

BREAKOUT



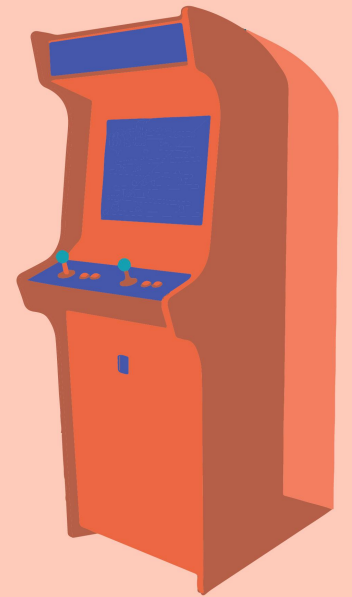
CS360 PROJECT
NORA ALSHAALAN
NORAH ALSABTI

WHAT IS BREAKOUT?

Breakout is an arcade game developed and published by **Atari, Inc.**

The game was ported to multiple platforms and upgraded to video games such as **Super Breakout**.

Atari, Inc. was an American video game development and home computer company founded in 1972 by Nolan **Bushnell** and Ted **Dabney** [1].



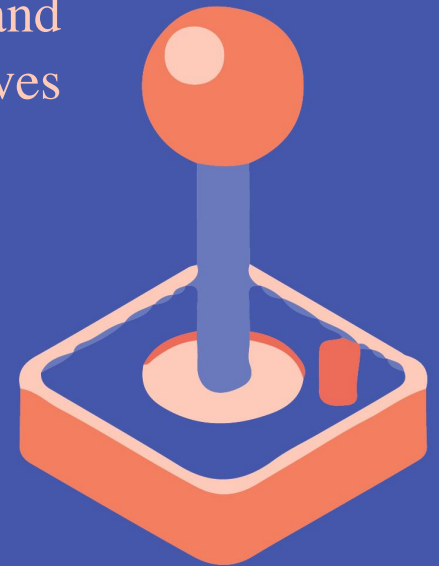
GAME PLAY

The game consists of **layers of blocks**, a **ball**, and a **movable controller block**.

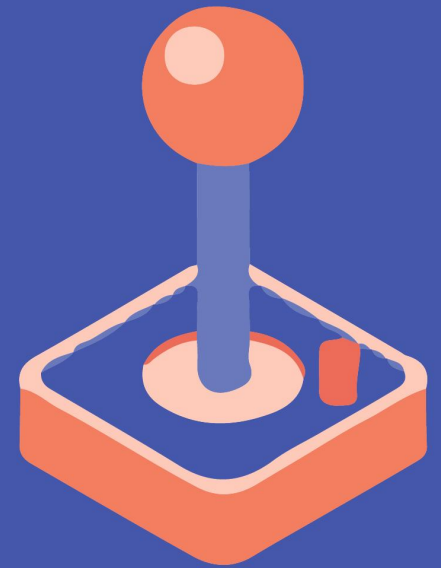
The game start once the player presses **SPACE** bar.

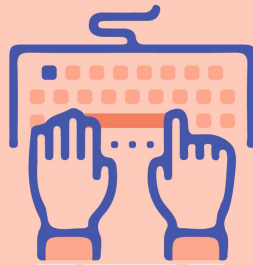
It starts with the ball traveling across the screen, bouncing off the top and side walls of the screen as well as on the movable controller, which moves by pressing the **right** and **left** keys.

When a block is hit, the ball bounces away and the block is **destroyed**.
If the ball falls, the player **loses**.



