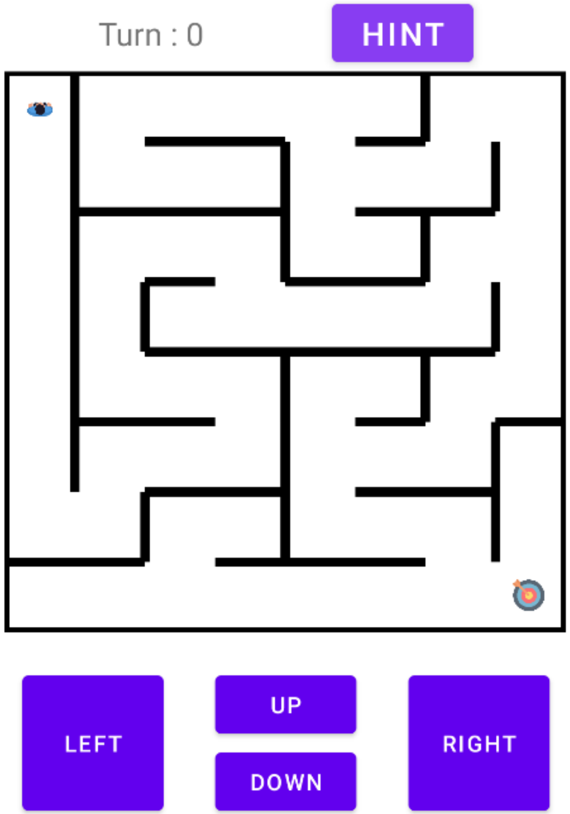
PA2
The Maze Runner

Mobile App Programming

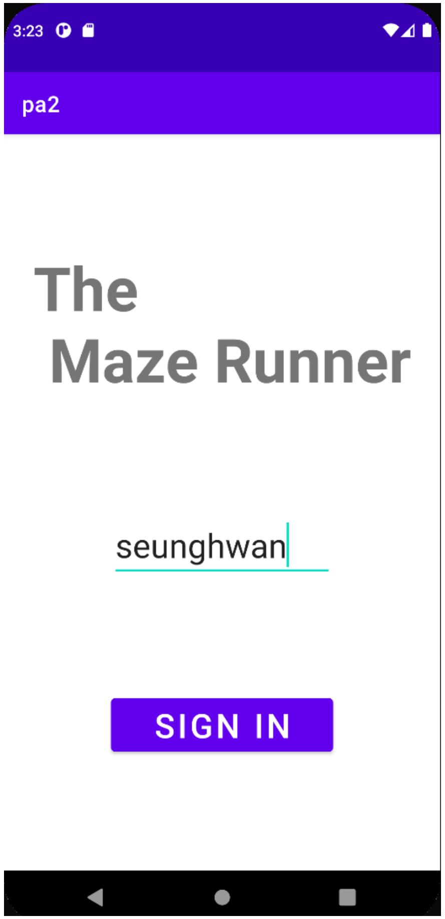
The Maze Runner

- Escape the maze using Arrow keys!
- There are 3 activities.
  - Sign in Activity
  - Maze Selection Activity
  - Maze Activity
- You should use List View, Grid View, HTTP Networking, and Explicit Intent.



