THE UNIVERSITY OF SYDNEY

INFO1113 / COMP9003 Assignment

Due: 20 October 2024, 11:59PM AEST

This assignment is worth 20% of your final grade.

Task De

In this assig essing library
for graphics nd the screen
and the play eflected off a
player-draw espawns.
Once all bal

You have be meplay
mechanics a osted it on
your online

You are enc e of the
specificatio nment for
you. As with ode or
solutions wi

Workin

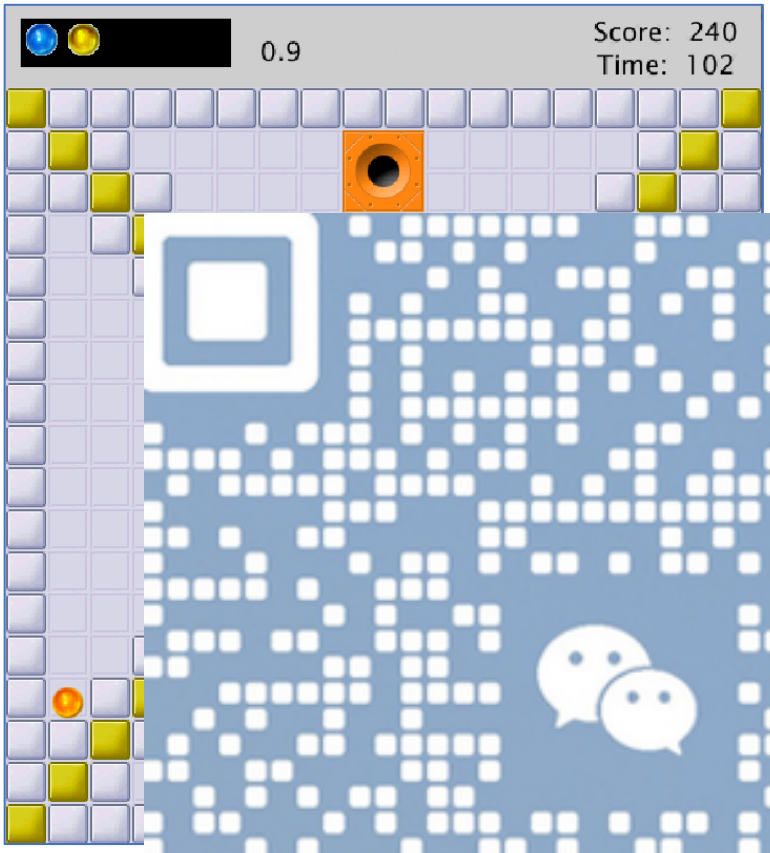
You have be download
the scaffold dencies. You
will be using draw
graphics. Yo

Gameplay

The game contains a number of entities that will need to be implemented within your application.

Level

Each level is read from a text file of characters 18x18. The size of the window should be 576x640, meaning each character in the file corresponds to 32x32 pixels.



The level layouts are defined in files provided in the "layout" attribute of the JSON configuration file described below. Each level must have an associated layout file.

need to
characters. It
hey are not
at the end of
ne bottom). In
am should
balls off the

ole level

There are 5

- X – me board
  doe reen.
- 1,2, ball hits one
  of tl
- S – available
  spawners in the current level, each time a ball is ready to be spawned).
- H – Holes. The hole takes up 4 tiles, where the 'H' character is the one in the top left. The number in the character to the right of the H is the colour of the hole.
- B – Balls. Instead of spawning after the spawn interval, a ball may be present immediately from the level beginning, at a specific place on the board. The colour of the ball is denoted by the character to the right of the 'B'.
- Spaces – empty space, just ignore it (blank tile).

```
{
  "levels": [
    {
      "layout": "level1.txt",
      "time": 120,
      "spawn_interval": 10,
      "score_increase_from_hole_capture_modif
      "score_decrease_from_wrong_hole_modifie
      "balls": ["orange", "blue", "orange", "
    },
    {
      "layout": "level2.txt",
      "time": 180,
      "spawn_interval": 15,
      "score_increase_from_hole_capture_modif
      "score_decrease_from_wrong_hole_modifie
      "balls": ["green", "grey", "grey", "blu
    }
  ],
  "s

  },
  "s

  }
}
```

Config

The config file is in located in config.json in the root directory of the project (the same directory as build.gradle and the src folder). Use a json library to read it. Sample config and level files are provided in the scaffold.

The map layout files will also be located in the root directory of the project. However, sprites or images such as the `ballx.png`, and `wallx.png` will be older all/) or kball/).