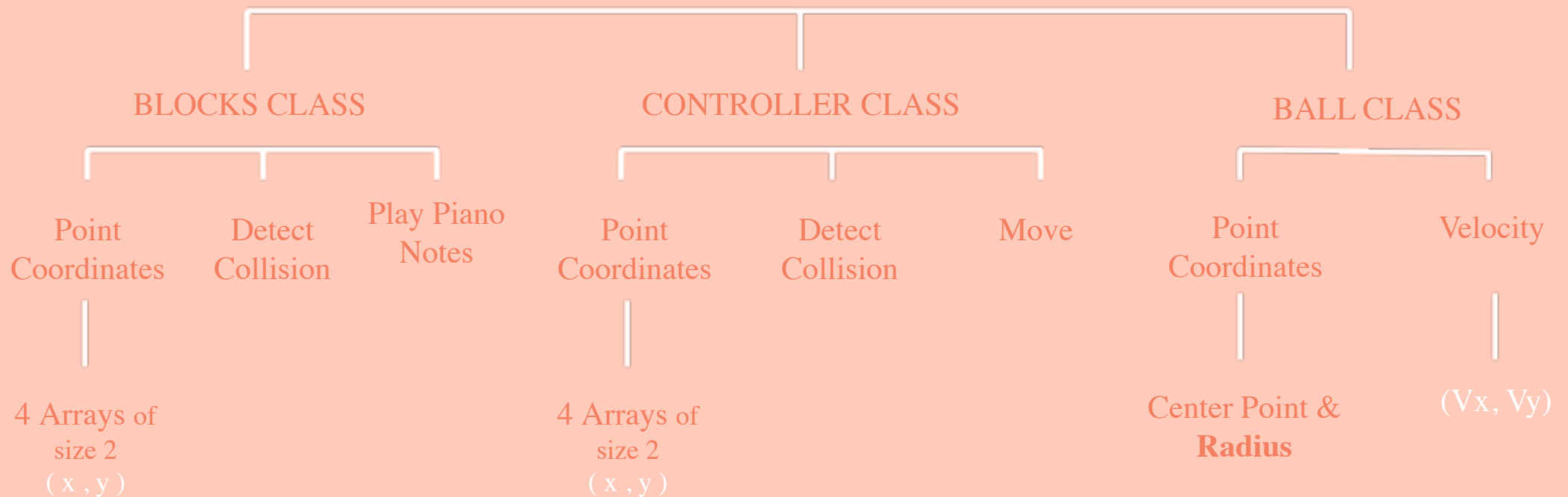
LET'S PLAY





IMPLEMENTATION

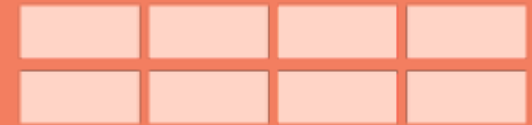
PROGRAM INFRASTRUCTURE



CONSTRUCT & DRAW

```
Blocks[][] BLOCKS[i][j] = new Blocks[rows][cols]....;
```

→ GL_QUADS



```
Controller player = new Controller....;
```

→ GL_QUADS



```
Ball ball = new Ball....;
```

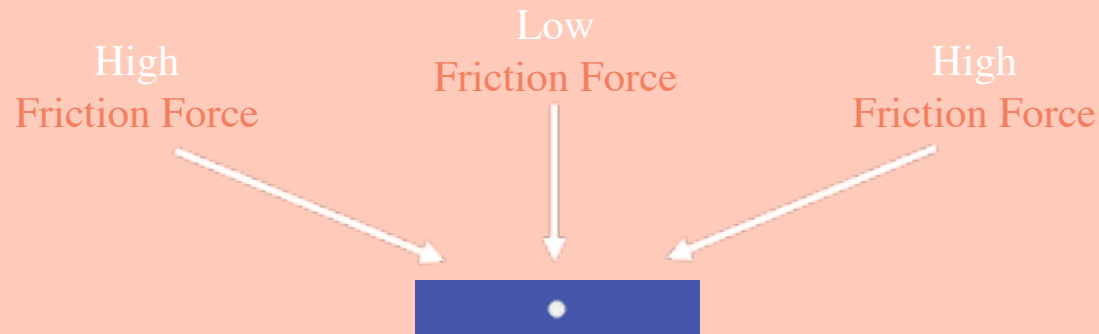
→ GL_TRIANGLE_FAN



CONTROLLER

It's not just bouncing back

The **controller** applies friction force to the **BALL** that will alter the bouncing behavior.



