

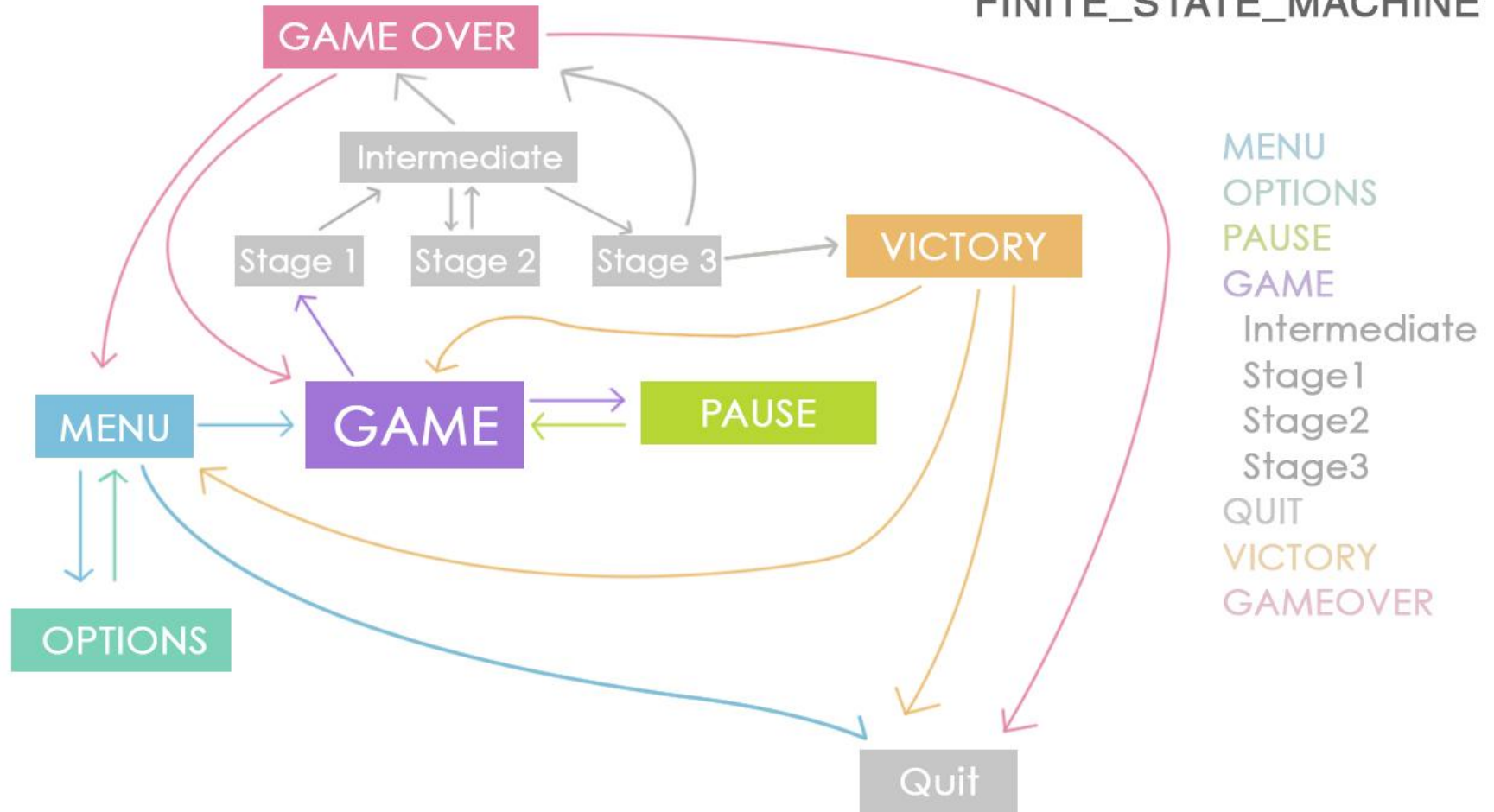
# Metal Sphere

EE552 Video Game Project

Lucille Morcos  
Zhen Pei  
Zhihuan Zhang  
Xiaoshi Xu



# FINITE\_STATE\_MACHINE



# Class Breakdown

31 Class in 10 packages

8 most crucial class:

**WARNING:**  
**No explicit codes will be explained**

1. Game
2. ResourceLoader
3. Controller
4. CoreObject
5. StageGenerator
6. Renderer
7. Input(KeyInput & MouseInput)
8. Screen Package

# Class: Game

## This is where everything comes from

Extends Canvas & Implements Runnable

### Most Crucial Functions:

1. **init()**: initialize everything.(load resource from hard drive,instantiate other related class, setup key/mouse input,initialize variables.)
2. **run()**: the main loop of the game, call the init() function, calculate the FPS and tick rate. Setup ending condition and transition between different stages.
- 3.**start()**: call under the main function. It will create a thread and call the run() function to run the game.