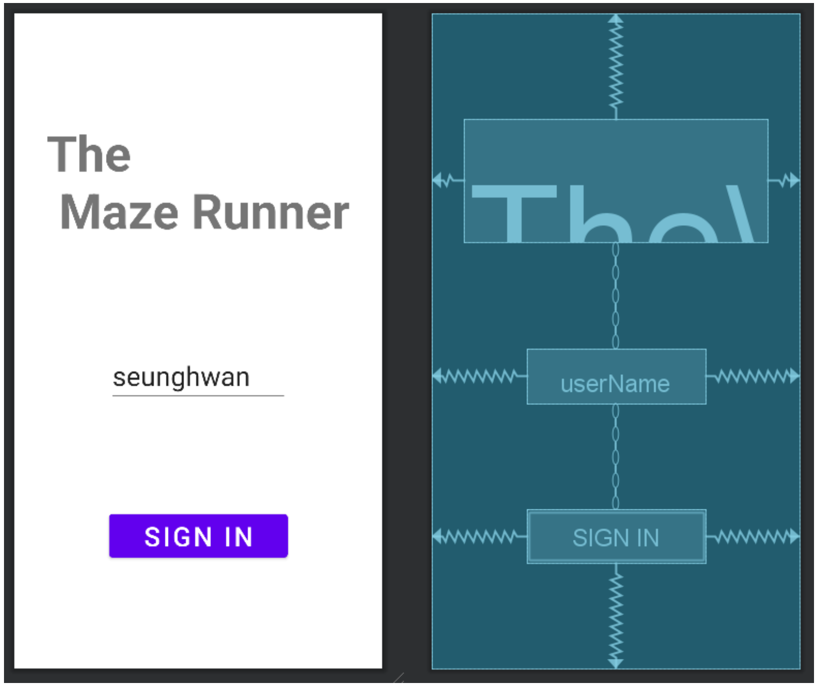
Activities – Sign in

- This activity check input user name is valid or not using networking with server.
- At first, Send POST request to server.
  - url : http://115.145.175.57:10099/users
  - Json Body : {"username" : <INPUT> }
- Response Json format is
  - {"success" : true} or {"success" : false}
- If input username exists in below list, it returns true.
  - ["seunghwan", "seongho", "seongmin", "mukoe"]
- If server returns true, go to map selection activity.
- If server returns false, show below toast shortly.
  - "Wrong User Name"



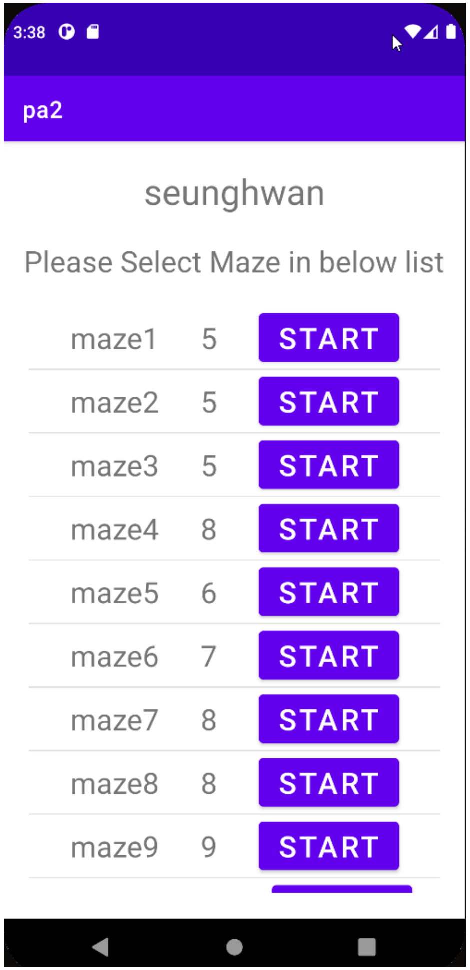
Layout – Sign in

- Title textview
  - Text size is “55sp”
  - Text Style is “bold”
  - Layout is “wrap_content”
- EditText
  - Text size is “30sp”
  - Layout width is “200dp”
  - Layout height is “wrap_content”
- Button
  - Text size is “30sp”
  - Layout width is “200dp”
  - Layout height is “wrap_content”



Activities – Map Selection Activity

- This activity show the possible maze list.
- Print username which user input sing in activity.
- At the start of activity, send GET request to server to get list of mazes.
  - url : http://115.145.175.57:10099/maps
  - No any query parameters
- Response Json format is Just list of Map<>.
  - EX) [{ "name": "maze1", "size": 5 },
    { "name": "maze2", "size": 5 },
    { "name": "maze3", "size": 5 },
    { "name": "maze4", "size": 8 },
    { "name": "maze5", "size": 6 },
    { "name": "maze6", "size": 7 }]
- To handle above Json format, read below document.
  - https://github.com/google/gson/blob/master/UserGuide.md#serializing-and-deserializing-generic-types



3:38

pa2

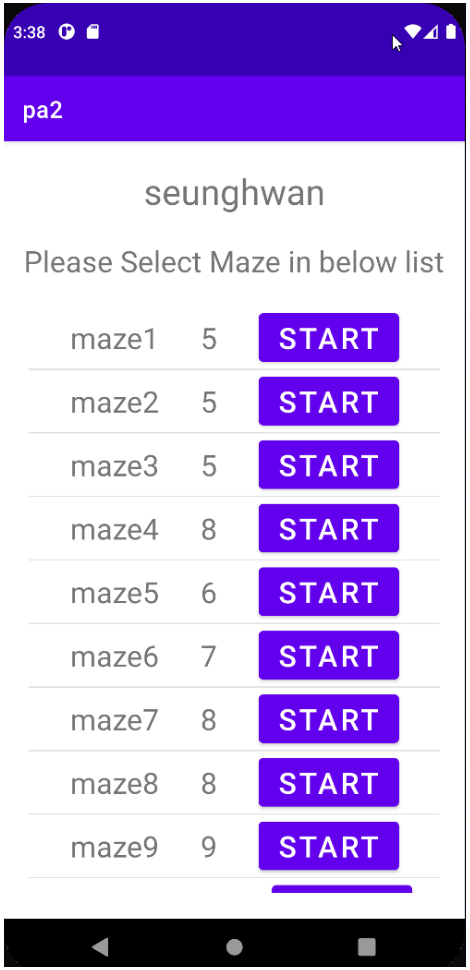
seunghwan

Please Select Maze in below list

maze1	5	START
maze2	5	START
maze3	5	START
maze4	8	START
maze5	6	START
maze6	7	START
maze7	8	START
maze8	8	START
maze9	9	START