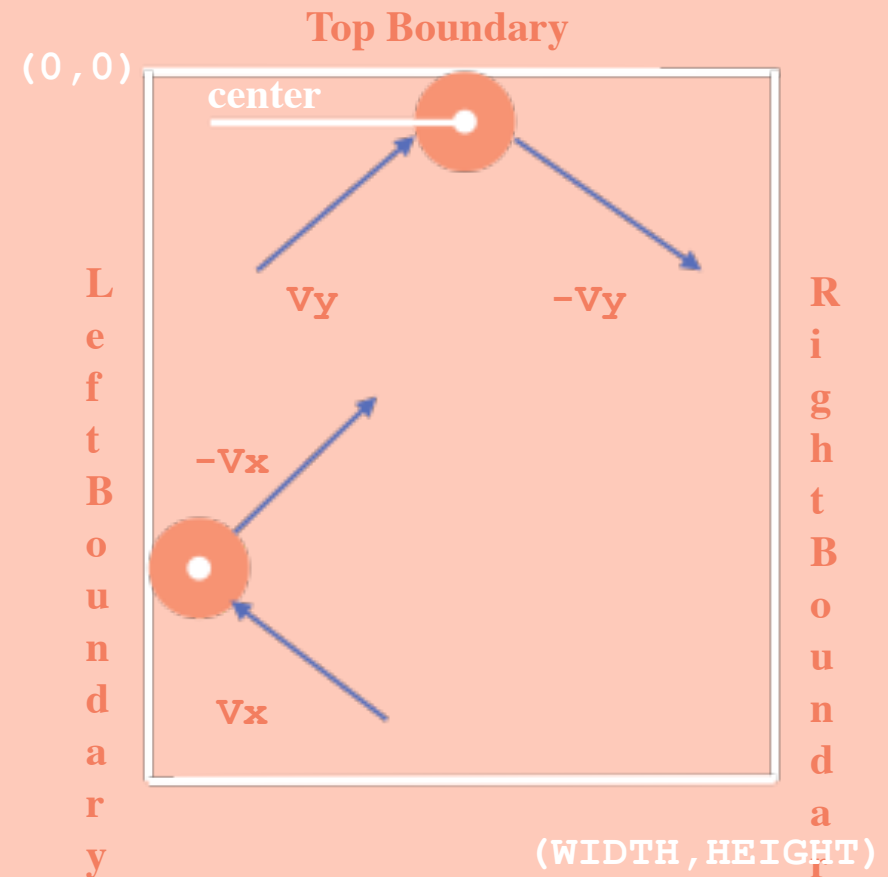
```
ratio= (collisionPoint - ControllerCenter) / WIDTH  
Vx= ratio * V  
Vy= (1-ratio)* V
```

BALL MOVEMENT

```
Ball.CenterX += Vx;  
Ball.CenterY += Vy;
```

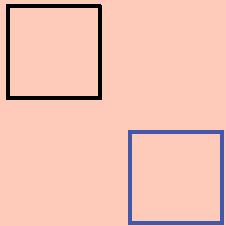
To make sure it doesn't fly out of the borders

```
public void withinBorders() {  
  
    if (CenterX + R >= WIDTH || CenterX - R <= 0.0)  
        Vel_com[0] *= -1;  
  
    else if (CenterY - R <= 0 )  
        Vel_com[1] *= -1;  
}
```

