

컴퓨터 탈출하기

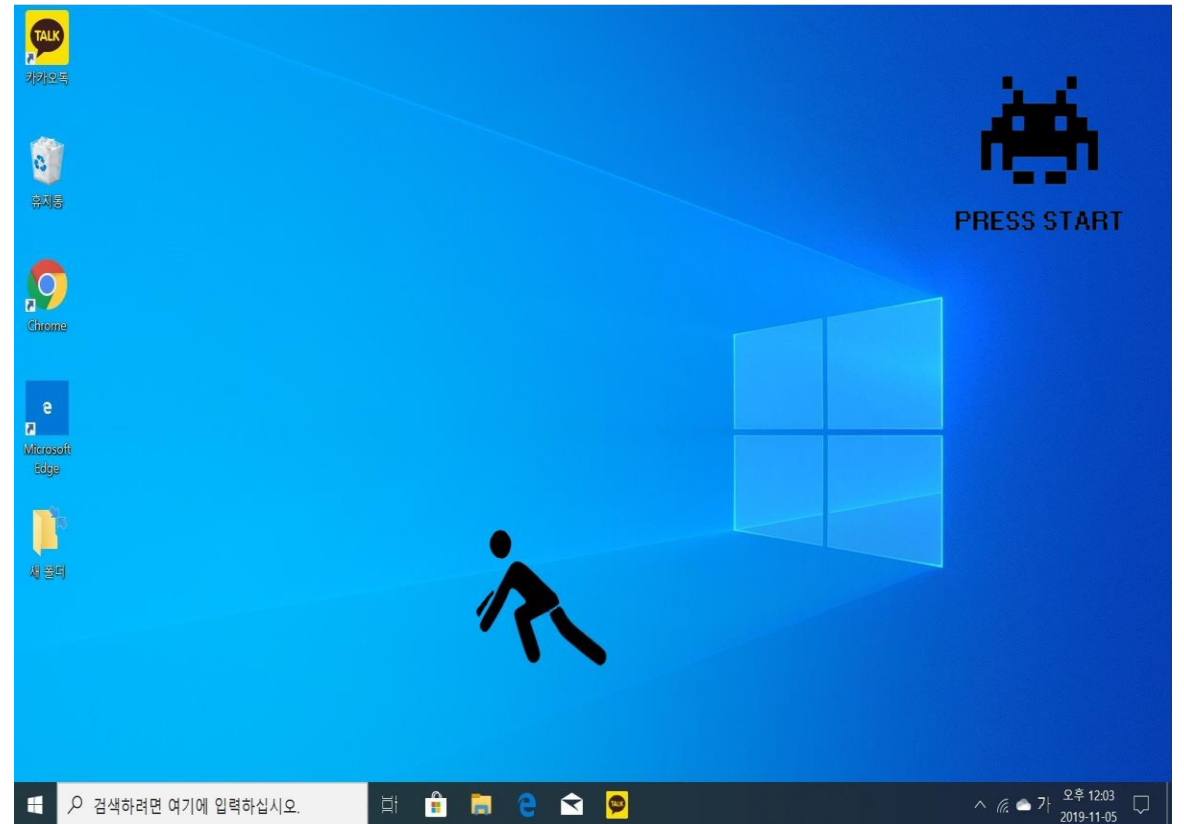
컴퓨터 탈출하기



시작버튼

Chrome scenario

