



# 강 동 혁

## INTRODUCE

- 게임을 만들어보고 싶다는 생각으로 프로그래밍을 공부한지 8년차입니다.
- 시간이 흐르면서 게임 제작 뿐 아니라 다양한 분야에 관심이 생겨 임베디드 등 여러 분야도 접해보았습니다.
- 항상 어떤 게임이 사람들에게 재미를 줄지 게임의 미래는 어떨지 생각하며 개발과 공부를 하고있습니다.

## EDUCATION

- 한국게임과학고등학교 ('11~'13)
- 한양대학교ERICA 소프트웨어 ('14~'20)

## CONTACT

- 010-2931-4487
- [kdh950812@nate.com](mailto:kdh950812@nate.com)
- [github.com/Kangdonghyuk](https://github.com/Kangdonghyuk)

## SKILL

- C
- C++
- Java
- Unity3D

## AWARD

- 2019 SW/ICT 종합학술대회 우수상
- 2018지능형모형차 BMW특별상
- 2013 Smarteen App Challenge 엔터부문 장려상
- 푸르덴셜 착한 프로젝트 공모전 우수활동팀
- 제11회 청소년 게임캠프 금상2013 전국 게임공모전 금상
- 제11회 전국학생게임공모전 공주대학교 1Page부문 우수상

## PROJECT

- TBall (Android Game)
- Sheep (Android Game)
- 짝자궁 (Android Game)
- TetrisAdventure (PC Game)
- BounceBall (PC Game)
- FruitPang(Android Game)
- Rear Occupant Alert (Android)

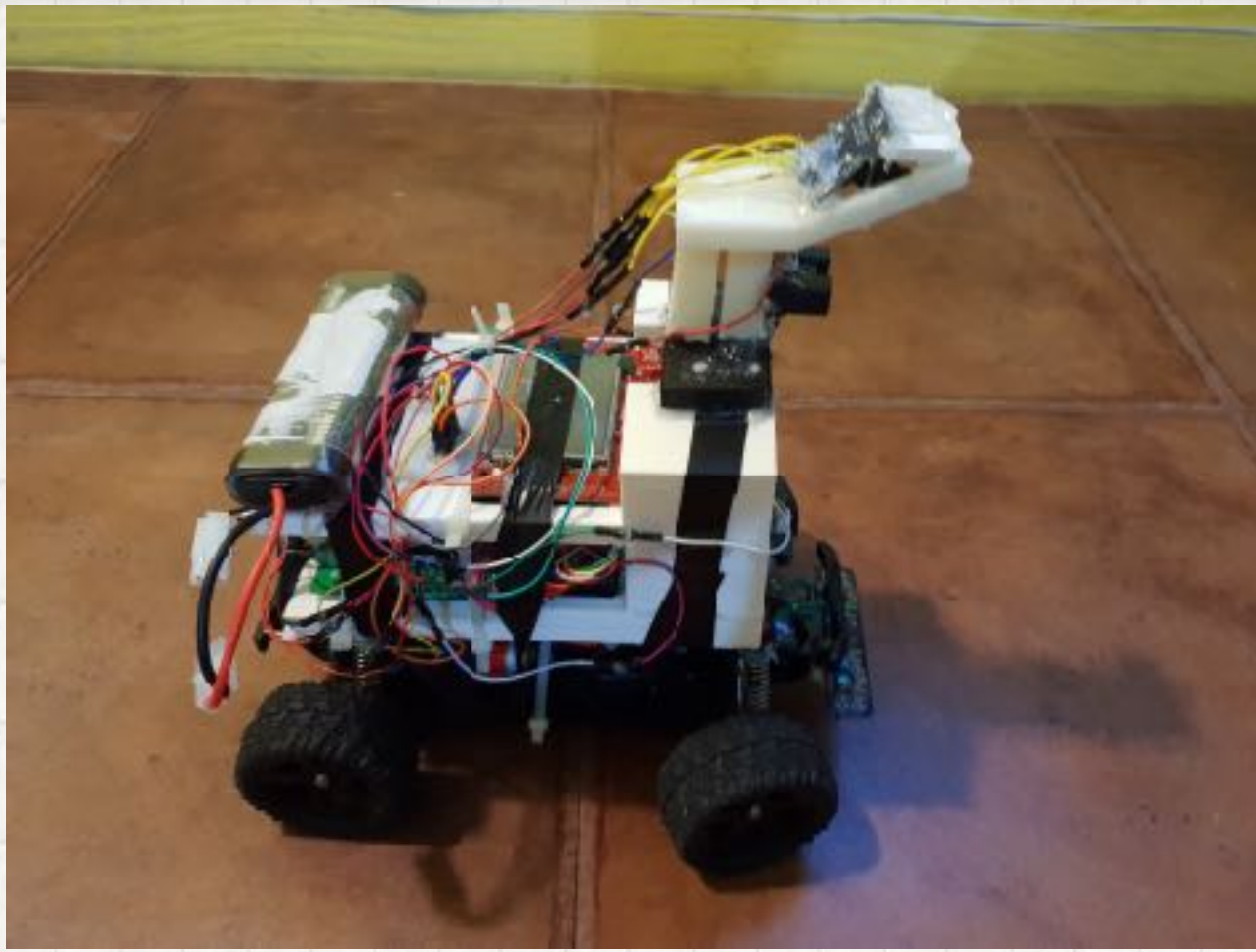
## ETC

- STAC Unity 프로그래밍 교육
- 대한민국 대학생 [함성소리, 알락달락 행복한 교실] 초등학생 프로그래밍 교육 봉사 참여



# 지능형모형차경진대회(2018 / 2019)

- 언어 : C
- 담당 : 개발 총괄 및 라인유지알고리즘 개발
- Git : <https://github.com/Kangdonghyuk/SmartLegoCar>