Activities – Map Selection Activity

- You must print all mazes using scrollable Listview.
- Each item include Name(textview), Size(textview), and Start Button(Button)
- If you click Start Button, go to Maze Activity.
- BECAREFUL, response json list(maze list) length will be changed when we test your assignment!



3:38

pa2

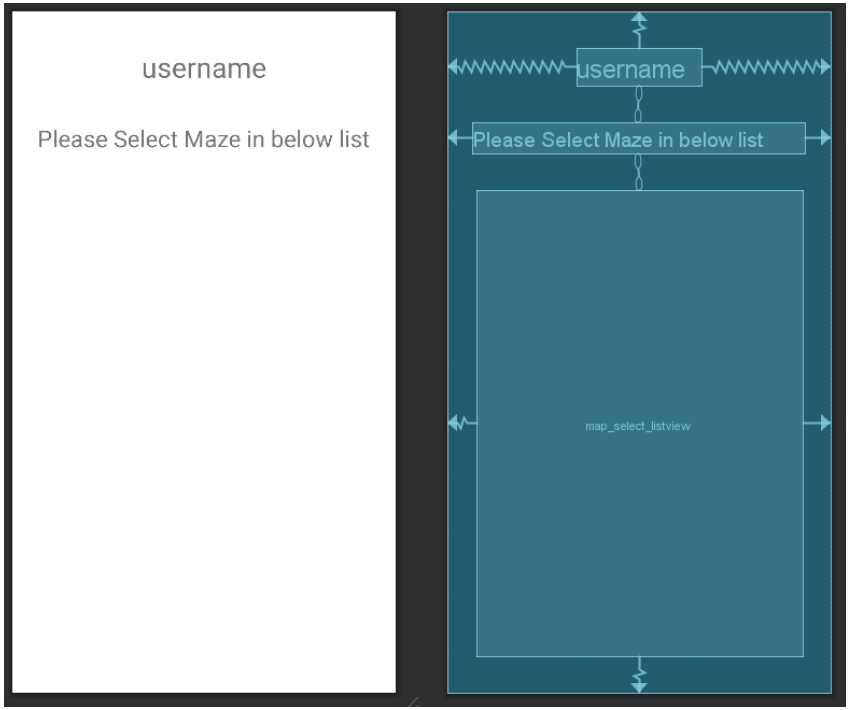
seunghwan

Please Select Maze in below list

maze1	5	START
maze2	5	START
maze3	5	START
maze4	8	START
maze5	6	START
maze6	7	START
maze7	8	START
maze8	8	START
maze9	9	START