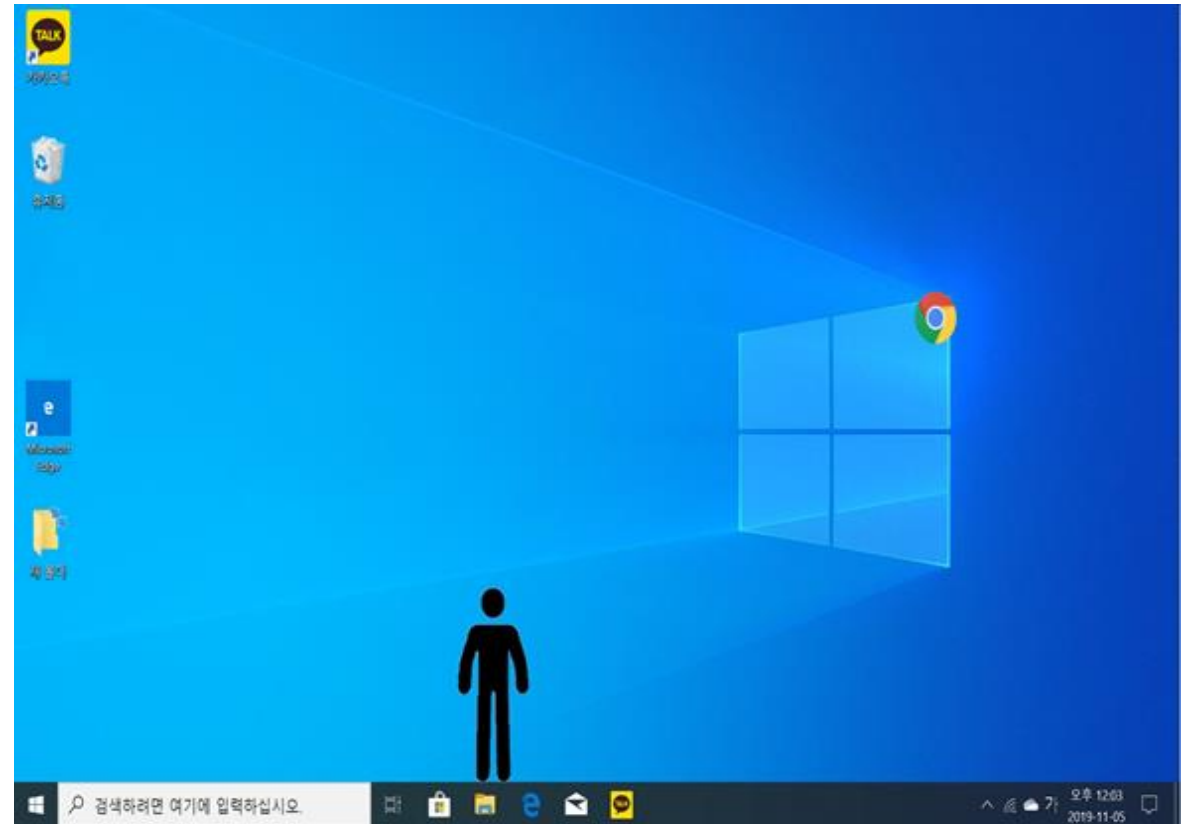
- 창문에서 캐릭터가 떨어지는 상황을 보여줌
- 상황을 보여줌과 동시에 게임이 시작됨