The layout of the window will be set up in the main function (includes the icon and cursor)

Window layout (title, size, etc.)

# How it works?

main()——>start()——>thread.start()——>run()——>init()  
——>main loop(while(isRunning))

Within the main while loop():

render(): a function to call other corresponding function in different objects to draw themselves

tick(): a function to call every object to update the status of themselves'

# Class: ResourceLoader

Before the program is able to function, every needed resource will be loaded from hard drive to memory

The thing we need: **Fonts, Sound effect, Music, Images**

**Class Font:**

```
ArrayList<Fonts> fontList=new ArrayList<Fonts>();
```

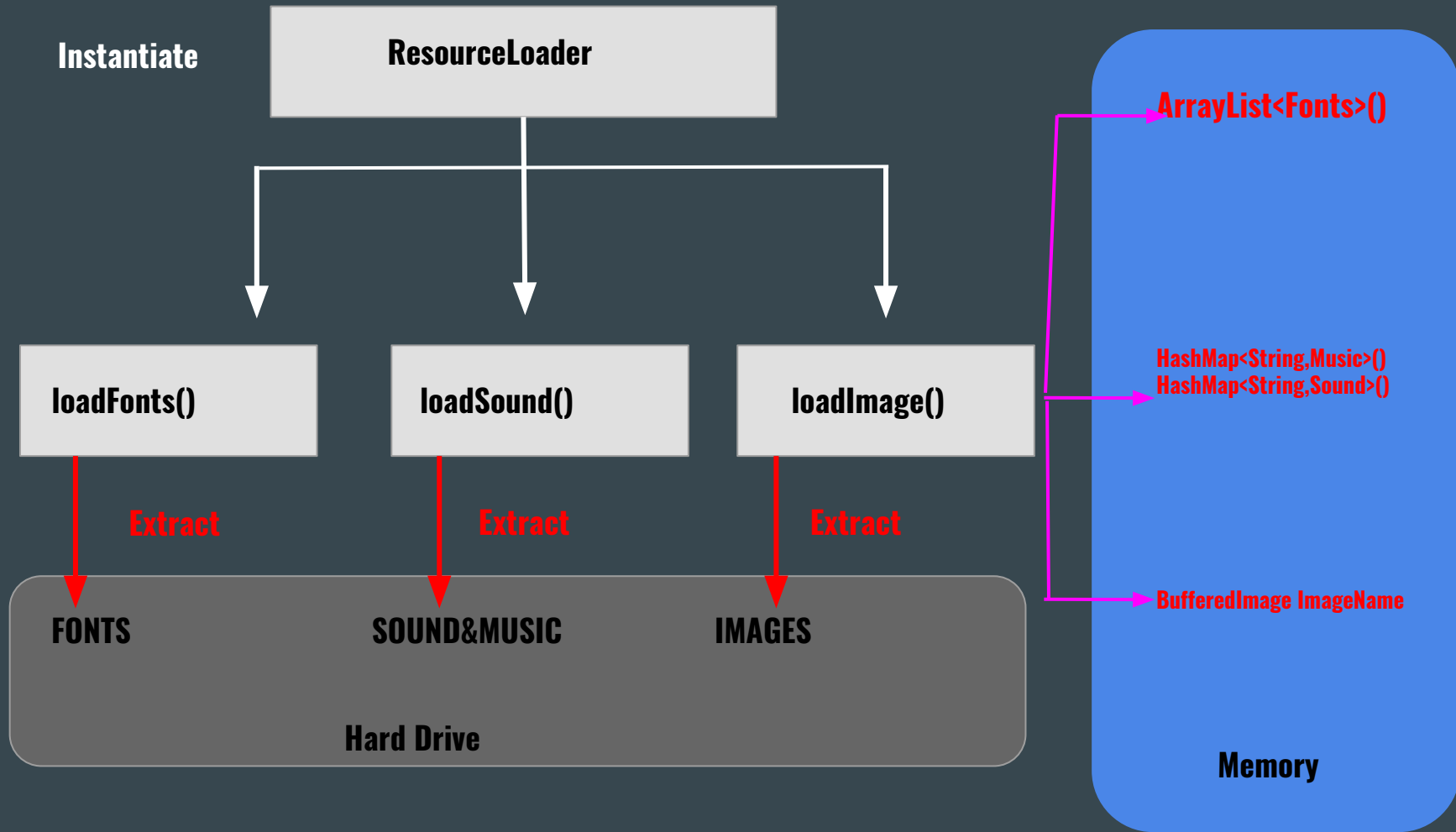
**Class AudioPlayer:**

```
Map<String,Music> musicMap =new HashMap<String,Music>();
```

```
Map<String,Sound> soundMap =new HashMap<String,Sound>();
```