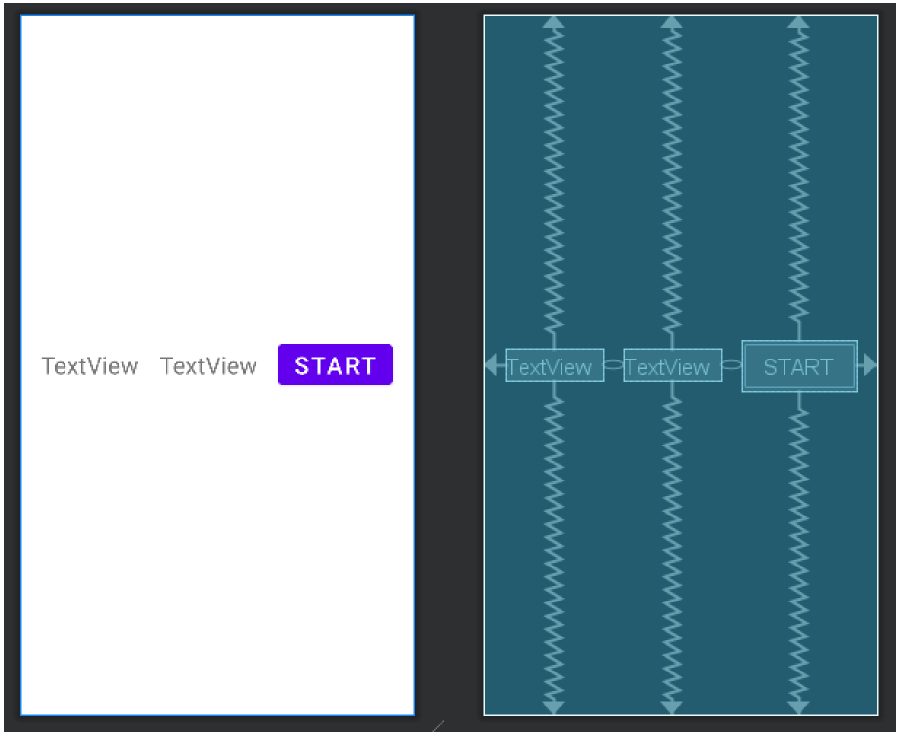
Layout – Map Selection Activity

- Username textView
  - Text size is “30sp”
  - Layout is “wrap_content”
- Explain textView
  - Text size is “25sp”
  - Layout is “wrap_content”
- Maze Listview
  - Layout width is “350dp”
  - Layout height is “500dp”


username
Please Select Maze in below list
username
Please Select Maze in below list
map_select_listview

Layout – Map Selection Activity

- ListView's element layout
  - Maze Name Textview
    - Text size is "25sp"
    - Layout is "wrap_content"
  - Maze Size Textview
    - Text size is "25sp"
    - Layout is "wrap_content"
  - Maze Start Button
    - Text size is "25sp"
    - Layout is "wrap_content"



Activities – Maze Activity

- This activity show the selected **square** maze.
- The # of col/row candidate is 5~10.
- Gridview width/height are always 350dp, then according to # of col/row size, each cell size will be changed
- At the start of activity, send GET request to server to get the maze shape.
  - url : http://115.145.175.57:10099/maze/map
  - Query Parameter : "name"

    (such as "maze1", "maze2")
- Response Json format is a String
  - EX) { "maze": "5□n