Game scene1

- 탈출게임 특성상 정적임

→ 동적인 요소 추가



Game scene1

코드

```
graphics = this.add.graphics();

let chrome_no_bg = this.add.image(58, 380, "chrome_no_bg").setInteractive(); //chrome 아이콘 떠다님.

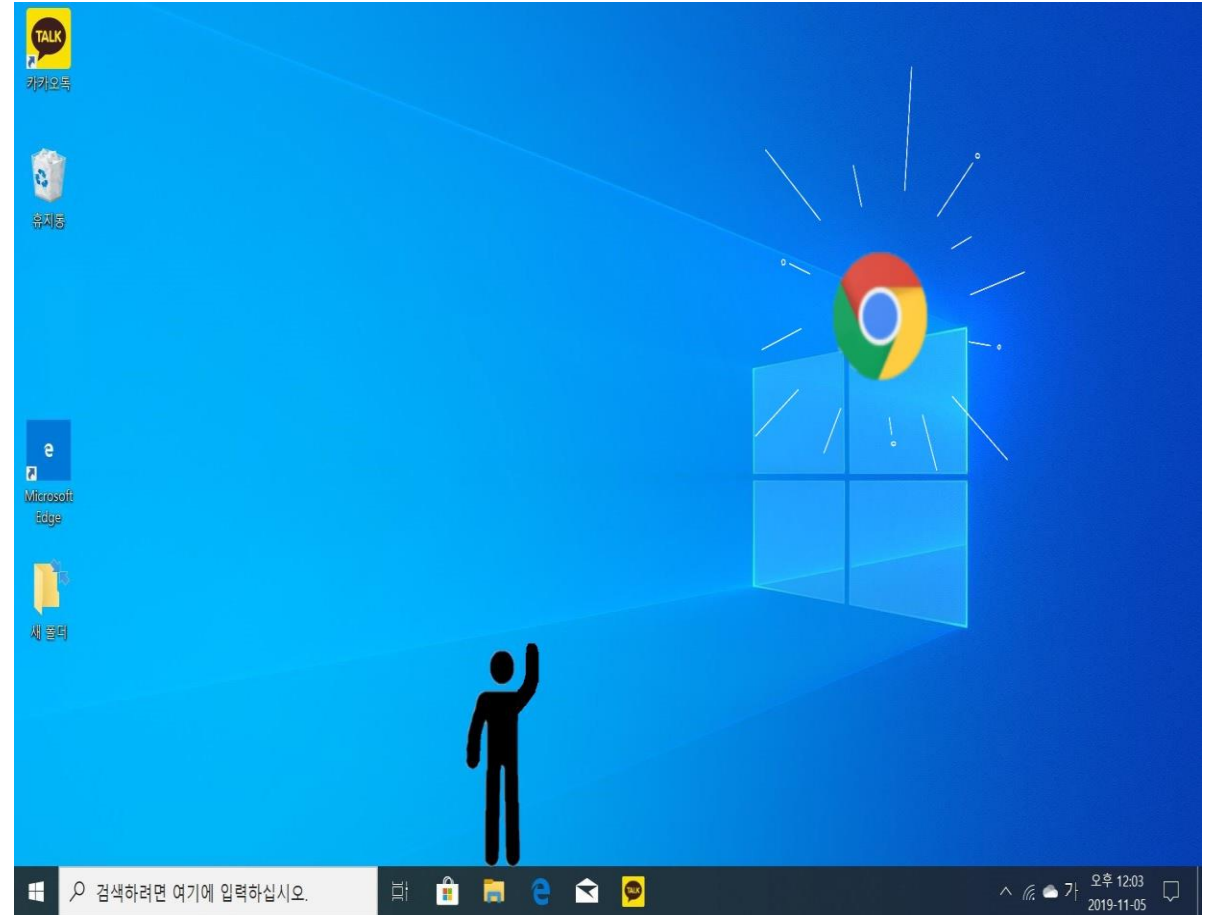
timeline = this.tweens.timeline({

  targets: chrome_no_bg,
  loop: -1, //무한반복

  tweens: [
    {
      x: 1800,
      ease: 'Sine.easeInOut',
      duration: 3000,
      yoyo: true
    },
    {
      y: 0,
      ease: 'Sine.easeOut',
      duration: 1000,
      offset: 0
    },
    {
      y: 600,
      ease: 'Sine.easeIn',
      duration: 1000
    },
    {
      y: 0,
      ease: 'Sine.easeOut',
      duration: 1000
    },
    {
      y: 100,
      ease: 'Sine.easeIn',
      duration: 1000
    },
    {
      y: 500,
      ease: 'Sine.easeOut',
      duration: 1000
    },
    {
      y: 380,
      ease: 'Sine.easeIn',
      duration: 1000
    }
  ]
});
```

