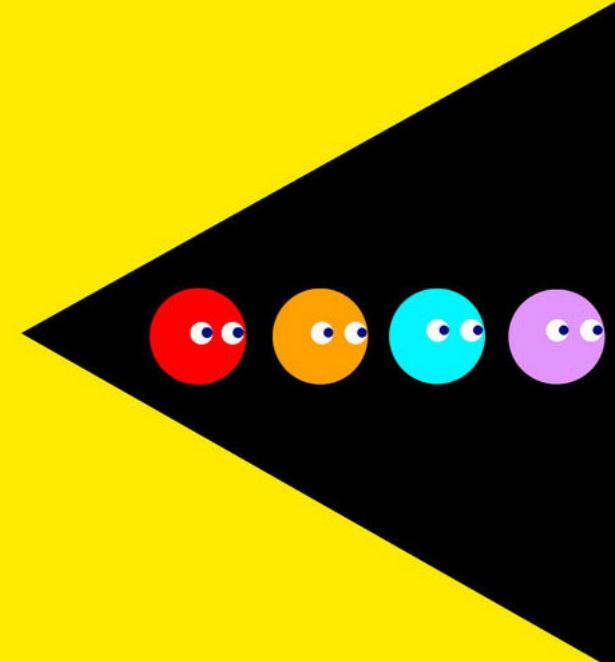


System.out.println ("PacmanGoHome");

```
/**  
 * @author: Linh Tang, Yolanda Jiang  
 * @course: CSC207 - Spring 2020  
 * @school: Grinnell College  
 */
```



Introduction

PACMAN

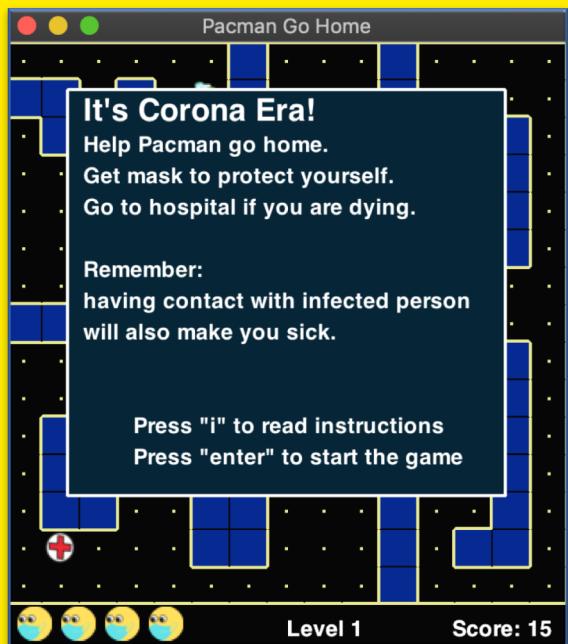
- Classic 2D game that is easy to understand and play
- Easy to implement with Java Swing
- Easy to build desktop GUI for cross-platform systems
- Support the messages through our self-designed graphics

GO HOME

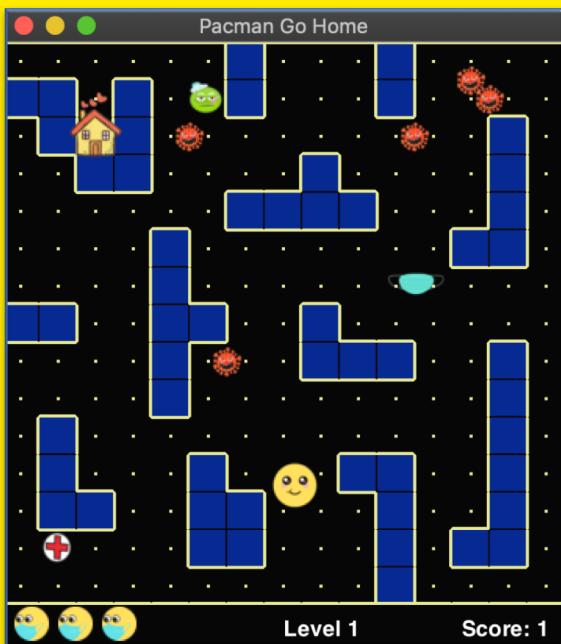
- This game is built in the middle of the pandemic with an educational purpose!
- Home is the safest place for you
- Once you are out, you will easily and quickly run into viruses and infected people.
- Wearing masks saves your life
- Hospital saves your life.