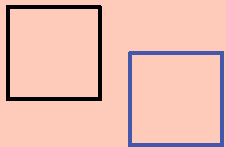


COLLISION DETECTION

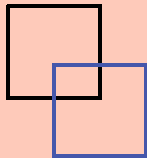
No overlap



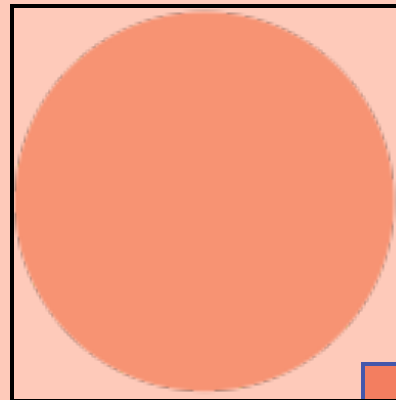
One axis overlap



Two axis overlap



AABB collisions [3]
Axis-aligned bounding box



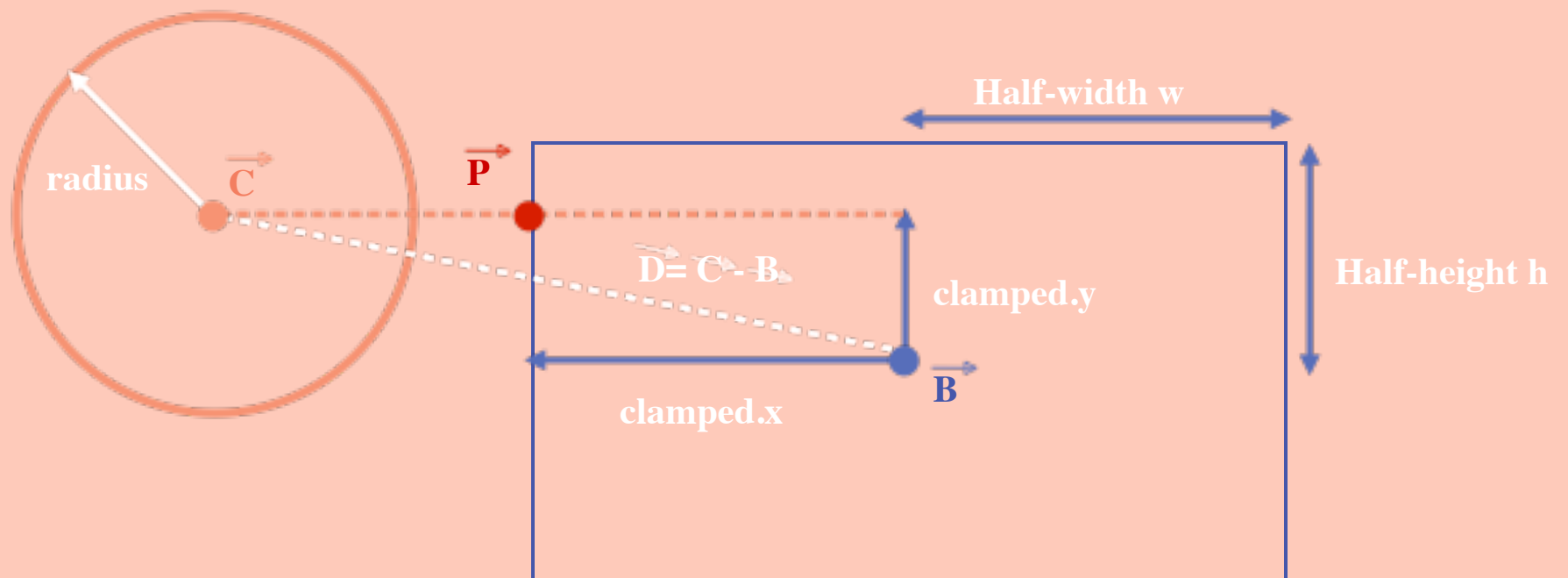
PROBLEM
This is considered
collision



CHALLENGE

AABB - Circle collision detection

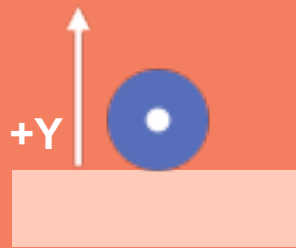
Finding \vec{P}



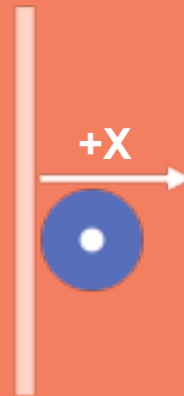
BOUNCING BACK

PROBLEM

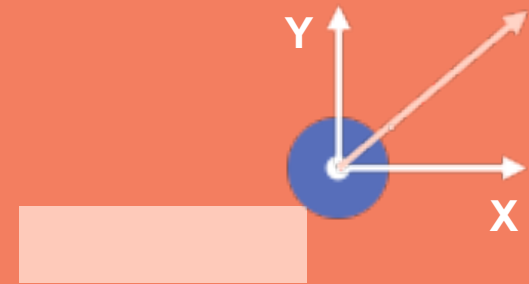