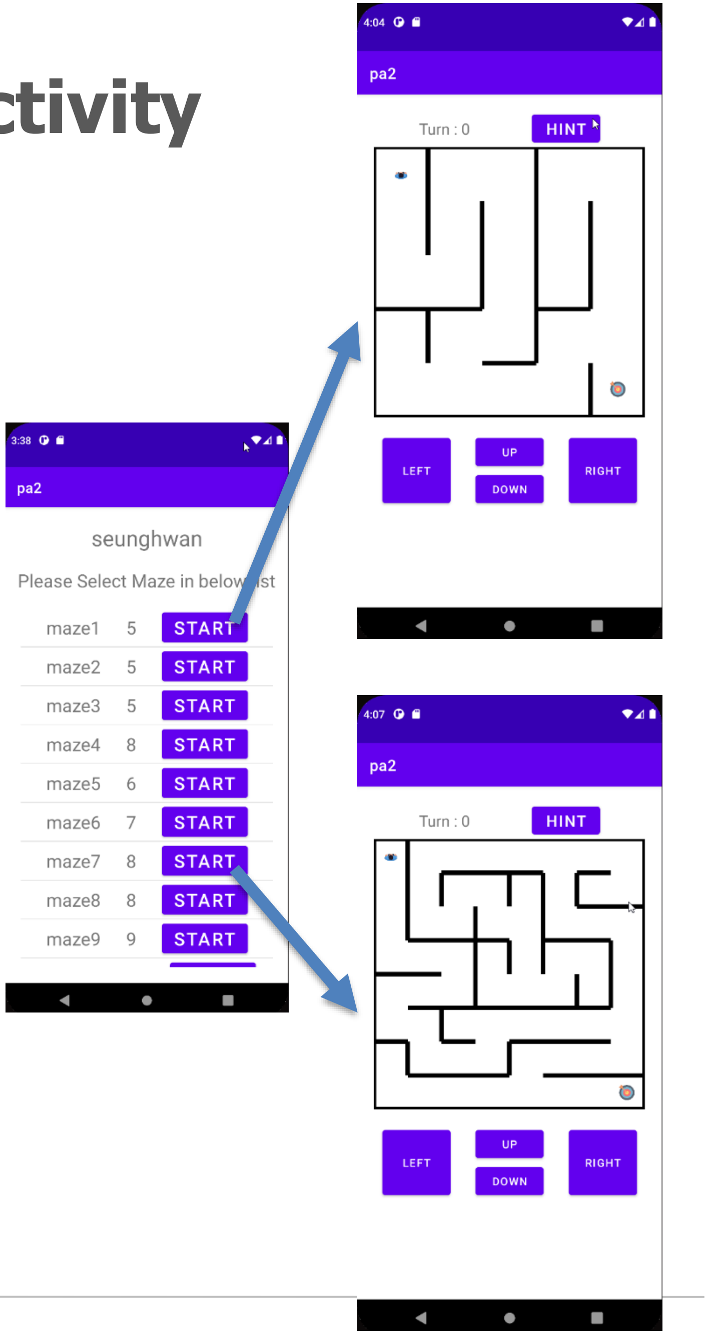
    14 10 10 9 13 □n

    12 8 9 6 1 □n

    5 5 4 11 5 □n

    5 5 7 12 3 □n

    7 6 10 2 11 □n"}



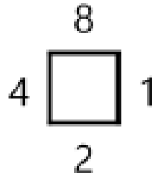
seunghwan

Please Select Maze in below list

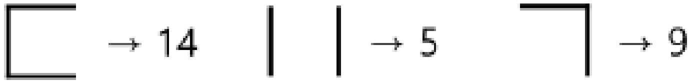
maze1	5	START
maze2	5	START
maze3	5	START
maze4	8	START
maze5	6	START
maze6	7	START
maze7	8	START
maze8	8	START
maze9	9	START

Activities – Maze Activity

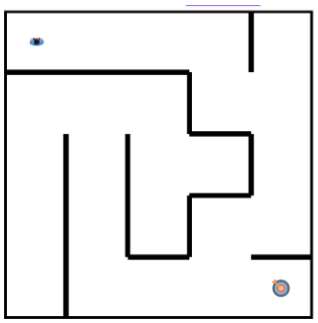
- Response means
  - First Integer means size(# of row and column)
  - Next, Get **size*size** number of integers.
  - Each integer means 1 maze cell.
  - Top wall is 8, Left wall is 4, bottom wall is 2, right wall is 1.
  - If value is 14, which is 8 + 4 + 2, it means the cell has up, left, and bottom walls.



(8+4+2) (4 + 1) (8 + 1)

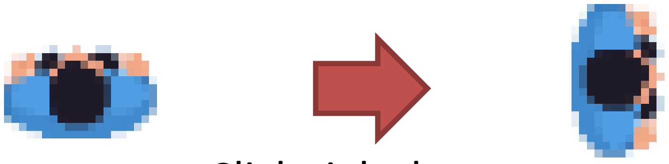


```
5
14 10 10 9 13
12 8 9 6 1
5 5 4 11 5
5 5 7 12 3
7 6 10 2 11
```



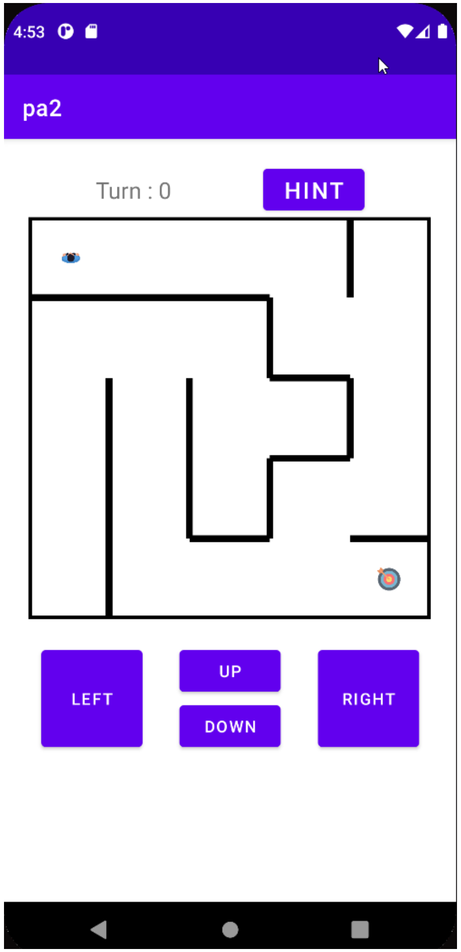
Activities – Maze Activity

- Maze
  - Start Point is always (0,0)
  - Goal Point is always (size, size)
  - If you click (up, down, left, right) button, user character will move each direction.
  - After move, user character must look in the move direction.