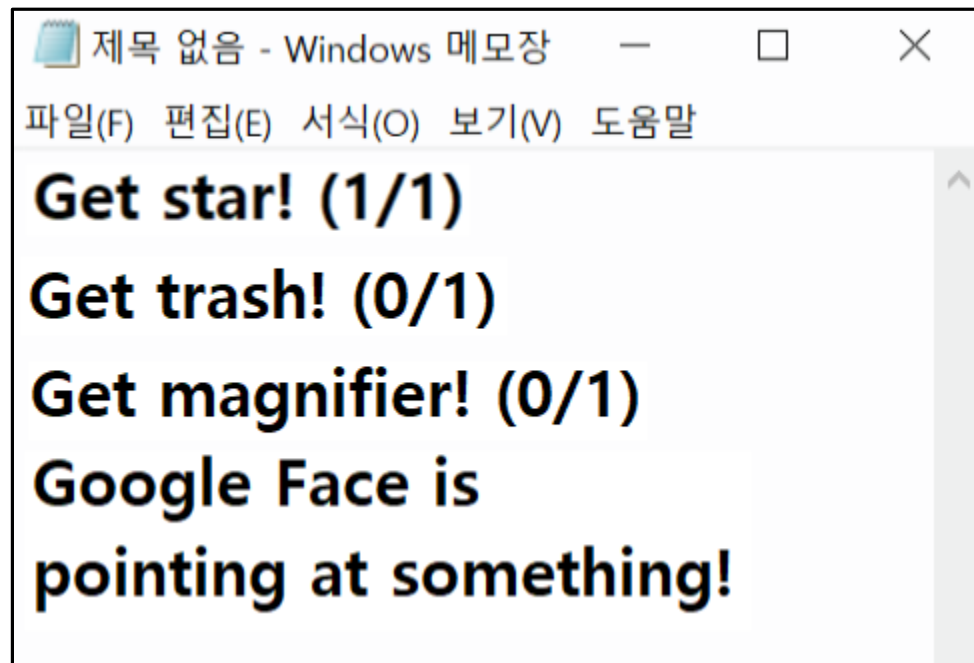
Chrome scenario

- 이미지를 순서대로 여러 개 배치해서 애니메이션 효과를 줌



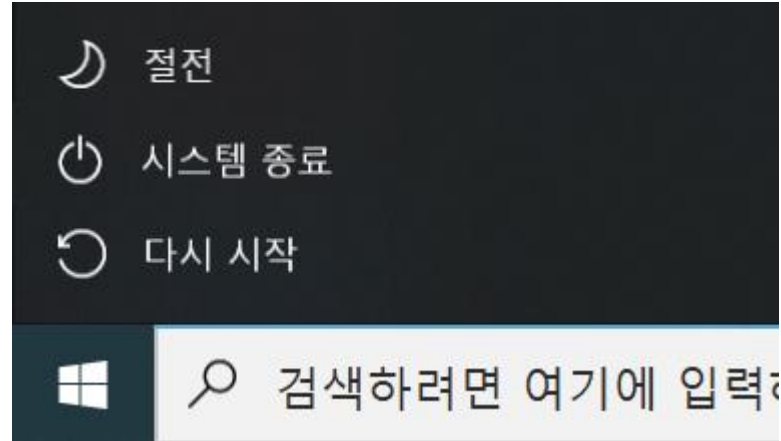
메모장

- 메모장을 사용하여 게임 진행 상황 안내
- 게임 진행에 맞추어 텍스트 출력 및 제거



윈도우 키 & 드래그

- 실제 있는 기능을 게임에 접목시킴
- 시스템 종료버튼, 다시 시작 버튼을 사용할 수 있게 함



- 드래그 & 겹침 반응 - 이미지가 겹쳐졌을 때 다음 상황으로 넘어감



드래그 & 검침 반응 코드

```
var frames = this.textures.get('magnifier_closeup').getFrameNames();
var x = 100;
var y = 100;

var zone = this.add.zone(2150, 970).setCircleDropZone(60);
var graphics = this.add.graphics();
graphics.lineStyle(1, 0xffffffff);
graphics.strokeCircle(zone.x, zone.y, zone.input.hitArea.radius);

this.input.on('dragstart', function (pointer, gameObject) {
    this.children.bringToTop(gameObject);
}, this);

this.input.on('drag', function (pointer, gameObject, dragX, dragY) {

    gameObject.x = dragX;
    gameObject.y = dragY;

});

this.input.on('dragenter', function (pointer, gameObject, dropZone) {

    graphics.clear();
    graphics.lineStyle(1, 0xffffffff);
    graphics.strokeCircle(zone.x, zone.y, zone.input.hitArea.radius);

});

this.input.on('dragleave', function (pointer, gameObject, dropZone) {

    graphics.clear();
    graphics.lineStyle(1, 0xffffffff);
    graphics.strokeCircle(zone.x, zone.y, zone.input.hitArea.radius);

});

this.input.on('drop', function (pointer, gameObject, dropZone) {
    gameObject.x = dropZone.x;
    gameObject.y = dropZone.y;

    gameObject.input.enabled = false;

});
```

```
this.input.on('dragend', function (pointer, gameObject, dropped) {

    if (!dropped) {
        gameObject.x = gameObject.input.dragStartX;
        gameObject.y = gameObject.input.dragStartY;
    }
    else {
        let wifi_password = this.add.image(960, 540, "wifi_password").setInteractive();
        let memojang_background = this.add.image(2150, 954, "memojang_background").setInteractive();

        internet_disconnect1.on("pointerup", function (event) {

            let wifi_list2 = this.add.image(1650, 827, "wifi_list");
            let wifi_hidden2 = this.add.image(1650, 777, "wifi_hidden").setInteractive();

            internet_disconnect1.destroy();
            internet_disconnect1 = null;

            let wifi_sejong = this.add.image(1650, 678, "wifi_sejong").setInteractive();

            wifi_sejong.on("pointerup", function (event) {
                let wifi_sejong_input = this.add.image(1650, 600, "wifi_sejong_input").setInteractive();
                let wifi_sejong_input_button = this.add.image(1563, 688, "wifi_sejong_input_button").setInteractive();

