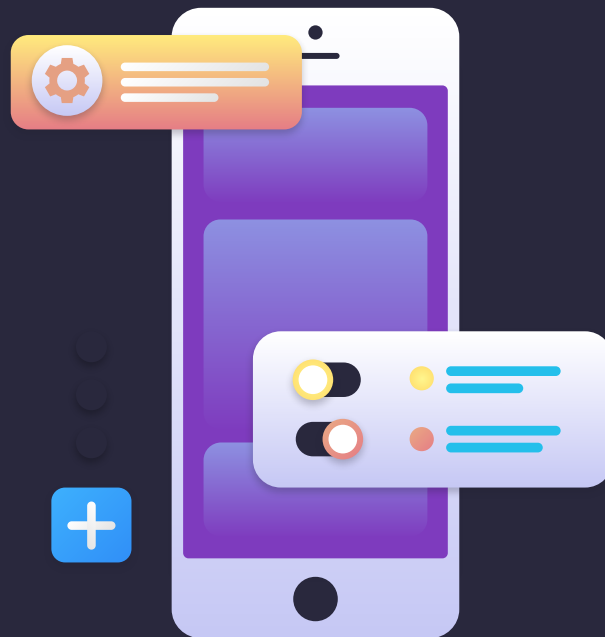
/04 tips





/01

intro





intro



- Today we will create a program, simulating an rpg/action game.
- This is a big complex game - the trick is to break it down to **the simplest core** functionality and pages, and then add more as the project is continuing.





intro



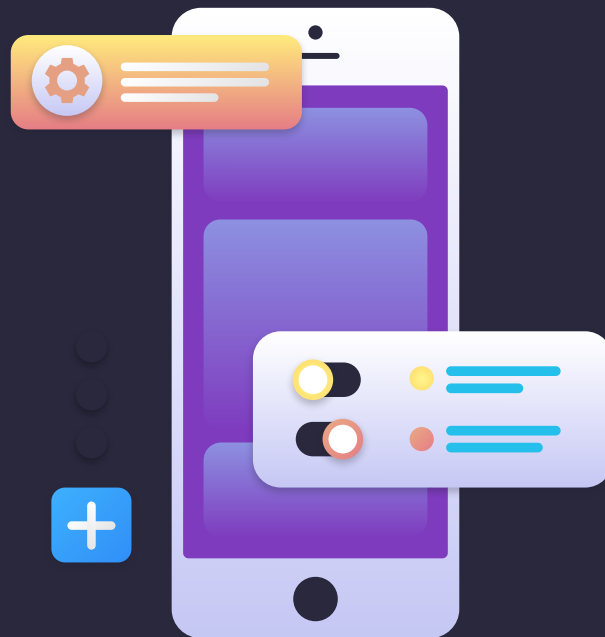
- The core of the game is a combat between a player and a monster.
- The combat is continuing until the player stops, or dies.
- The idea is that the full game is a list of locations, in each location there are monsters with gold and xp rewards, and the player can fight them.





/02

JS





JS



- Again, try to move from simple to complex.
- Each functionality you see, try to create the simplest version of it, and add features and complexity later.
- This means that you don't have to create everything as I describe here, you can start with some of the features, and add the rest later.





JS



- The two main entities are the player and the monsters.
- The player should have the following specs:
 - Strength (will be used to determine damage)
 - Defence (will be used to determine damage)
 - Dexterity (for who attacks first)
 - Gold amount
 - Potion amount
 - Level (will be used to determine damage)
 - Health points
 - Current XP





The monster should have:



- Name
- Strength
- Defence
- Level
- Health points
- Xp reward
- Dexterity
- gold



JS



- The core of the game is the combat.
- This is the combats flow:
<https://docs.google.com/drawings/d/1rn9VGMC-Y3xsPUn2DuMGmVm-h34ixtAl0ReuuSRVrXc/edit?usp=sharing>
- First we decide who is starting - player or monster.





JS



- How do we decide who starts?
 - Player and monster roll d20 (random number 1-20)
 - Each add their own dexterity score to the roll results.
 - The higher result - starts.
 - If equal - roll again.