Go home, as soon as you can.
Stay home, as long as you can.

Overview



Start screen



In Game



Moving



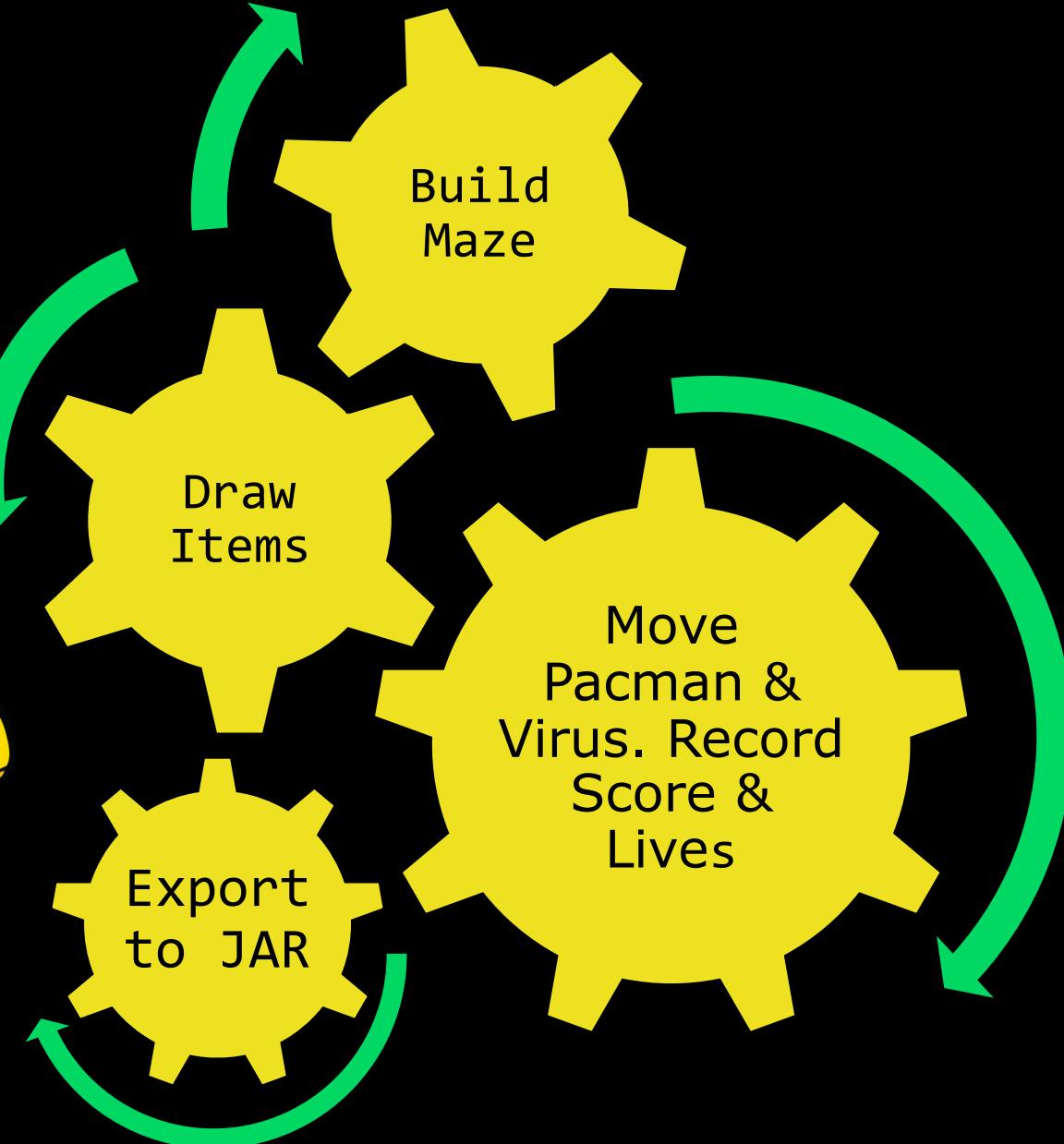
Game Over



DEVELOPMENT
FLOW

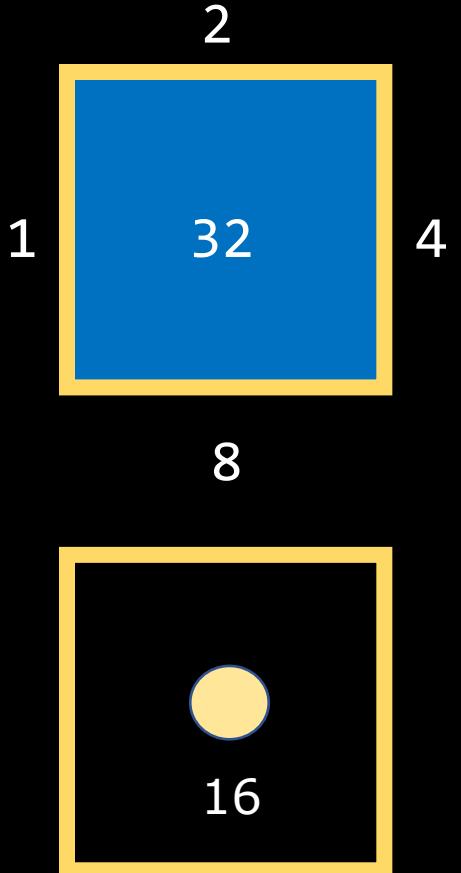


PacmanGoHome



Implementation - Maze

- The interface is a 2D plane of size 360x410.
- The maze size is 360x360, each axis consists of 15 squares, each square size is 24.
- Maze data is stored as an array of 255 16-bit integers. We construct the maze by using **bitwise operation (&)** to detect whether that square has left/right/up/down wall, has a point or in a block.
- 1 = left, 2 = top, 4 = right, 8 = bottom, 16 = block, 32 = point. These numbers can be added, for example if square = 35, it has top, left borders and there is a point inside ($32 + 1 + 2$).



Implementation - Items

- For each item, we keep their (x,y) coordinates as a property of the game class to put in the maze and determine collisions later.
- This is how we built the maze and choosing the appropriate square to put our items in. The maze is static at all levels.
- The **Pacman** always starts at the **yellow** position.
- **Viruses** are evenly distributed at 4 starting points colored in **red**.
- Hospital is grey, mask is blue, sick person is green and home is purple

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	43	42	34	42	34	38	18	35	34	38	18	35	34	34	38
2	16	16	37	16	33	36	16	33	32	36	16	33	32	40	36
3	39	16	45	16	33	32	34	32	40	32	34	32	36	16	37
4	33	38	16	16	33	32	40	44	16	41	32	32	36	16	37
5	33	32	34	34	40	36	16	16	16	16	33	32	44	16	37
6	33	32	32	36	16	33	34	34	34	34	32	36	16	16	37
7	41	40	32	36	16	41	32	32	40	32	32	32	34	34	36
8	16	16	33	36	16	16	33	36	16	41	40	32	32	40	36