                wifi_sejong.destroy();
                wifi_hidden2.destroy();

                wifi_sejong = null;
                wifi_hidden2 = null;

                //다음 클릭했을 시
                wifi_sejong_input_button.on("pointerup", function (event) {
                    wifi_sejong_input.destroy();
                    wifi_sejong_input_button.destroy();
                    chrome_home_search_transparent.destroy();

                    wifi_sejong_input = null;
                    wifi_sejong_input_button = null;
                    chrome_home_search_transparent = null;

                    var internet_connect = this.add.image(1640, 1052, "internet_connect");
                    var wifi_sejong_connctet = this.add.image(1486, 692, "wifi_sejong_connect");

                }, this);

            }, this);

        }, this);

        //와이파이 비번 이미지 클릭했을 시
        wifi_password.on("pointerup", function (event) {
            let txt_wifipassword = this.add.image(2100, 382, "txt_wifipassword");

            wifi_password.destroy();

            wifi_password = null;

        }, this);

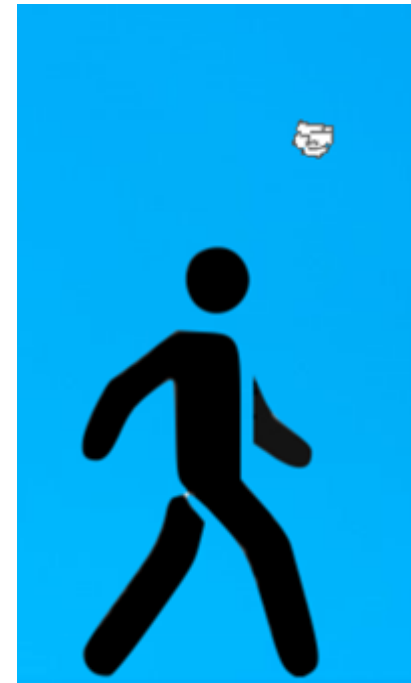
        graphics.clear();
    }
}, this);
```

동적인 요소 추가1

- 캐릭터 실제로 움직이기



- 이미지 충돌 검사



```

function create() {
    player2 = this.physics.add.sprite(660, 540, 'dude_female');
    player3 = this.physics.add.sprite(1500, 250, 'dude');

    player2.setBounce(0.2);
    player2.setCollideWorldBounds(true);

    player3.setBounce(0.2);
    player3.setCollideWorldBounds(true);
    //////////시리////////////////////
    this.anims.create({
        key: 'left2',
        frames: this.anims.generateFrameNumbers('dude_female', { start: 0, end: 3 }),
        frameRate: 10,
        repeat: -1
    });

    this.anims.create({
        key: 'turn2',
        frames: [{ key: 'dude_female', frame: 4 }],
        frameRate: 20
    });

    this.anims.create({
        key: 'right2',
        frames: this.anims.generateFrameNumbers('dude_female', { start: 5, end: 8 }),
        frameRate: 10,
        repeat: -1
    });
    //////////남캐////////////////////
    this.anims.create({
        key: 'left3',
        frames: this.anims.generateFrameNumbers('dude', { start: 0, end: 3 }),
        frameRate: 10,
        repeat: -1
    });

    this.anims.create({
        key: 'turn3',
        frames: [{ key: 'dude', frame: 4 }],
        frameRate: 20
    });

    this.anims.create({
        key: 'right3',
        frames: this.anims.generateFrameNumbers('dude', { start: 5, end: 8 }),
        frameRate: 10,
        repeat: -1
    });
}