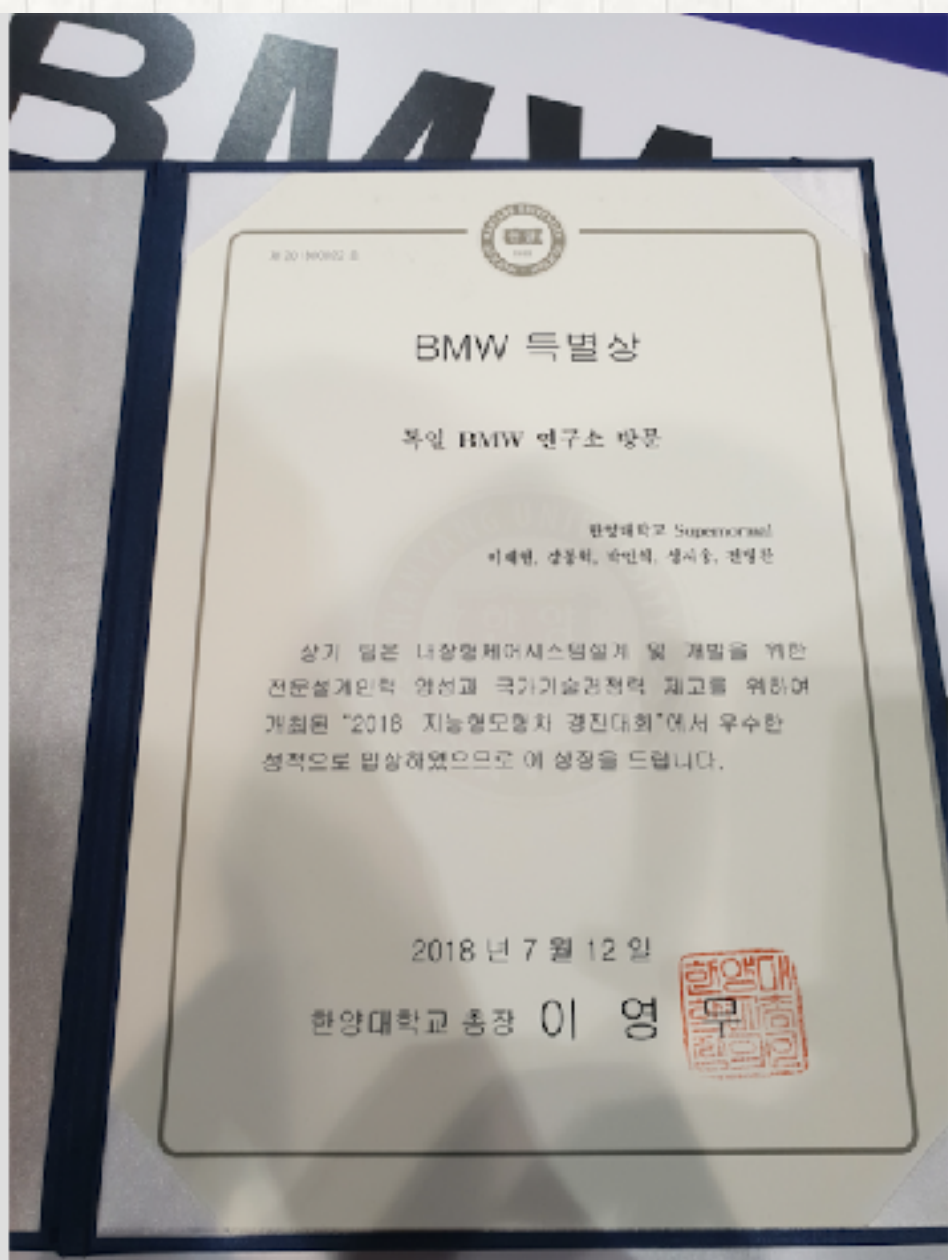
```
int FindCenter(int(*_line)(SIZE)) {
    int i, index = 0, leftIndex = 0, rightIndex = 127, zeroCount = 0;

    for (i = 0; i < SIZE; i++) {
        if ((*_line)[i] < 0)
            (*_line)[i] = 0;
        if ((*_line)[i] < THRESHOLD) {
            zeroCount++;
            (*_line)[i] = 0;
        }
    }

    if(zeroCount <= 128) return -1;

    for(i=1; i<75; i++) {
        if((75 - i - 1) >= 1) {
            if(leftIndex == 0 && (*_line)[75 - i] != 0 && (*_line)[75 - i - 1] == 0)
                leftIndex = 75 - i;
        }