9	35	34	32	36	16	35	32	36	16	16	16	33	36	16	37
10	33	40	32	36	16	33	32	32	34	34	34	32	36	16	37
11	37	16	33	32	34	40	32	32	40	40	40	32	36	16	37
12	37	16	41	32	36	16	41	32	36	16	16	33	36	16	37
13	37	16	16	33	36	16	16	33	32	38	16	33	44	16	37
14	33	34	34	32	36	16	16	33	32	36	16	37	16	16	37
15	41	40	40	40	40	42	42	40	40	44	24	41	42	42	44

pacman 
hospital 
mask 

home 
virus 
sick 

Implementation - Score & Collision

- If Pacman moves to a square with a point (bean), score is incremented and the point disappears (being eaten).
- Score reflects how well the users play the game, but not the essential to win. As Pacman gets home, the current level is finished and score is showed.
- Score and maze are then reset to default at the beginning of each level.
- Collision is determine if 2 objects coordinates are equal with an offset = 24 in any directions, i.e. they met at the same square.
- As with the beans, hospital and mask are one-time-use item, but sick Pacman always stands here no matter how many times Pacman bump into it. Viruses can avoid any item on their ways, except for Pacman. Viruses can't enter the house.

Implementation - Virus & Pacman

- **Moving algorithm:**

- Find the current position, i.e. the square in the maze that they are in
- Use bitwise operation (&) to determine if there exists barrier (wall) in any side of that square.
- Pacman can't change direction and move forward if there is barrier. Pacman moves at constant speed.
- Viruses randomly choose one of available directions to advance. If they are in a tunnel (2 opposite walls are blocked), they keep heading forward until they get out. Viruses are randomly assigned different valid speeds (not relatively too fast/slow compared to Pacman).