```

```

function create() {
    player2 = this.physics.add.sprite(660, 540, 'dude_female');
    player3 = this.physics.add.sprite(1500, 250, 'dude');

    player2.setBounce(0.2);
    player2.setCollideWorldBounds(true);

    player3.setBounce(0.2);
    player3.setCollideWorldBounds(true);
    //////////시리////////////////////
    this.anims.create({
        key: 'left2',
        frames: this.anims.generateFrameNumbers('dude_female', { start: 0, end: 3 }),
        frameRate: 10,
        repeat: -1
    });

    this.anims.create({
        key: 'turn2',
        frames: [{ key: 'dude_female', frame: 4 }],
        frameRate: 20
    });

    this.anims.create({
        key: 'right2',
        frames: this.anims.generateFrameNumbers('dude_female', { start: 5, end: 8 }),
        frameRate: 10,
        repeat: -1
    });
    //////////남캐////////////////////
    this.anims.create({
        key: 'left3',
        frames: this.anims.generateFrameNumbers('dude', { start: 0, end: 3 }),
        frameRate: 10,
        repeat: -1
    });

    this.anims.create({
        key: 'turn3',
        frames: [{ key: 'dude', frame: 4 }],
        frameRate: 20
    });

    this.anims.create({
        key: 'right3',
        frames: this.anims.generateFrameNumbers('dude', { start: 5, end: 8 }),
        frameRate: 10,
        repeat: -1
    });
}

```

캐릭터 움직이기 코드

```
ground_window = this.physics.add.image(445, 750, 'ground_window');//윈도우바탕화면에 숨겨진 바닥  
ground_window2 = this.physics.add.image(55, 285, 'ground_window2');//휴지통 밑 바닥
```

```
ground_window.setImmovable(true);  
ground_window2.setImmovable(true);  
ground_window.body.allowGravity = false;  
ground_window2.body.allowGravity = false;
```

```
function collectStar2(player, trash) {  
    trash.disableBody(false, true); //false를 함으로서 1번먹고나선 더이상 만나와서 못먹게 함.  
    let txt_gettrash_1 = this.add.image(2048, 150, "txt_gettrash_1");  
    let trash_closeup = this.add.image(2150, 970, "trash_closeup").setScale(2);  
}
```

```
function createStar2(x, y, vx, vy) {  
    var trash = trashes.get();  
  
    if (!trash) return;  
  
    trash  
        .enableBody(true, x, y, true, true)  
        .setVelocity(vx, vy);  
}
```

```
cursors = this.input.keyboard.createCursorKeys();
```

```
trashes = this.physics.add.group
```

```
((  
    key: 'trash',  
    frameQuantity: 1,  
    maxSize: 1,  
    active: false,  
    visible: false,  
    enable: false,  
    collideWorldBounds: true,  
    bounceX: 0.5,  
    bounceY: 0.5,  
    dragX: 30,  
    dragY: 0  
));
```

```
this.physics.add.collider( //캐릭터랑 휴지통 밑 or 위 부딪힐 때  
    player,  
    ground_window2,  
    function (_player, _ground_window2) {  
        if (_player.body.touching.up && _ground_window2.body.touching.down) {  
            createStar2(  
                _player.body.center.x,  
                _ground_window2.body.top - 16,  
                _player.body.velocity.x,  
                _player.body.velocity.y * -3  
            );  
        }  
        else if (_player.body.touching.down && _ground_window2.body.touching.up) {  
            createStar2(  
                _player.body.center.x,  
                _ground_window2.body.top + 16,  
                _player.body.velocity.x,  
                _player.body.velocity.y * 3  
            );  
        }  
    }  
);
```