This is very hard to
implement



$$V_y = -V_y$$



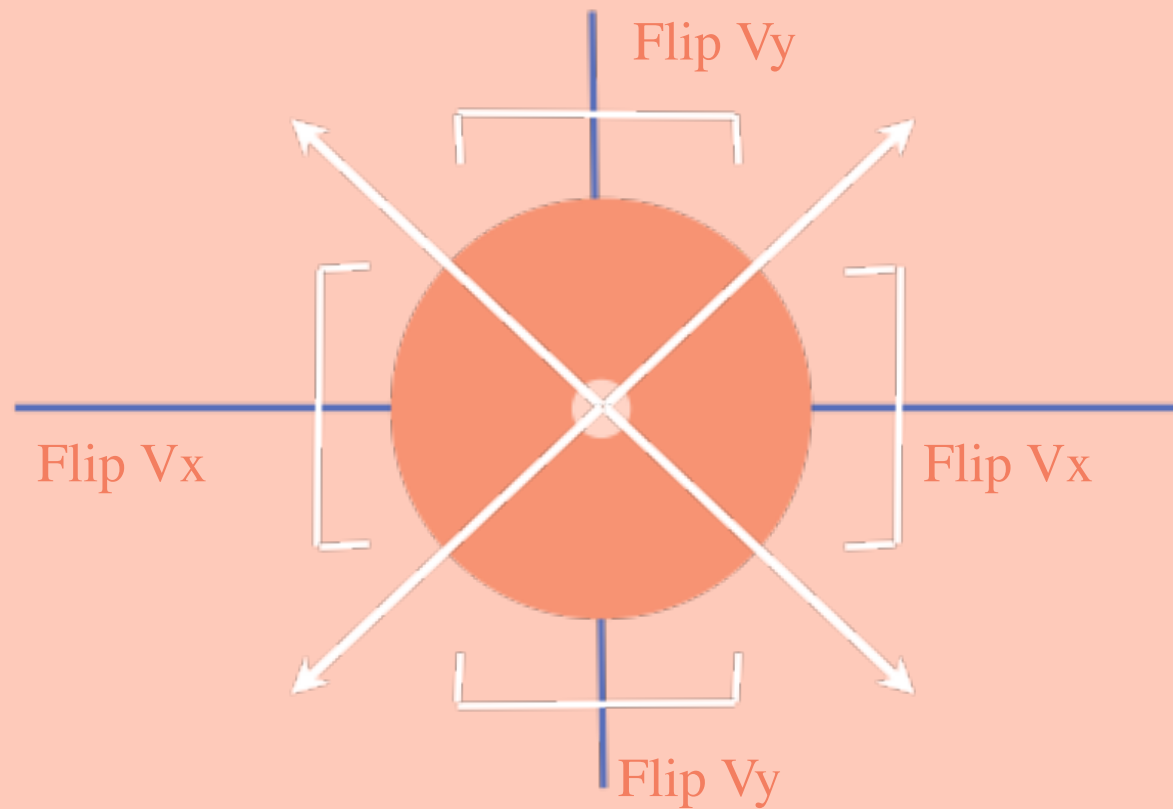
$$V_x = -V_x$$



$$V_x = V * x/r$$
$$V_y = V * y/r$$

SOLUTION

ROUGH APPROXIMATION



BREAKING BLOCKS

```
//In display(GLAutoDrawable drawable)
```

```
if( !BLOCKS[i][j].isBroken() ){
```

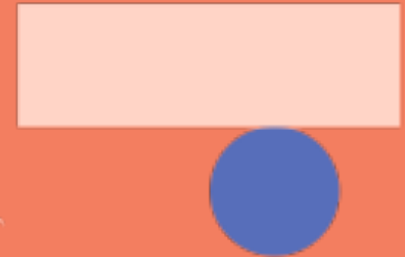
```
    //DRAW BLOCKS
```

```
}
```

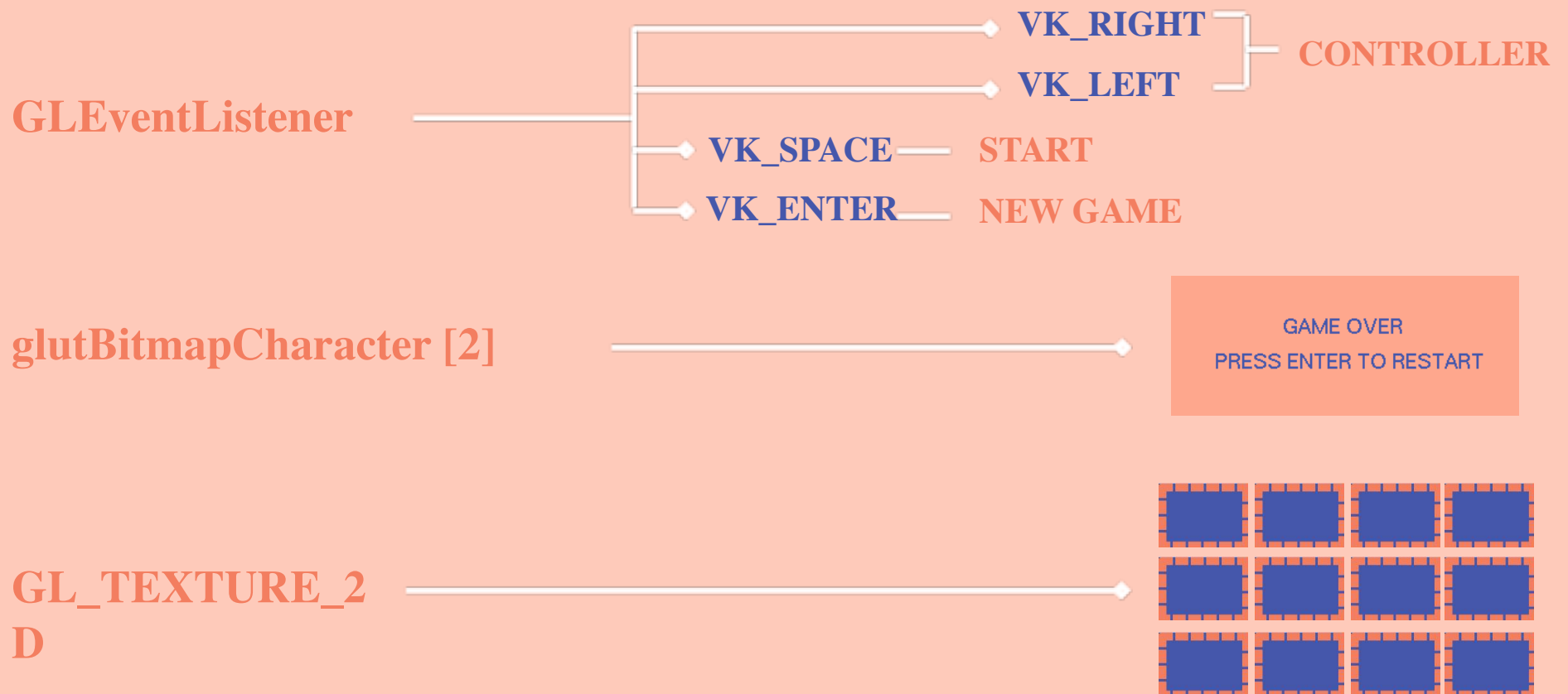
```
•  
•  
•  
•
```

```
if( !BLOCKS[i][j].isBroken() && BLOCKS[i][j].collision(ball) )
```

```
    BLOCKS[i][j].break_it();
```