For each lev

- `lay` the grid for wall
- `tim` eded, the play e value like -1) or n
- `spa`
- `sco` sed for each ball
- `sco` lues gained on this level by this modifier.
- `score_decrease_from_wrong_hole:` The amount of units score is decreased for each ball type when they enter the wrong hole.
- `score_decrease_from_wrong_hole_modifier:` Multiply the score values lost (when a ball enters a wrong hole) on this level by this modifier.

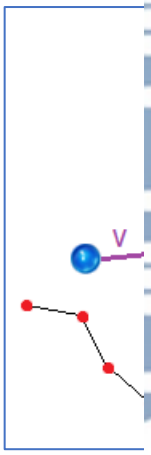
Balls 

Balls may appear in the level layout file, as “B0”, “B1”, “B2”, etc in which case they are spawned immediately in that location when the level begins. Alternatively, they may also be specified in the configuration file, which will cause them to be spawned at a spawner  throughout the duration of the game. The frequency of spawning is determined by the `spawn_interval` configuration property of that level, which determines how many seconds in between when balls spawn. From being initially at that given value * App.FPS, it counts down on each frame and is displayed in the top bar, next to the display of where balls yet to be spawned appear. The order of balls in this display should be the same as the configuration file (only the next 5 balls yet to be spawned are shown). When the spawn interval counter reaches 0, the next ball is spawned in the game. All other balls remaining yet to be spawned, will gradually move to the left in the display at a rate of 1 pixel per frame.

When balls spawn in the game, they have a random velocity vector which is either -2, or 2 pixels in the x direction, ar halved). Throughout x direction and `j` is the change their velocity vec

Hitbox

A hitbox is a -drawn line may appear



ints ine vector v the line, 2 are the shown in

The steps to

1. Det nts). To do this points, P1 and P2. The distance formula is as below:

$$distance(A, B) = \sqrt{(B_x - A_x)^2 + (B_y - A_y)^2}$$

If distance(P1, ball+v) + distance(P2, ball+v) < distance(P1,P2) + ball’s radius, then the ball is considered to be colliding with the line segment connecting P1 and P2.

2. Calculate the normal vectors of this line segment, N1 and N2 from P1(x1,y1) and P2(x2,y2). If we define dx = x2 - x1 and dy = y2 - y1, then the normals are (-dy, dx) and (dy, -dx).[1]

[1] Source: https://stackoverflow.com/questions/1243614/how-do-i-calculate-the-normal-vector-of-a-line-segment

3. Normalise the normal vectors so that their magnitude is 1. (ie, divide by $\sqrt{i^2 + j^2}$).

4. Find the normal vector on the side of the line that we want to use, either N1 or N2. The one that should be used is the one which is closer to the ball. To do this, perhaps check the distance of the midpoint of the line segment + normal vector with the ball's position (these are the blue points shown in the diagram), and choose the vector which results in a closer distance.

5. Calculate the new trajectory of the ball. This is given by the following formula: [2]

$$\mathrm{u} = \mathrm{v} - 2(\mathrm{v} \cdot \mathrm{n})\mathrm{n}$$

Where $\mathrm{v} \cdot \mathrm{n}$ is the dot product, and n must be normalised – the normal vector of the line segment.

Hitboxes for player drawn lines are rendered on the game board with a black line of thickness 10 units. Once a colli

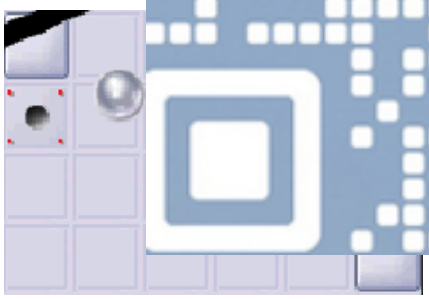
Walls

A wall is a ti ng for walls will work th nge, blue, green or yel

Even if the g ne edges.

Holes

Holes take u ntre of a hole (from the ce nally to how close it com aptured by the hole anc e ball to the centre of th



n in hole. n ion) of the ball to the hole, to the ball's velocity on each frame.

If the hole colour matches the ball's colour (or it's a grey ball, or grey hole), it is a success and the score increases by the amount given in the configuration file, multiplied by the level multiplier. Grey balls are allowed to enter any holes, and balls of any colour can enter a grey hole to count as a success.

If the colour capture was not successful, the ball rejoins the queue of balls yet to be spawned, and score will instead decrease by the amount specified in the configuration file.

[2] Source: https://math.stackexchange.com/questions/13261/how-to-get-a-reflection-vector