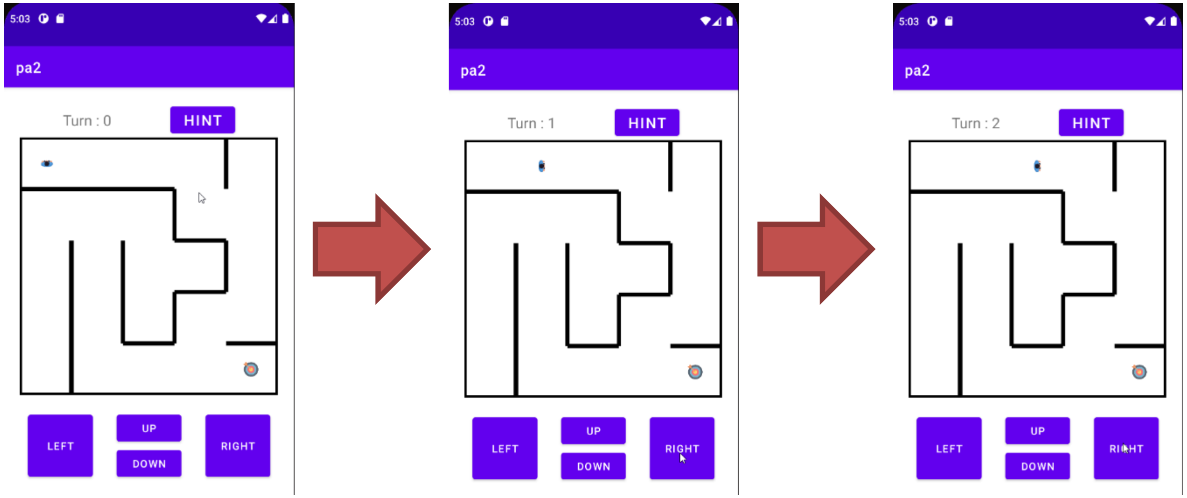
Click right button

  - User character can't pass through the wall.



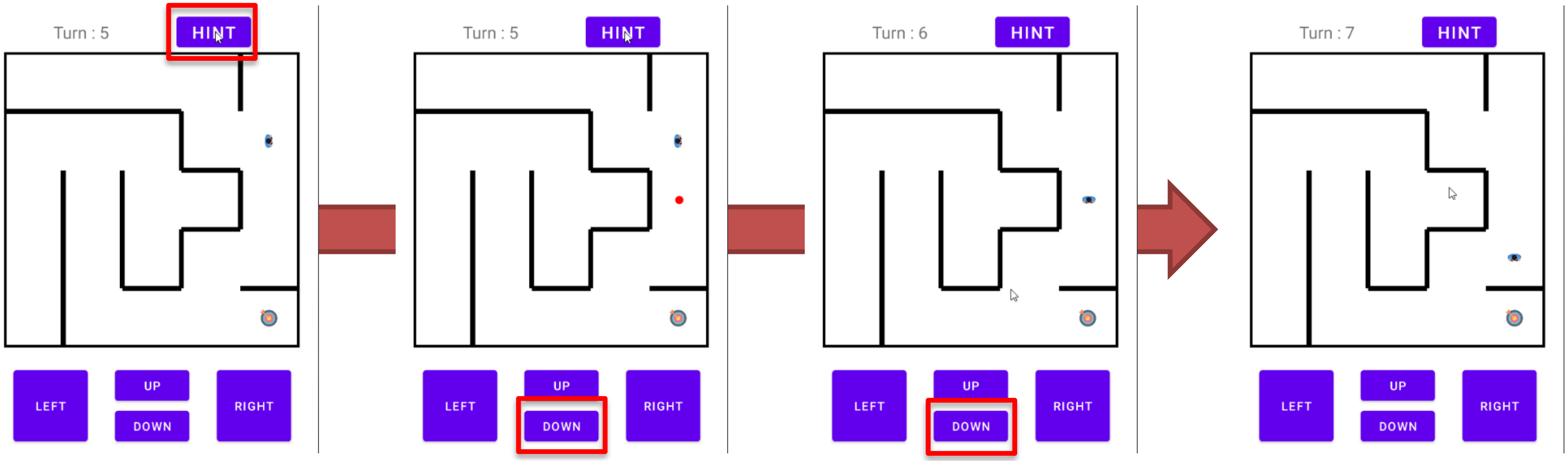
Activities – Maze Activity

- At the top of the layout, there are Turn textview and hint button.
  - Turn textView
    - Every movement, increase the number(turn).
    - **Don’t increase** turn number if the character can’t move because of the wall.



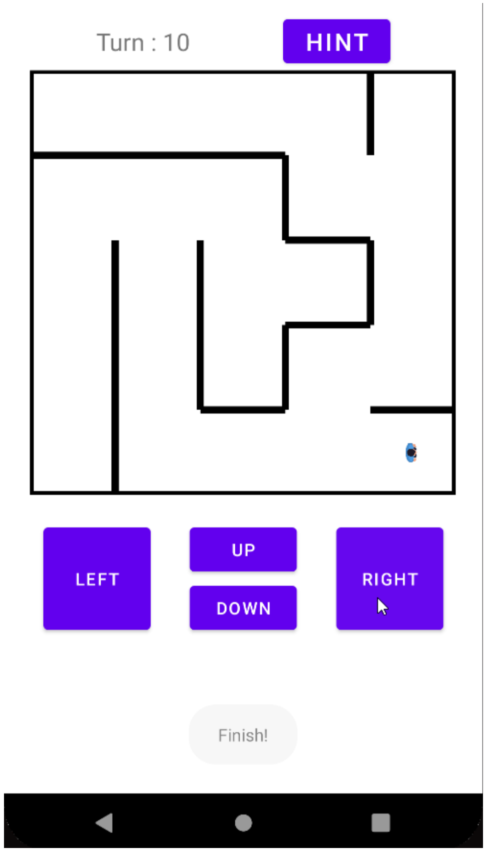
Activities – Maze Activity

- Hint
  - If you click hint button, calculate the shortest path to goal point, and make a dot at the next position in the path.
  - When you pass that position, the red dot disappears.
  - Hint button works only **1 time** at each challenge.