        if((75 - i + 1) <= 125) {
            if(rightIndex == 127 && (*_line)[75 + i] != 0 && (*_line)[75 + i + 1] == 0)
                rightIndex = 75 + i;
        }
    }

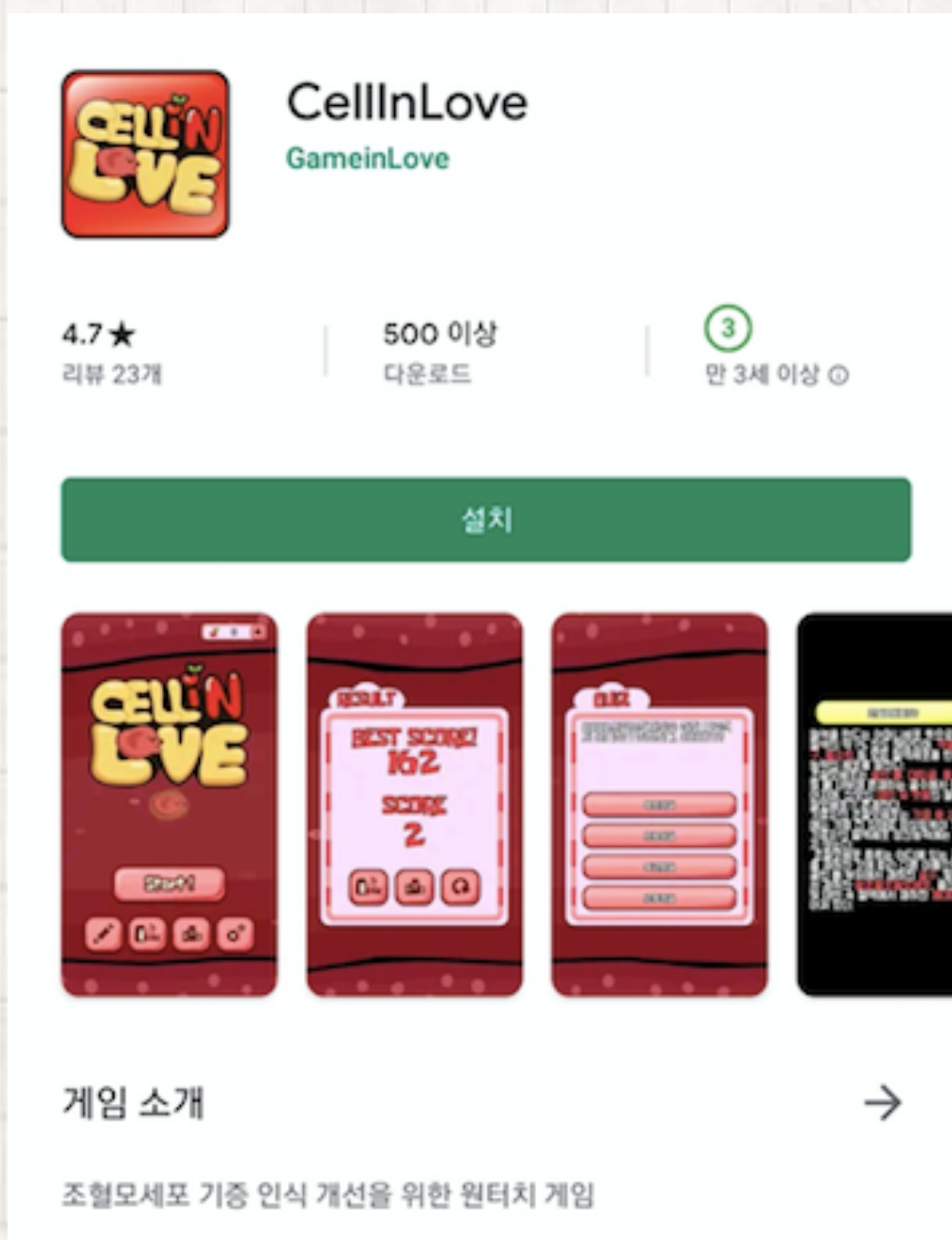
    if(cntTotal < LEFTMAXCOUNT && isDashLine == true) {
        cntTotal += 1;
        if(leftIndex != 0)
            cntLeft += 1;
        if(cntTotal == LEFTMAXCOUNT) {
            if(cntLeft >= LEFTTHRESHOLD)
                whereDashLine = F_RIGHT;
            else if(cntLeft < LEFTTHRESHOLD)
                whereDashLine = F_LEFT;
        }
    }
}
```