ALSO USED



SCENARIOS

All blocks destroyed



WINNER



CONGRATULATIONS
YOU'RE A WINNER
PRESS ENTER TO PLAY AGAIN

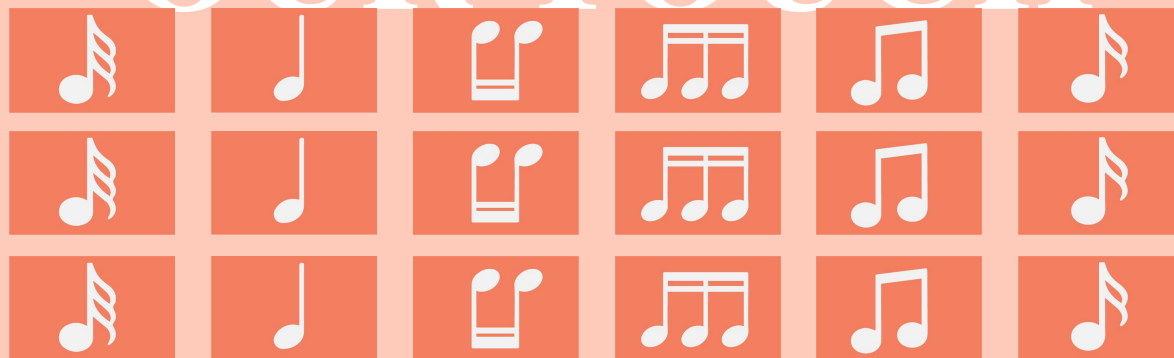
Ball falls down



GAMEOVER

GAME OVER
PRESS ENTER TO PLAY AGAIN

OUR TOUCH



ADDING SOUNDS

EACH BLOCK WAS ASSIGNED A **KEY** AT THE CREATION STAGE

```
public boolean collision(Ball b){  
    //code.....  
    pianoKey(this.key); //pass this block's key note  
}
```

```
public static void pianoKey( int i ) {  
    AudioInputStream audioIn = AudioSystem.getAudioInputStream(new  
                                         File ("audio\\"+i+".wav"));  
    Clip clip = AudioSystem.getClip();  
    clip.open(audioIn);  
    clip.start();  
}
```

REFERENCES

- [1] Atari History. (2012). Retrieved from <https://www.atari.com/history/1972-1984-0>
- [2] Kilgard, M. (February 23). 10.1 glutBitmapCharacter. Retrieved 1996, from <https://www.opengl.org/resources/libraries/glut/spec3/node76.html>
- [3] DeVries, J. (2015). Collision detection. Retrieved from <https://learnopengl.com/In-Practice/2D-Game/Collisions/Collision-detection>