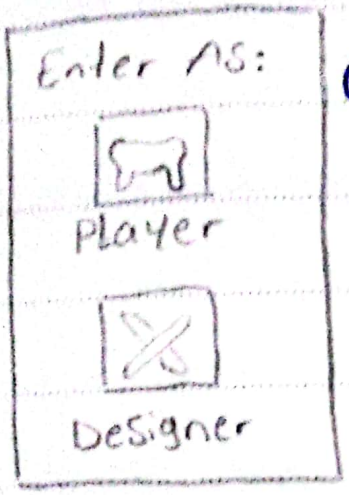
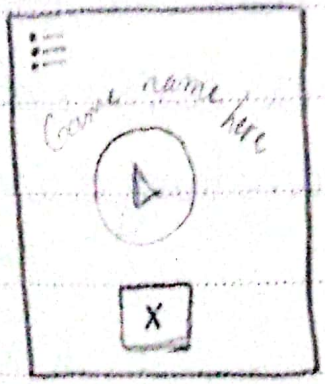
Single or multiword puzzle

Idea: A class that describes every level and holds related data (Image, words, Correct ans, animation)

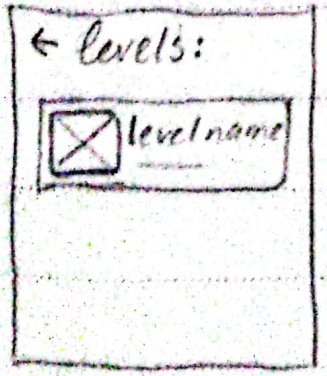


0) Menu Screen:
Container for 2 related apps:
the game itself and a Configuration app that allows adding new levels.

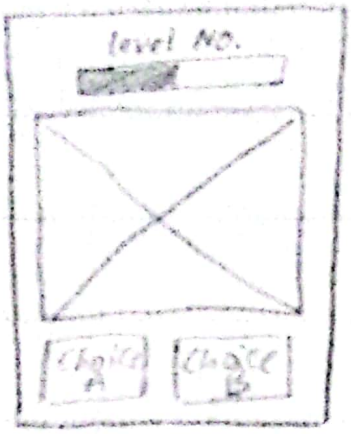
1) The Game:



1) Game main menu:
▷ : to start Game
☰ : to check existing levels
X: the no. of generated levels



2) Existing levels:
A prefab that will be filled with class data. (clickable to replay level)



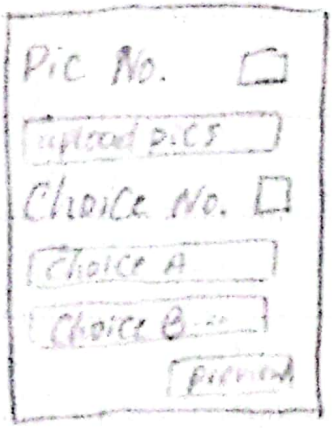
3) the level:

A level might Contain more than one action hence the bar at the top.

the Image and choices are from class data.

---

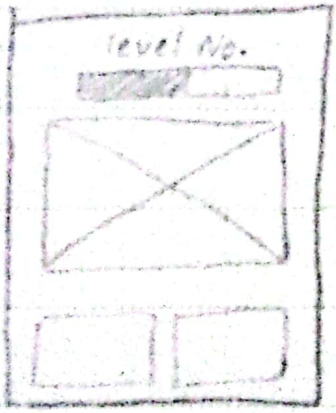
② the Designer app:



1) the upload page:

Each element Could have more than one instance hence predefine no. before upload.

Preview button: to check every thing is working.



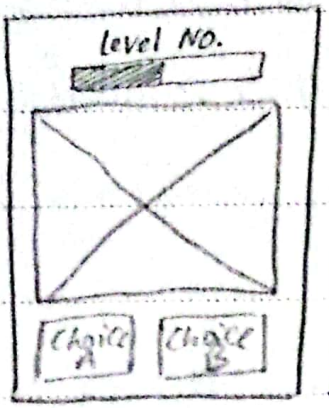
2) the previewed level:

Exactly as a level would be and is playable. It's only *intention* is to see every thing in real time.

NB: upload order is important

M T W T F S S

DATE:



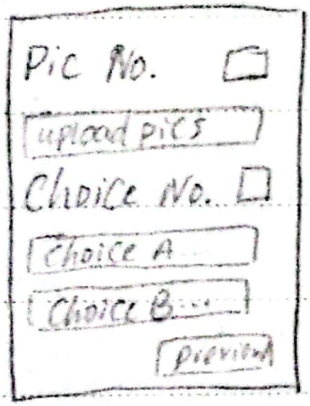
3) the level:

A level might Contain more than one action hence the bar at the top.

the Image and choices are from class data.

---

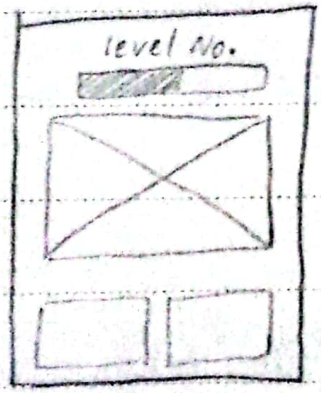
② the Designer app:



1) the upload page:

Each element Could have more than one instance hence predefine no. before upload.

Preview button: to check every thing is working.



2) the previewed level:

Exactly as a level would be and is playable. It's only intention is to see everything in real time.

NB: upload order is important

M T W T F S S

DATE

① Game Implementation:

level class:

int level_parts: (How many images in 1 level)

list<Images> Images: (level images)

int no._of_ choices:

list<string> choices:

bool External_Animation:

Image Good_Anim: (for right answers)

Image bad_Anim: (for wrong answers)

bool default_Anim: (triggered if no Image Anim added)

list<level> levels: (Contains all existing levels)

M T W T F S S

DATE

① Main Menu: ✓

* SCripts:

① Manager: On an empty object (don't destroy)

* to load SCenes & Handle button actions

* Save no. of levels to be generated

② levels list: ✓

* Hirearcy:

① SCroll View

② level Prefab

* SCripts:

① level list population: on SCroll view

* Instantiate prefab and position it

② Instantiate level: on level prefab

* Make obj of level class and Send it to Manager (Populate an instance found in Manager Script)

③ level

* Hirearcy:

① UI

M T W T F S S

* Scripts:

① UI_Adoptation:

* Populate UI

* How to Implement:

① SOLID Principle:

② Event pattern: listening to an Event.
levelList Population listening to Instantiate Level

③ Event Action (observer Pattern): loosely Coupled "IObserver"
Instantiate level & Populate UI Sending level obj in onNotify

④ factory pattern: break UI_Adoptation according to each case:
one image/level or x choices/level animation/image .....

⑤ MVC Code pattern:

* Good Spliting & Naming

*Deadline: 3rd March, 2025 (2nd phase)

*Deleverables: Game Side Completed and Working
Designer mode documentation