- **Viruses:**

- Coordinates of N viruses are stored in 2 arrays of N integers `virus_x`, `virus_y`.
- Current directions that viruses are moving are also kept in 2 arrays `virus_dx`, `virus_dy`.
- Speeds of viruses are also kept in an array
- Each virus stores its current states by indexing those arrays above.
- Number of viruses are different in each level.

- **Pacman:**

- Coordinates are stored in 2 integers `pacman_x`, `pacman_y`.
- Delta changes in each of its moving direction are kept in `pacman_dx`, `pacman_dy`.
- Current directions that pacman is moving are also kept in `req_dx`, `req_dy`.

Game Design

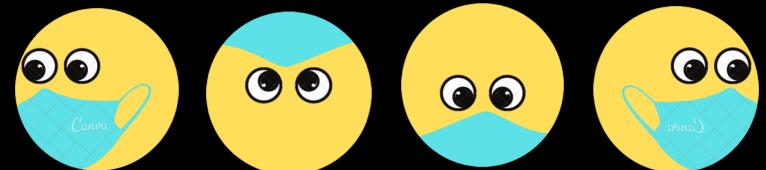
- Goal: Get Pacman Home

- User Inputs:

- 4 cursor keys to control movement of Pacman
- Use SPACE, ESP, ENTER, I to control the game (pause/resume, quit, start, instruction,...)

- Rules:

1. Pacman initially has 3 lives. If Pacman has no life left, the game is over.
2. Every time Pacman encounters viruses or infected person, it loses 1 life.
3. Viruses move around randomly with different speeds, sick person stands still.
4. If Pacman gets the mask or go to the hospital, it gains 1 extra life
5. There are 2 levels of the game: level 1 has 5 viruses, level 2 has 10 viruses.