# 푸르덴셜 착한프로젝트 공모전(2017)

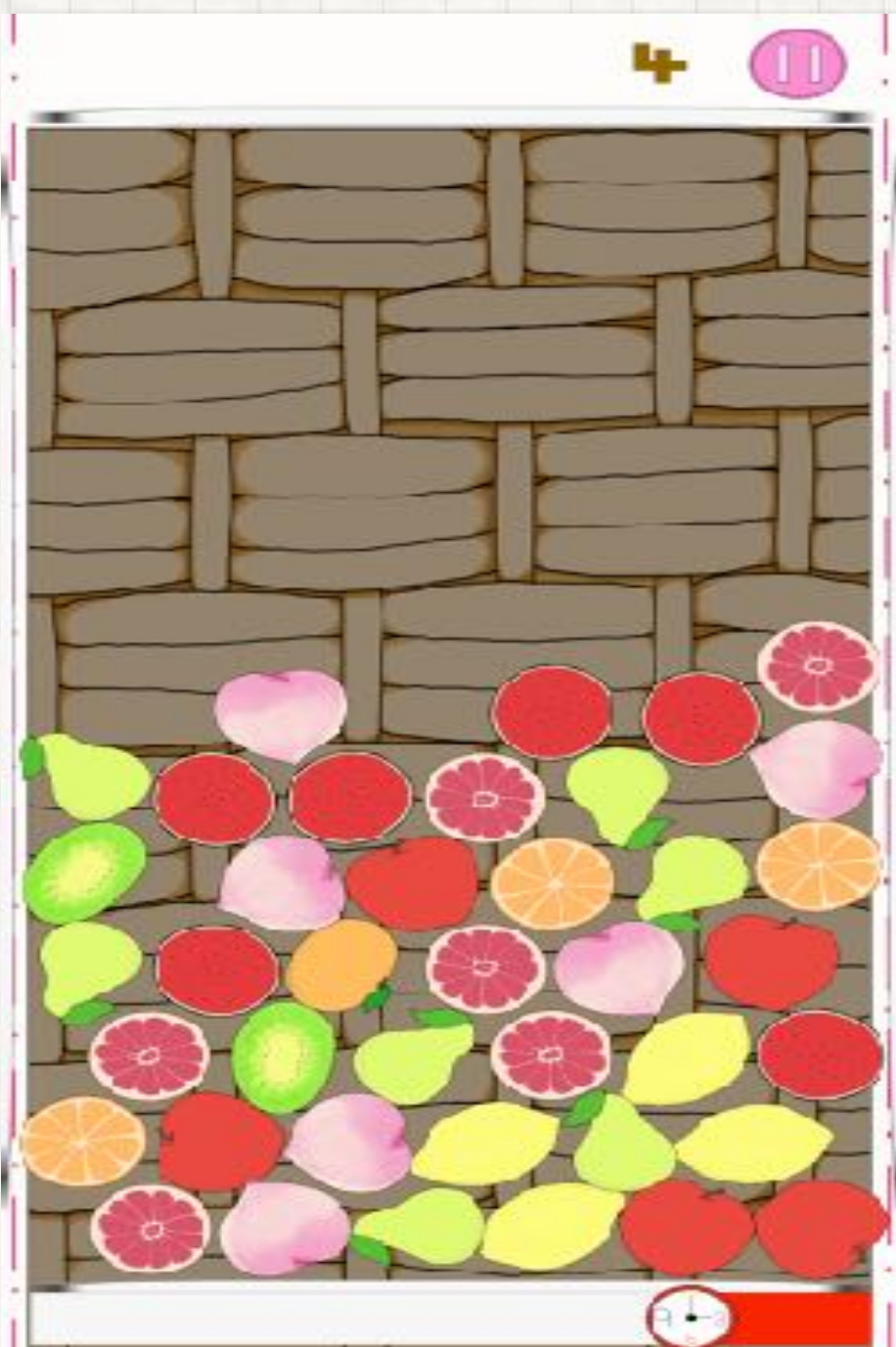
- 프로그램 : Unity
- 언어 : C#
- 담당 : 테스트 및 개발 보조
- 설명 : 조혈모세포 기증 인식 개선 게임과 융합
- 링크 : 구글스토어 SellInLove(<https://play.google.com/store/apps/details?id=com.GIL.CellInLove>)