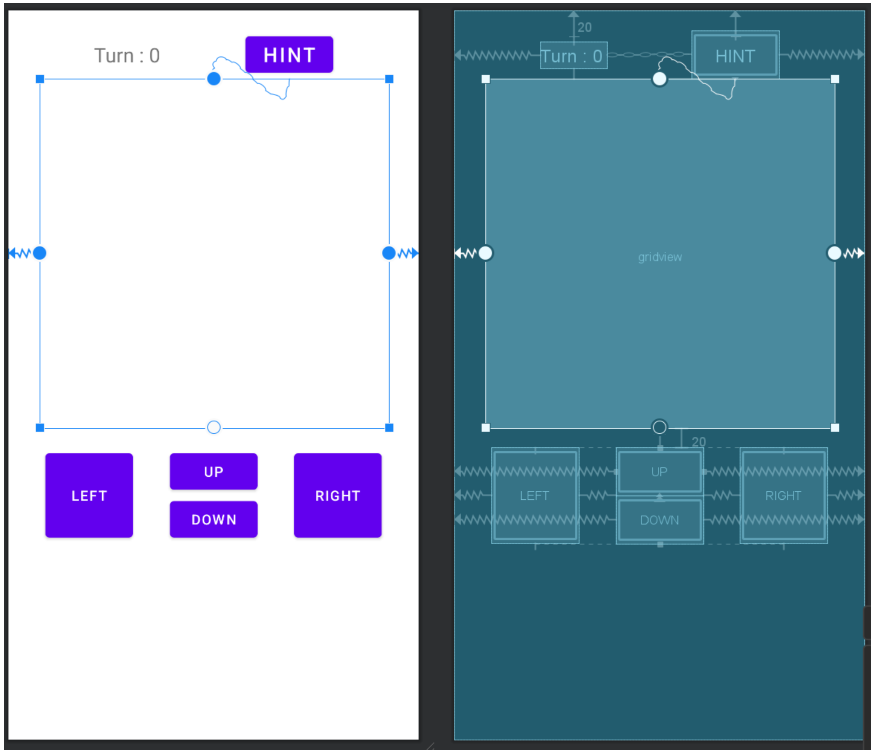
Activities – Maze Activity

- Arrive at goal point
  - If character arrives at goal position, show toast message, shortly.
    - "Finish!"



Layout – Maze Activity

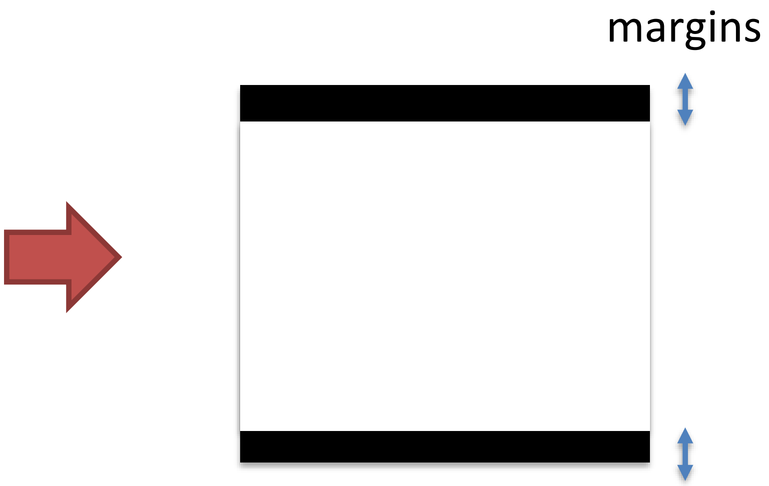
- Turn Textview and Hint button
  - Textsize is "20sp"
  - Margin Top is "20dp"
  - Layout is "wrap_content"
- GridView
  - Layout is both "350dp"
- Arrow buttons
  - Margin Top is "20dp"
  - Set left/right buttons top to top of up button
  - Set left/right buttons bottom to bottom of down button
  - Textsize is default