JS



- Then we need to attack.
- Attack is calculated like so:
 - Roll D20 (random number 1-20)
 - Add the roll to the attackers strength
 - Multiply the results by attackers level.
 - Remove from the results the defenders defence score.
 - If results is > 0 : this is the damage.





JS



For example:

Player:

Strength: 5

Dex: 2

Defence: 3

Level: 2

Life: 60

Who starts?

Player roll 14, monster roll 10.

Player score: $14+2 = 16$

Monster score: $10+4 = 14$

Player starts.

monster:

Strength: 4

Dex: 4

Defence: 2

Level: 1

Life: 60

Player attack:

Roll 14.

Score: $(5+14)*2 = 38$

Subtract defence: $38 - 2 = 36$

Subtract from monster's life:
 $60-36=24$.



JS



- Additional things to create before the game begins:
 - An array of locations. Create at least 5. Create them at random, using an arrays of words. For example:
["the dark", "the village", "the scary"]
["Mim", "ragnarok", "bogo"]
= the dark ragnarok, the village bogo, the scary Mim





JS - additional features



- Additional things to create before the game begins:
 - Match each location with random number of monsters between 5-10.
 - For each monster create a contract - monster name with the number of gold and XP to be earned, if defeated.
 - Create functions that list the locations and the contracts within a location.



JS - additional features



- Once the game starts, a player can decide to which location to go. A player can fight only the monsters in the same location.
- Once in a location, sees all the monsters in the location, with their gold and xp rewards.
- The player can select a monster to fight, and then will be transformed to the combat section.

JS



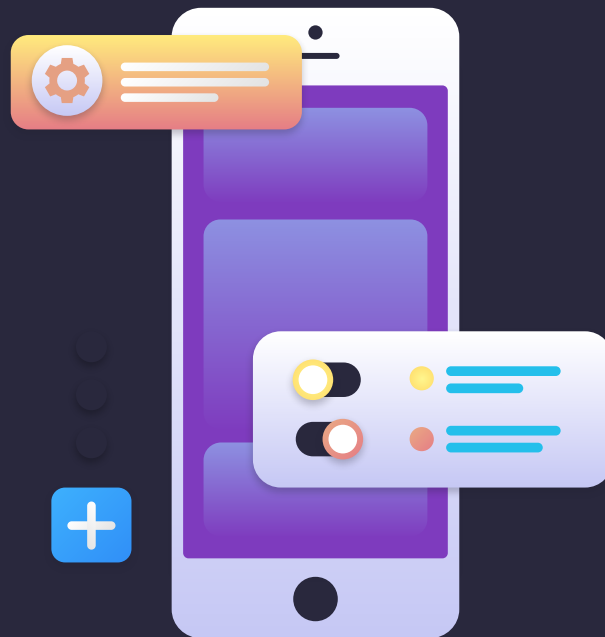
- Tips:
 - Break the functionalities to the smallest possible part.
 - Use OOP principles or FP principles to make your code better.
 - Try to break down each section with as little as possible overlap - so you can work within the group without blocking





/03

HTML CSS





HTML CSS



The main pages are:

- **Fight screen – the most important**
- Character creation
- Locations map
- Location page
- Shop screen





HTML CSS



- fight screen:
 - Image for the player on one side, and the monster on the other side.
 - Life for each are presented as a number or as a health bar (better!).
 - Buttons for attack or run away
 - Suggested design:
<https://i0.wp.com/www.useapotion.com/wp-content/uploads/2020/09/rpg01.jpg?resize=1024%2C576&ssl=1>





HTML CSS



- Character creation:
 - A form with the specs we listed in JS part.
 - Suggested design (use images instead of 3d backgrounds/modals):

<https://i0.wp.com/www.icicledisaster.com/wp-content/uploads/2020/05/dark-souls-jrpg-with-character-creation.jpg?fit=1024%2C576&ssl=1>





HTML CSS



- Locations map:
 - Buttons with the names you gave to locations, ordered in an visually interesting way.
 - Suggested design:
<https://www.rpginsanity.com/wp-content/uploads/2019/01/trial-tower-map.png>