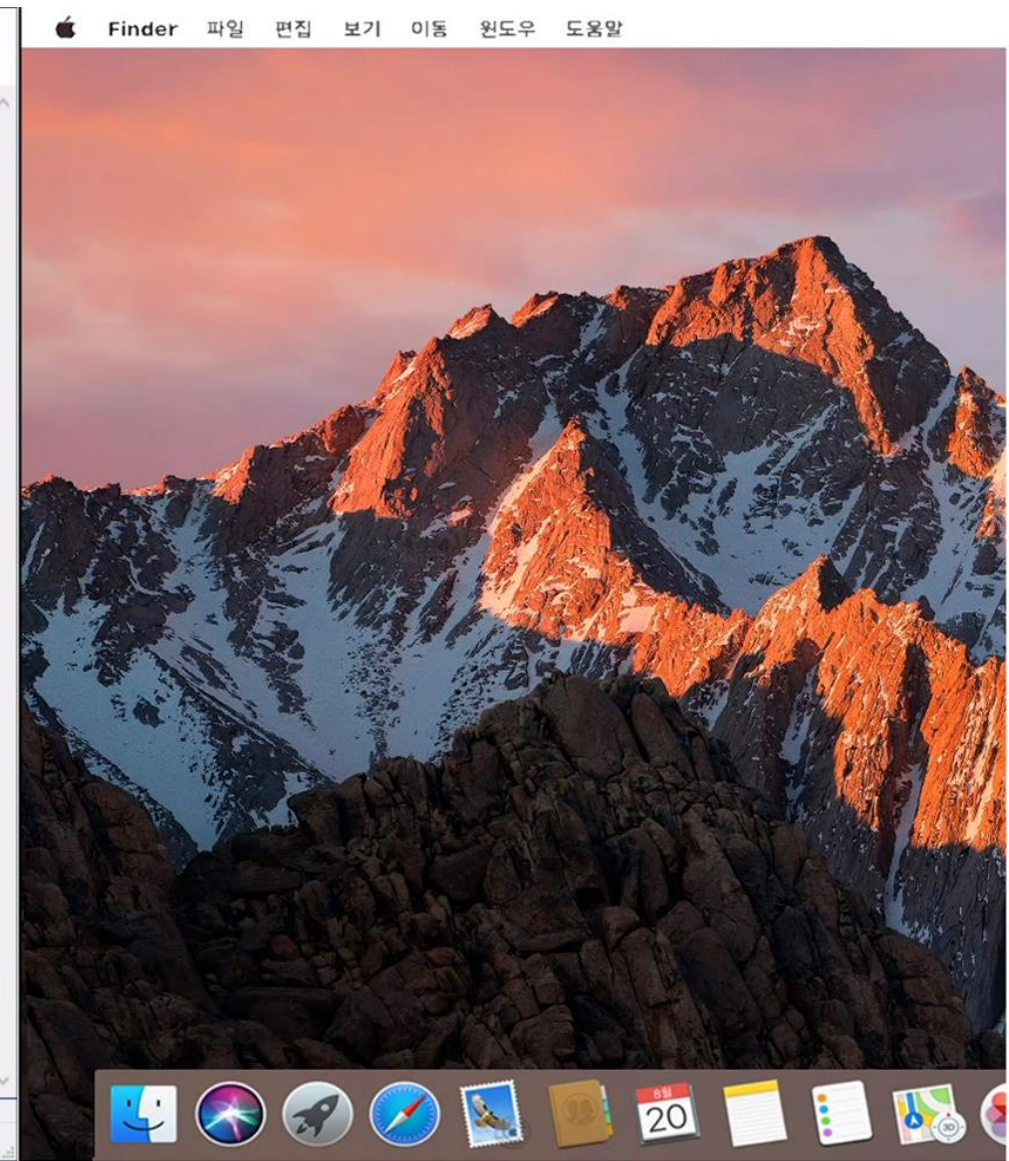
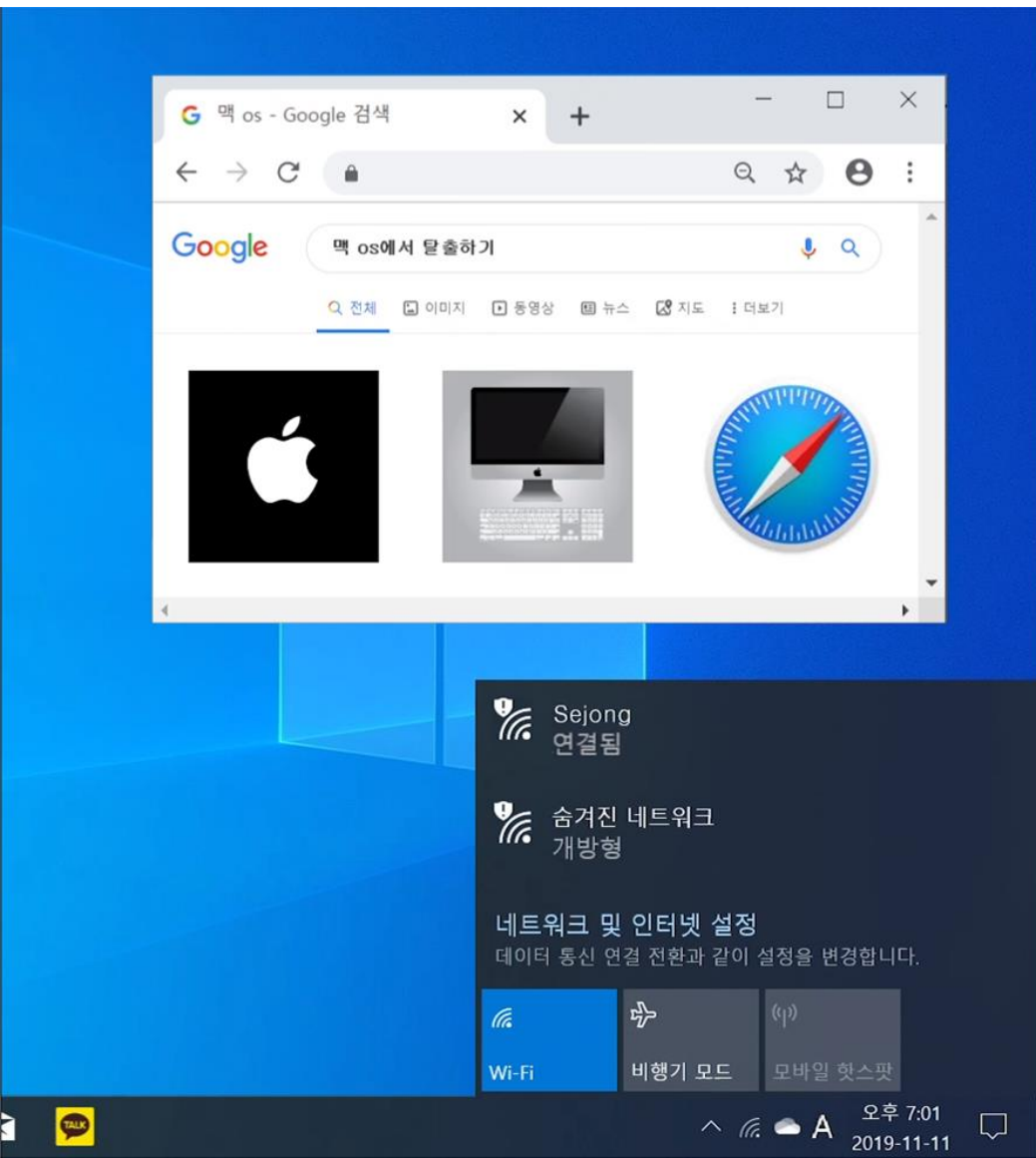
```
this.physics.add.collider(player, ground);  
this.physics.add.collider(player, ground_window);  
this.physics.add.collider(trashes, ground);  
this.physics.add.collider(trashes, ground_window);  
this.physics.add.collider(trashes, ground_window2);
```

```
this.physics.add.overlap(player, trashes, collectStar2, null, this);
```

이미지 충돌 검사 코드

동적인 요소 추가2

• 화면 전환



동적인 요소 추가3

- 화면 흔들림