Layout - Maze Gridview

- Recommendation how to represent maze
  - Students must print maze **clearly**
  - We recommend to use Gridview (Very similar with Listview)
    - https://developer.android.com/reference/android/widget/GridView
  - When you generate a cell(item), input 2 image view in layout file.

```
<constraint layout
        android:background=“@color/black” >
<image view
        android:background=“@color/white”
        marginTop=“3dp“
        marginBottom = “3dp“  />
<image view
        layout_width = “30dp”
        layout_height = “30dp”
/>
</constraint layout>
```

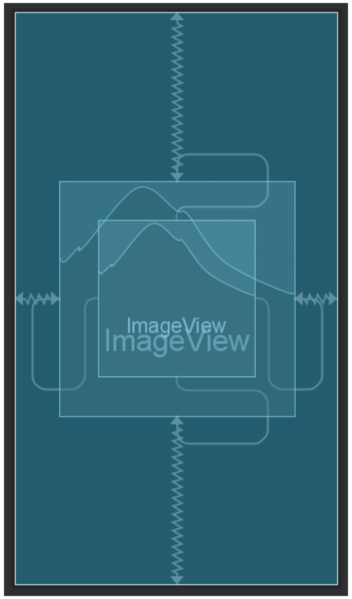


Layout - Maze Gridview

- Therefore, you should set each cells width, height and margins in .java file programmatically.

=> (May this is little bit difficult!! Good LUCK!)

- At this time, you must consider **dpi,** because programmatical setup only use **px**.
- To put image (user character, goal, hint) over the cell, make new image view. Next, set layout likes below
  - Bottom to bottom of {cell image view}
  - Top to Top of {cell image view}
  - End to End of {cell image view}
  - Start to Start of {cell image view}
- User Character, goal, hint image's size is both "30dp".



Networking - summary

- TA run personal server to handle your http request.
  - Don't send to many requests. (The server can ban your ip)
- Send to http://115.145.175.57:10099/
- If server is dead, please send ICAMPUS message to 정승환TA or leave a question on google Q&A sheet.
- I will upload server code (python Flask) and explain how to run it, then **if server is dead, please make local server and use it.**

Networking - summary

1) POST http://115.145.175.57:10099/users

-> Input : Put json object which has “username” key

-> Return : Return json object which has “success” with Boolean value

2) GET http://115.145.175.57:10099/maps

-> input : Just send GET http without any query parameter

-> Return : Return json object with Array of Dictionary(or Map)

3) GET http://115.145.175.57:10099/maze/map

-> Input : Send GET http with “name” query parameter

Ex) http://115.145.175.57:10099/maze/map?name=maze1

-> Return : Return json object which has “maze” with String value

PA2 Specification

- Project Settings
  - Minimum SDK: Must be 26(Android 8.0)
  - Target & Compile SDK: Must be 31(Android 12.0)
  - Application ID(Package): Must be edu.skku.cs.pa2
- Application Execution
  - The application must be started in 10 seconds.
  - UI must not stop more than 5 seconds.
  - No error while build/execution

PA2 Criteria

- If your application works exactly same to example video (in ICAMPUS), you will get full points.

- Sign In Activity – 20 points
- Map Select Activity – 30 points
- Maze Activity – 50 points
- **Each error -> -5 points**
  - Ex) 미로의 벽이 이상하게 출력됨 -> 5점 감점
  - Ex) 벽에 막혀서 이동하지 못했는데 turn이 증가 -> 5점 감점
  - Ex) 버튼을 눌렀는데 10초 이상 로딩이 걸림 -> 5점 감점

PA2 Others

- Do not care about
  - # of Adapter class, Adapter object, class
- Questions?
  - https://docs.google.com/spreadsheets/d/1gJ1yWfpirsS-7F8Nj-s-HpHZPFF1g6yHbWIU_KBkCRw/edit?usp=sharing
  - We will not receive a question about error or how to implement the Application
  - Ex) “미로의 크기가 각각 다르나요?”
    - A. 네, 다를 수 있습니다.
  - Ex) “미로를 GridView로 구현해야하나요?”
    - A. 아니오, 자유롭게 하시면 됩니다. 저희는 GridView를 권장드립니다.
  - Ex) “미로가 이상하게 출력이 됩니다. 왜 그런가요?”
    - A. 답변해드릴 수 없습니다.

PA2 Submission

- Submit single zip file with name “<Student ID>_pa2.zip”
  - Export to zip -> Change file name
  - Do not care about ending ‘-<Number>’ (ex: 2022524288-1.zip)
- Submission Due
  - 5/1 23:59
  - Delayed Submission
    - ~5/4 23:59
    - Your score will be penalized by 25%p per day.
      - 70/100, 2 day late = 70*(1-0.25*2) = 35/100