# FruitPang

- 프로그램 : Unity3D - 언어 : C# - 담당 : 프로그래밍
- 설명 : 같은 종류 과일이 3개 이상 모이면 터지는 3match게임입니다. 블록을 자유롭게 움직일 수 있습니다.
- Git : <https://github.com/Kangdonghyuk/Pang-3MatchGame-unity->
- 링크 : 구글스토어 FruitPang ( <https://play.google.com/store/apps/details?id=com.OpenGames.Pang> )



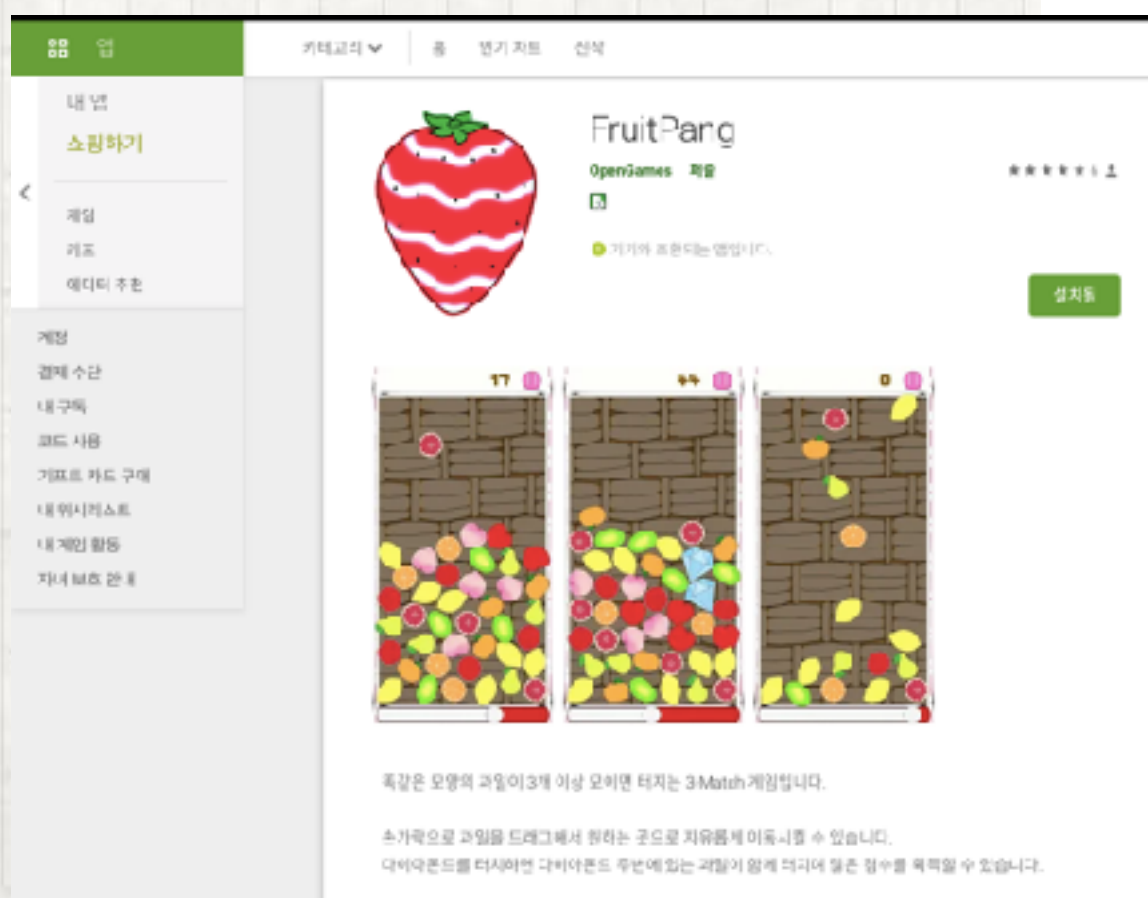
```
for(int index = 0; index < PangInfo.PANG_MAX; index++) {
    if(pangList[index].isActive == true) {
        if(IsNear(position, pangList[index].position, 1.5f) == true) {
            PangCreator.I.Destroy(index);
            GameMNG.I.AddScore(11);
        }
    }
}

/*If(comboTime >= nowTime) {
    comboCount += 1;
    comboTime = nowTime + 1.0f;
    combo.SetCombo(comboCount);
}*/
comboCount = combo.AddCombo();

ItemCreator.I.Boom();
GameMNG.I.AddScore(10);

#if UNITY_IPHONE || UNITY_ANDROID
    Handheld.Vibrate();
#endif
}

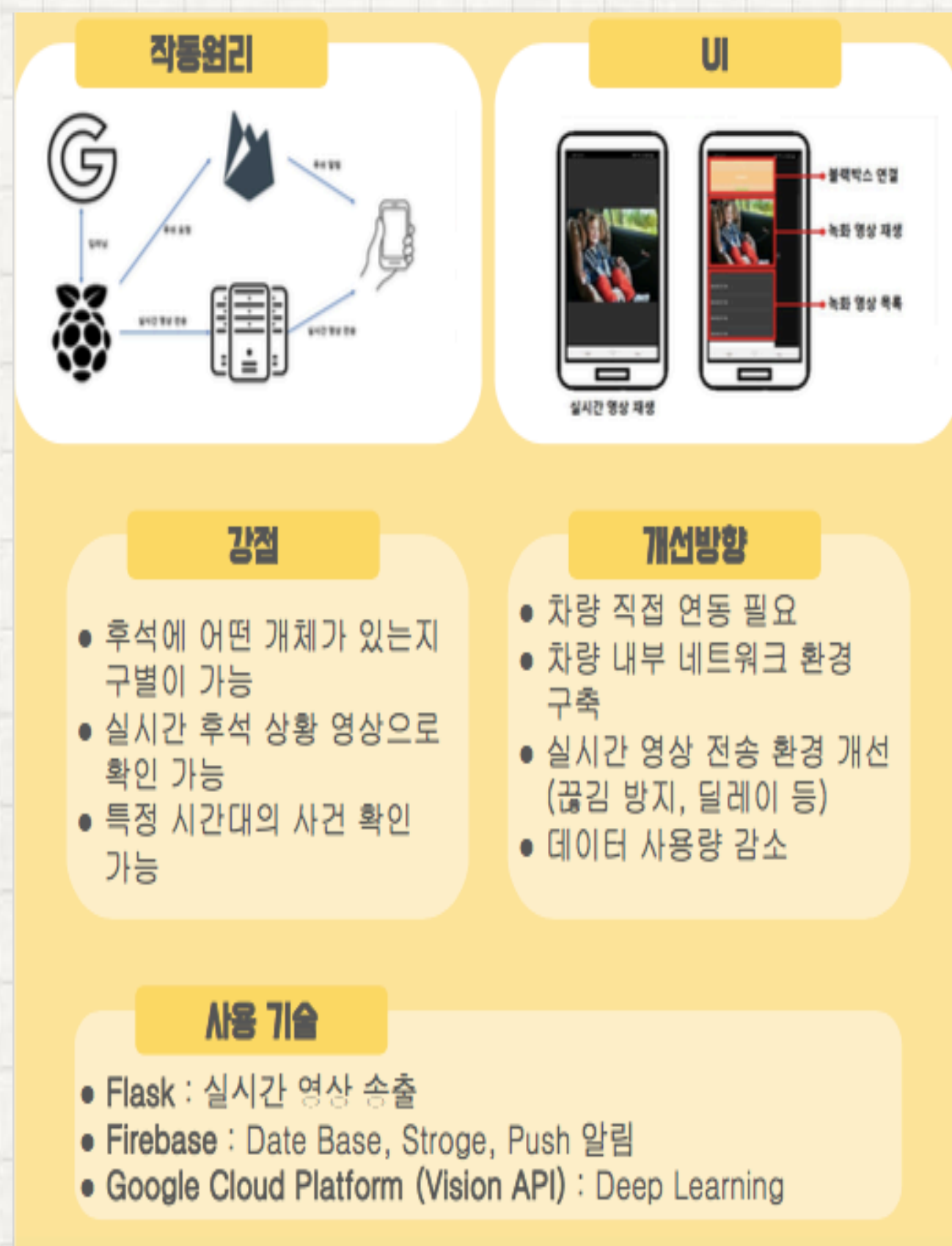
void CheckAround(int offset) {
    for(int index = 0; index < PangInfo.PANG_MAX; index++) {
        if(pangList[index].isActive == true && pangList[index].type == checkType &&
            pangList[index].checkedCount == 0) {
            if(IsNear(pangList[offset].position, pangList[index].position, 0.85f) == true) {
                pangList[index].checkedCount = 1;
                checkedCount += 1;
                //Debug.DrawLine (pangList[offset].position, pangList[index].position, Color.red, 1.0f);
                CheckAround(index);
            }
        }
    }
}
```