**Class Image: Variables:**

```
e.g.      public static BufferedImage ImageName;
```





# Class: CoreObject

**Void tick()**

**Void render(Graphics g)**

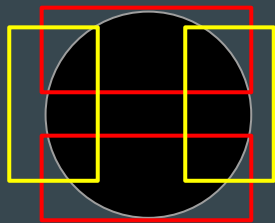
**Public Rectangle getTopBounds(){**

**...  
}**

**-getBottomBounds()**

**-getRightBounds()**

**-getLeftBounds()**



Left & Right bounds

Top & Bottom bounds

# Class: Controller

**-addObject()**

**-removeAll()**

```
ArrayList<CoreObjects> objects = new ArrayList ...
```

```
public void tick(){  
    for(CoreObject obj: objects)  
        obj.tick();  
}
```

```
public void render(Graphics g){  
    for(CoreObject obj: objects)  
        obj.render(g);  
}
```

# Class: StageGenerator

**BuildTerrain():**

**Init():**



# Class: Renderer

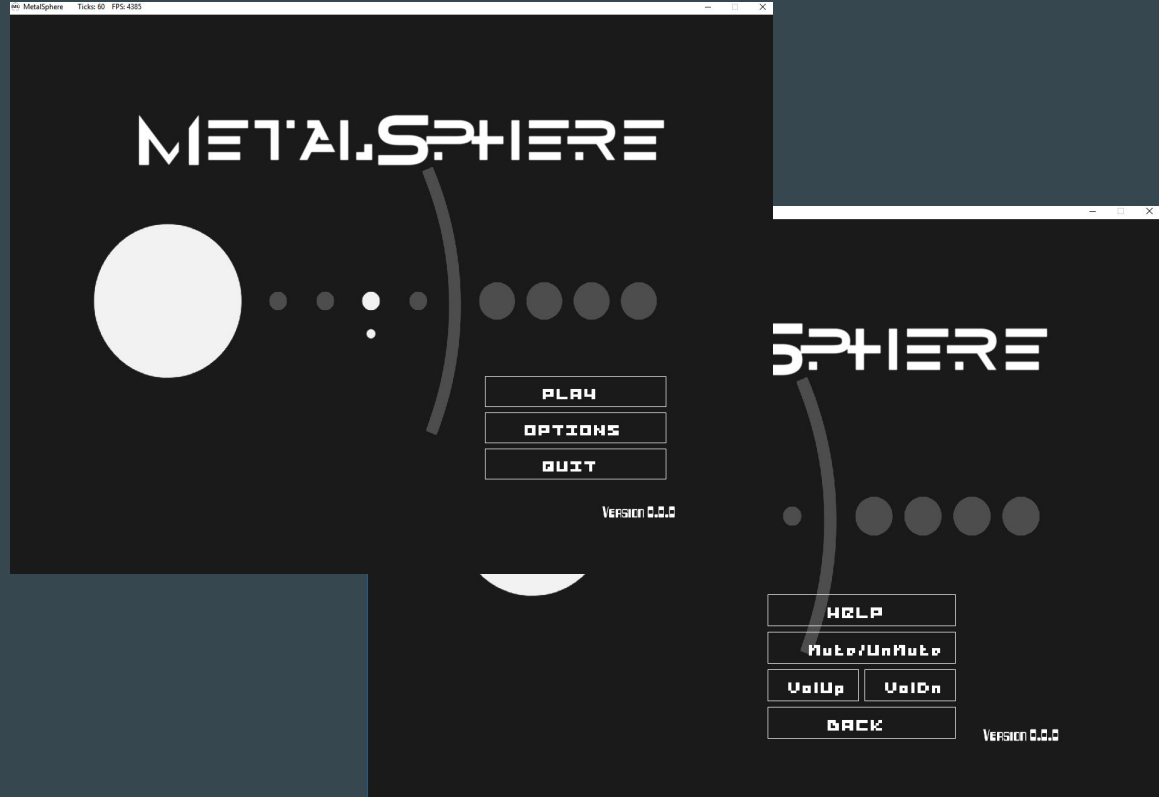
**renderBackground()**



**renderForeground():** metal ball, blocks....

# Class: Screens Packages

Menu  
Option  
Help  
Play  
Intermediate  
Pause  
Over  
Victory



# Class: Input

<b>A</b>	<b>left</b>
<b>D</b>	<b>right</b>
<b>Space</b>	<b>Jump</b>
<b>Esc</b>	<b>Pause</b>

**Case Game:**

```
if(key==KeyEvent.VK_D && ...  
    && !player.isLeftCollision()) {  
    ...  
}  
  
if(key==KeyEvent.VK_A && ...  
    && !player.isRightCollision()) {  
    ...  
}
```

**Demo**

# Conclusion & Future work

**More Feature**

**Map Editor**

**Other Game**

**...**



Q&A