HTML CSS



- Location screen:
 - A message board with all the monsters contracts.
 - Each message shows the amount of gold and xp to be won, a monster picture and a button that will lead the player to fight this monster.
 - Suggested design:
<https://damnellpress.files.wordpress.com/2020/02/tavernboard001.png>





HTML CSS



- Shop screen:
 - Images of health potions, the players gold amount, potion amount and a button to buy.
 - Suggested design:
<https://i.pinimg.com/736x/dd/86/b4/dd86b481a9b695117bdd1b8b5ef6fbdc--ui-patterns-mobile-game.jpg>





HTML CSS



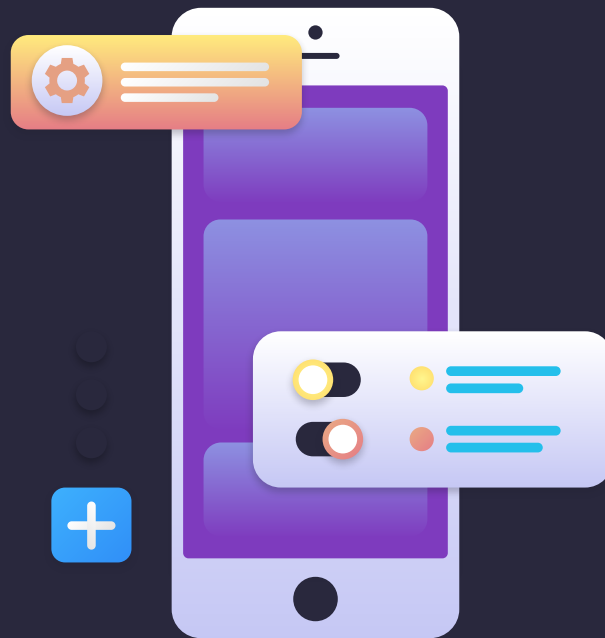
- Connect pages when its logical (for example – connect shop screen to location page, for when you are done shopping).





/04

Tips





TIPS



- This is a complex task.
- Take time for planning. Allocate time, priorities. **Use github issues, milestones and projects! Track your time on each task**
- Start from implementing the simple functions or UI elements.
- You can use this full game flow chart:
https://docs.google.com/drawings/d/1ilbp4_kDSblSdLrmeC5J_XMHTmeWyc_2GVbmCViHVRy/edit?usp=sharing



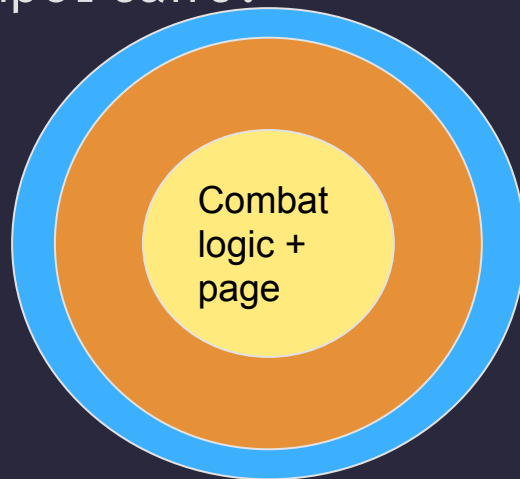
TIPS

- Game priorities -
inside more important:

Create player
page and logic

Potion shop
page and logic

Potion shop
page and logic



Generate
locations

Locations page

Select specific
location and
load monsters
inside

Inside a
location -
location page
with all the
monsters and
rewards



TIPS



- For the responsive design, you can use these principles:
 - <https://uxplanet.org/principles-of-mobile-site-design-c4c721693c42>
 - https://www.thinkwithgoogle.com/_gs/documents/538/multi-screen-mobile-whitepaper_research-studies.pdf



Thanks, and good luck!

Contact us:

Yishai

yishain@appleseeds.org.il

0524605642

Shadi

shhadys@appleseeds.org.il

0543113297