# Rear Occupant Alert

- 프로그램 : AndroidStudio, Firebase
- 언어 : Java, Python
- 담당 : Android 개발
- 설명 : 교내 컴퓨터 캡스톤 디자인 개발작품
- Git : <https://github.com/Kangdonghyuk/ROACapstone>



```
{
    setOnClickListener(new View.OnClickListener() {

        onClick(View v) {
            // Not-Connection Check
            Map<String, Object> deviceList = new HashMap<>();
            Map<String, Object> deviceListValue = new HashMap<>();
            // Connection Check
            Map<String, Object> serialList = new HashMap<>();
            Map<String, Object> serialListValue = new HashMap<>();

            boolean isConnected = false;

            if (isConnected) {
                TextView_isConnected.setText("connected");
                TextView_isConnected.setTextColor(Color.GREEN);
                EditText_serialNumber.setEnabled(false);

                deviceListValue.put("Serial", editText_serialNumber.getText().toString());
                deviceListValue.put("Device", deviceToken);

                TextView_success.setText("Success Connect");

            } else (!isConnected) {
                TextView_isConnected.setText("not connected");
                TextView_isConnected.setTextColor(Color.RED);
                EditText_serialNumber.setEnabled(true);

                TextView_disconnect.setText("Disconnect");
            }

            deviceList.put(deviceToken, deviceListValue);
            serialList.put(editText_serialNumber.getText().toString(), serialListValue);

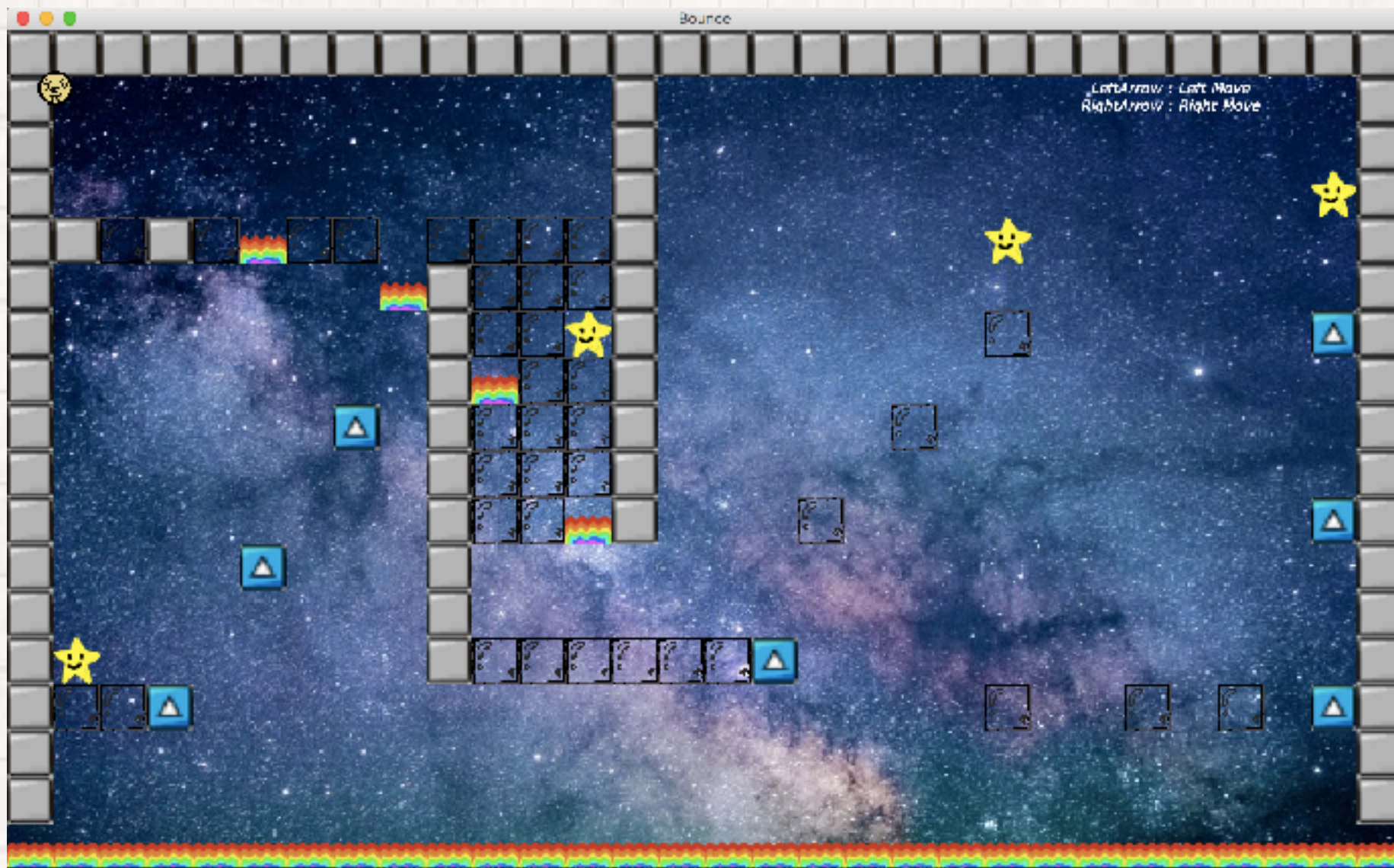
            mPostReference.updateChildren(deviceList);
            mPostReference.updateChildren(serialList);

            mNotificationManager.notify(0, builder.build());
        }
    });
}
```