Fours states of Pacman



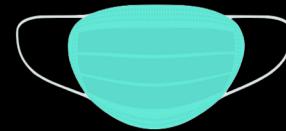
Sick



Virus



Pacman



Mask



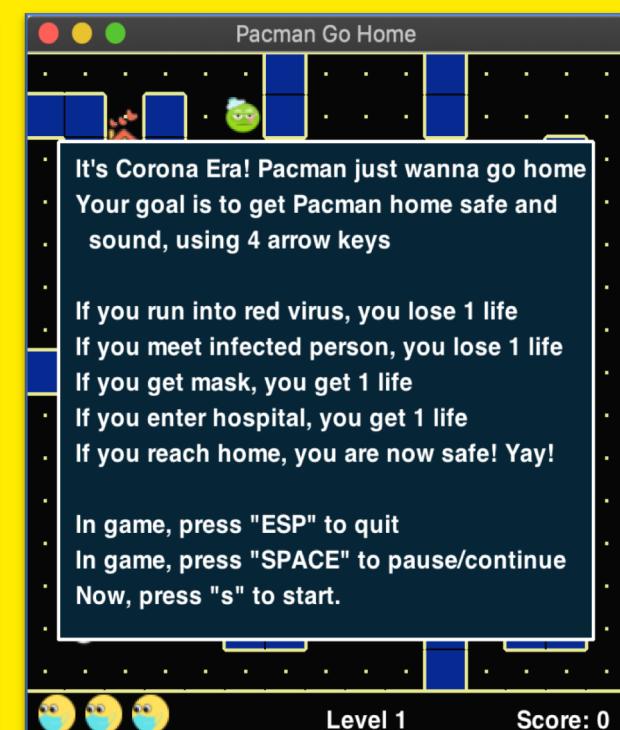
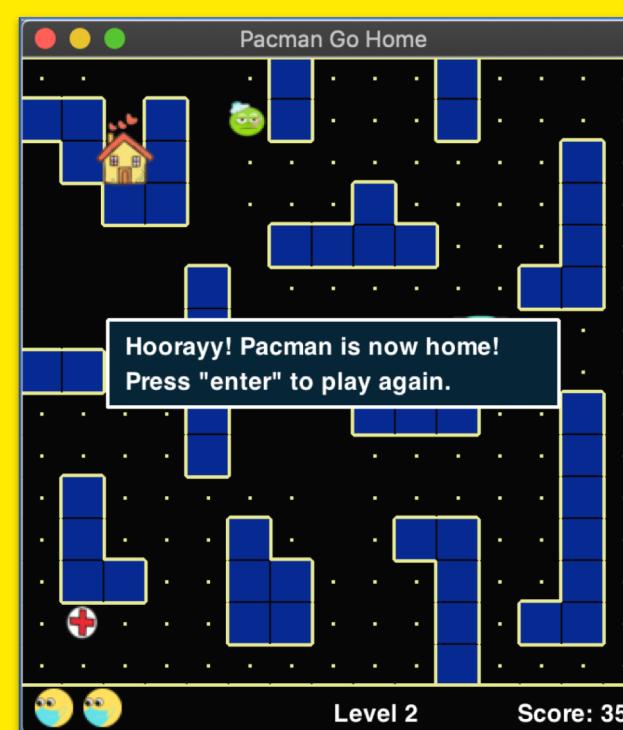
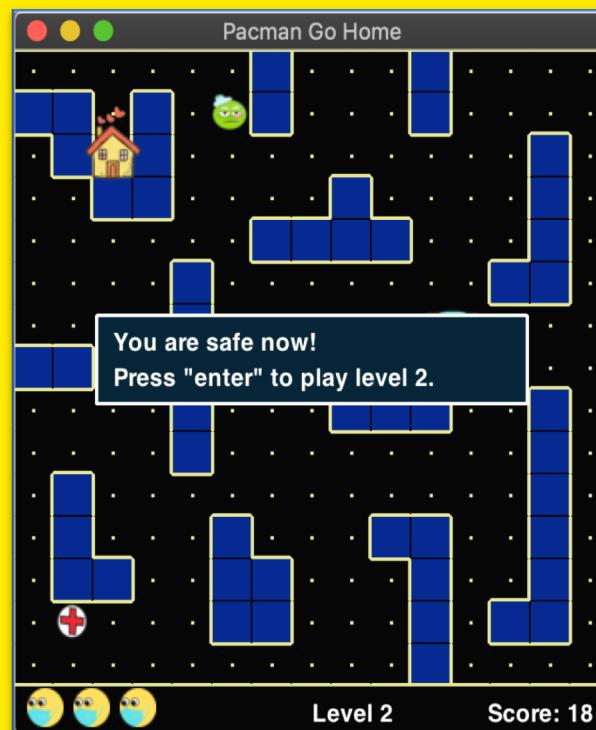
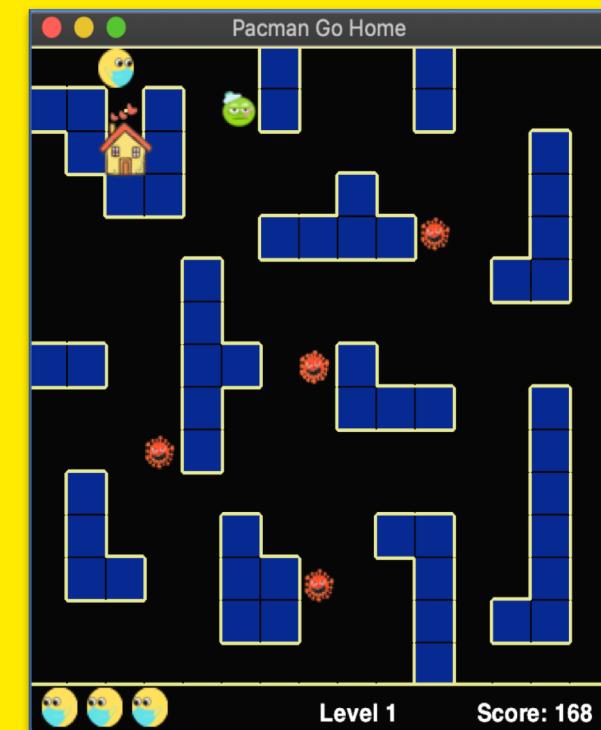
Hospital



Home

Product

- Video demo of the game



Ate all beans

Level Up

Win

Instructions



<https://bit.ly/PacmanGoHome>

Download link, available for MacOS and Windows with JDK 12+

In-progress Developments:

- ✓ Make the game window resizable on multiple-size desktops
- ✓ Create Linux and improve Window version of the game app
- ✓ Incorporate more sound effects into game (currently just music)
- ✓ Build high score boards that save user's playing history
- ✓ Update higher levels of the game
- ✓ Make the game more challenging with more messages delivered in game, such as when meet infected people, etc..

References

- Original Pacman game tutorial: <http://zetcode.com/tutorials/javagametutorial/pacman/>
- Music: <https://www.classicgaming.cc/classics/pac-man/sounds>
- Images (viruses, hospital, mask): multiple sources from Internet