# BounceBall

- 언어 : Java
- 담당 : 1인 개발
- 설명 : 스마트폰 게임으로 유명한 바운스볼 PC버전입니다.
- Git : <https://github.com/Kangdonghyuk/JavaBounceBall>



```
class AnimationRenderer extends ObjectRenderer {
    private BufferedImage[] animationImage;
    private int frame, frameSize, frameTime, nowTime;
    private int xPosition, yPosition;
    private int width, height;
    private boolean isLoop, isLooped;

    public AnimationRenderer(String fileName, int frameSize, int frameTime,
        int xPosition, int yPosition,
        int width, int height, boolean isLoop) {

        int index;

        this.frameSize = frameSize;
        this.frameTime = frameTime;
        this.xPosition = xPosition;
        this.yPosition = yPosition;
        this.width = width;
        this.height = height;
        this.isLoop = isLoop;
        isLooped = false;

        frame = 0;

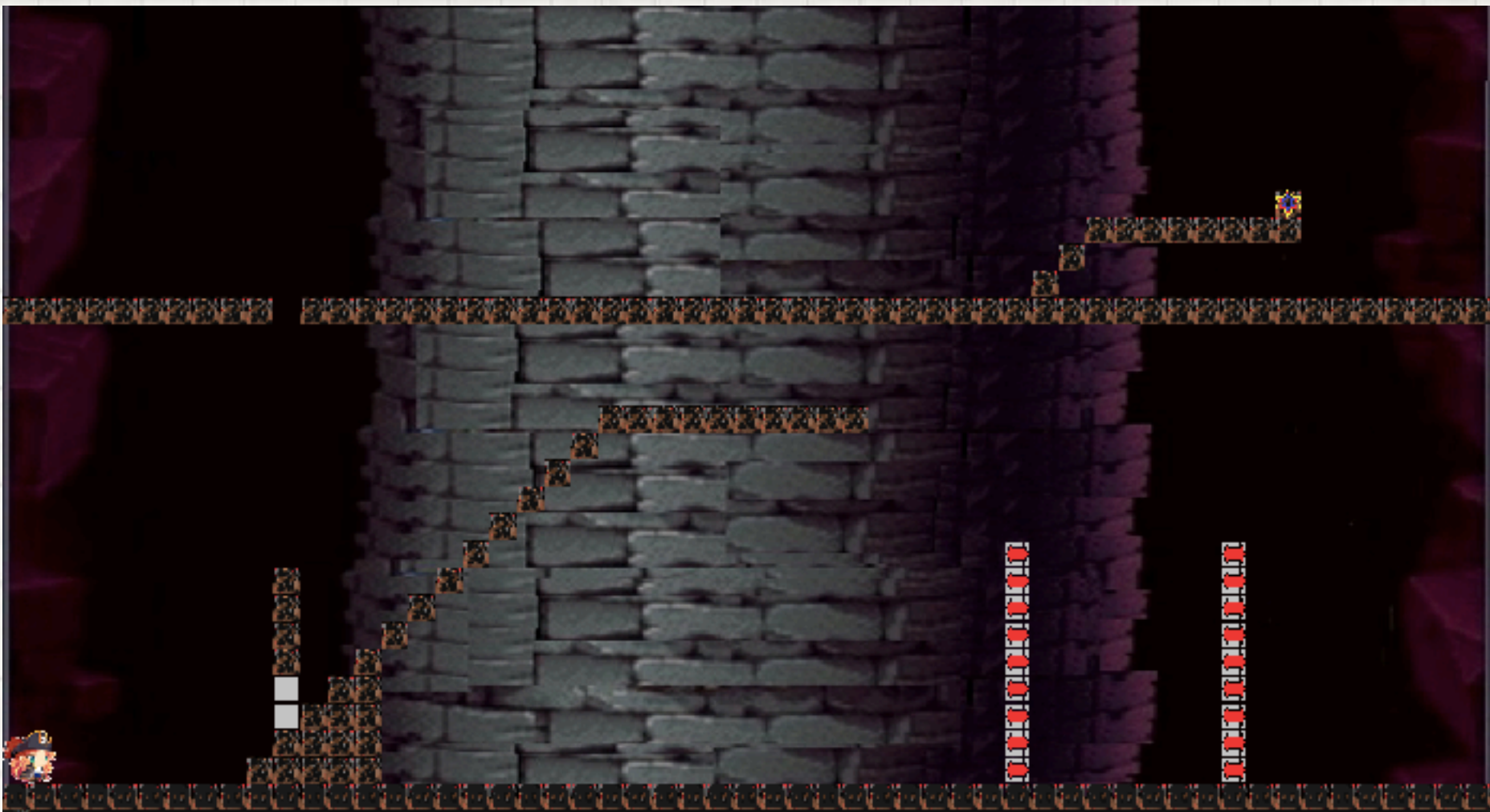
        animationImage = new BufferedImage[frameSize];
        for(index = 0; index < frameSize; index++) {
            animationImage[index] = FileManager.I.getImage(
                "src/image/" + fileName + index + ".png");
        }
        isAnimation = true;
    }

    public void paint(Graphics g) {
        if(isLooped == false) {
            g.drawImage(
                animationImage[frame],
                xPosition, yPosition,
                width, height, GameRenderer.I);
            if(GameManager.I.nowTime >= nowTime) {
                nowTime = GameManager.I.nowTime + frameTime;
                frame = (frame + 1) % frameSize;
                if (frame == 0 && isLoop == false)
                    isLooped = true;
            }
        }
    }
}
```



# TetrisAdventure

- 언어 : Python
- 담당 : 1인 개발
- 설명 : 테트리스와 캐릭터의 움직임 요소를 합친퍼즐 게임입니다.
- Git : <https://github.com/Kangdonghyuk/pygame>



```
def play_Hero(pygame, tiles, y_line, x_line, now_time) :
    global left_move, right_move
    global hero_speed, hero_gravity
    global x_position, y_position
    global jump_state, hero_state

    copyPosition(hero_copypos, hero_position)

    for event in pygame.event.get():
        if event.type == pygame.KEYDOWN :
            if event.key == pygame.K_LEFT :
                left_move = True
                if hero_state != 1 :
                    change_HeroState(1)
            if event.key == pygame.K_RIGHT :
                right_move = True
                if hero_state != 0 :
                    change_HeroState(0)
            if event.key == pygame.K_UP and jump_state == True:
                upTilePosition(hero_copypos)
                if tiles[y_position[0]-1][x_position[0]].block == 0 :
                    if tiles[y_position[0]-1][x_position[1]].block == 0 :
                        hero_gravity = -3.3
                        jump_state = False
            if event.key == pygame.K_SPACE :
                hero_gravity = -6.3
            if event.type == pygame.KEYUP :
                if event.key == pygame.K_LEFT :
                    left_move = False
                if event.key == pygame.K_RIGHT :
                    right_move = False

        if left_move == True :
            hero_copypos[0] -= hero_speed
            upTilePosition(hero_copypos)
            checkTileX(tiles, y_line, x_line, hero_copypos)
        if right_move == True :
            hero_copypos[0] += hero_speed
            upTilePosition(hero_copypos)
            checkTileX(tiles, y_line, x_line, hero_copypos)

        if left_move == False and right_move == False :
            if hero_state != 2 :
                change_HeroState(2)

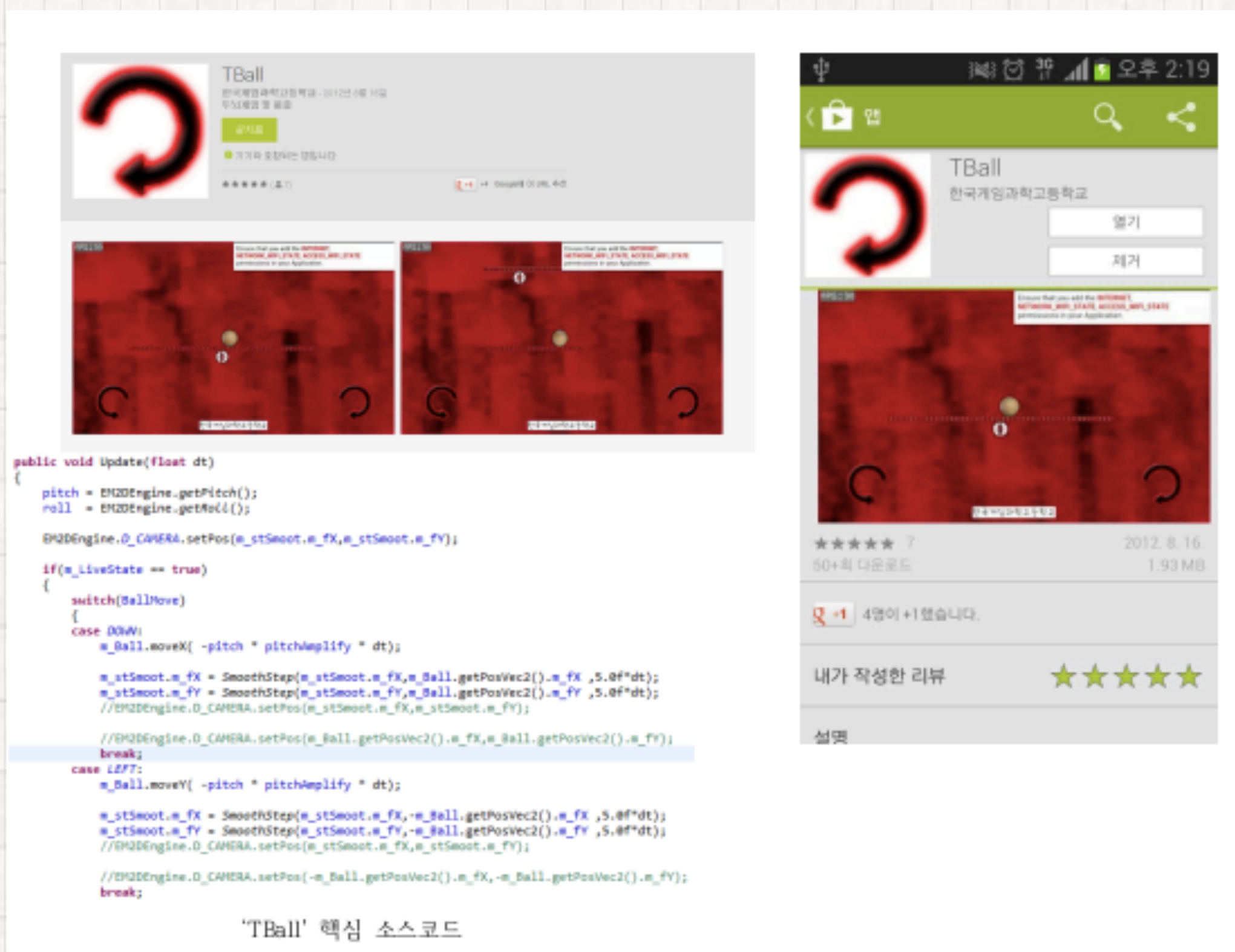
    hero_copypos[1] += hero_gravity
    if hero_gravity <= 19 :
        hero_gravity += 0.3

    upTilePosition(hero_copypos)
    checkTileY(tiles, y_line, x_line, hero_copypos)
    copyPosition(hero_position, hero_copypos)
```



# TBall

- 프로그램 : Eclipse
- 언어 : Java, EMEngine(교내 자체 제작 엔진)
- 담당 : 1인 개발
- 설명 : 좌 우 화살표를 클릭해 중력을 조절해 공을 목표지점으로 도달시키는 퍼즐게임입니다.
- Git : <https://github.com/Kangdonghyuk/Highschool-Project/blob/master/ETC/TBall.zip>





# Sheeper

- 프로그램 : Eclipse
- 언어 : Java, EMEngine(교내 자체 제작 엔진)
- 담당 : 프로그래밍
- 설명 : 양이 늑대를 이길 수 있도록 양 크기를 조절하는 게임입니다.
- Git : <https://github.com/Kangdonghyuk/Highschool-Project/blob/master/ETC/Sheeper.zip>



'Sheeper' 게임화면

## 'Sheeper' 핵심 소스코드

```
public void Battle() {
    EMPlane _tPlane = null;
    _tPlane = isCollision(E_SPHERE_VS_SPHERE, m_Sheep.GetPlane(), 4);
    if (_tPlane != null) {
        int _PlaneType = _tPlane.getType();
        if (m_Sheep.GetUnbeatableState() == false) {
            if (m_Sheep.GetGrowState() == true) {
                m_Sheep.Lose();
            } else {
                if (_PlaneType == m_Wolf.WOLF)
                {
                }
                else if (_PlaneType == m_Wolf.WOLF_STAY) {
                    m_ShowItem.Show(0, 4000);
                    m_Sheep.Poison();
                } else if (_PlaneType == m_Wolf.WOLF_GROW)
                {
                }
                else if (_PlaneType == m_Wolf.WOLF_POWER) {
                    m_Sheep.NGrow();
                }
                m_Sheep.NGrow();
            }
        }
        m_Wolf.Fight(_tPlane);
        m_Effect.CreateEffect(m_Sheep.GetPlane().getPosVec2(),
            _tPlane.getPosVec2(), _tPlane.getType(),
            m_Effect.EFFECT_COLLISION);
        if (m_Renderer.m_MusicState == true) {
            if (EM2DEngine.D_SOUNDING.isPlaying("fight") == false) {
                try {
                    EM2DEngine.D_SOUNDING
                }
            }
        }
    }
}
```