

## Commit: Add sprites

### Początek prezentacji; pokaz gotowej gry; wyjaśnienie poruszania się po scenie

#### USTAWIANIE SCENY

1. Ustawić Layout na 2 by 3
2. Przeciągnąć Hierarchy View
3. Ustawić Game View na 16:9
4. Dodać ball do sceny
5. Zresetować ball
6. Zmienić nazwę ball na Player
7. Dodać glasswall
8. Zresetować glasswall
9. glasswall scale y = 2
10. glasswall position x = -8,3
11. Zduplikować glasswall
12. Zmienić nazwę glasswall na Left Wall
13. glasswall (1) position x = 8,3
14. Zmienić nazwę glasswall (1) na Right Wall
15. Right Wall rotation Z = 180
16. Dodać background do sceny
17. Zresetować background
18. background Order in Layer = -1
19. Zmienić nazwę background na Background Sprite
20. Dodać platform do sceny
21. Zresetować platform
22. Zmienić nazwę platform na Platform
23. Platform scale x = 6
24. Platform position y = -4
25. Save Scene
26. Dodać folder Scenes
27. Zapisać jako scene

## Commit: Set scene

### Slajdy o colliderach oraz rigidbody

#### USTAWIENIE COLLIDERÓW ORAZ RIGIDBODY

1. Dodać Rigidbody 2D do Player
2. Uruchomić grę

3. Dodać Circle Collider 2D do Player
4. Frame Selected na Player
5. Disable Background Sprite
6. Zmienić radius w Circle Collider 2D od Player na 0.45
7. Enable Background Sprite
8. Dodać Box Collider 2D do Platform, Left Wall, Right Wall
9. Save Scene
10. Uruchomić grę

### Commit: Set basic colliders and rigidbody

### Slajdy o dodawaniu skryptów oraz podstawowych funkcjach (Start, Update, FixedUpdate)

#### PIERWSZY SKRYPT

1. Dodać skrypt PlayerController do Player
2. Dodać folder Scripts
3. Przenieść PlayerController do Scripts
4. Dodać zmienne oraz Start() do PlayerController

```
private Rigidbody2D rigidbody2D;
private CircleCollider2D collider;
```

```
void Start () {
    rigidbody2D = GetComponent<Rigidbody2D>();
    collider = GetComponent<CircleCollider2D>();
}
```

5. Dodać sterowanie na boki

```
void FixedUpdate()
{
    float moveHorizontal = Input.GetAxis("Horizontal");

    Vector2 movement = new Vector2(moveHorizontal, 0f);

    rigidbody2D.AddForce(movement);
}
```

6. Uruchomić grę
7. Dodać zmienną do PlayerController

```
public float speed = 10f;
```

8. Zmienić

```
rigidbody2D.AddForce(movement * speed);
```

9. Uruchomić grę
10. Pokazać zmienianie wartości Speed w Inspector
11. Dodać do FixedUpdate() w PlayerController

```
if (Input.GetButtonDown("Jump"))
{
    rigidbody2D.AddForce(new Vector2(0, 100 * jumpForce));
}
```

12. Wyjaśnić, że 100, bo grawitacja i masa

13. Dodać zmienną do PlayerController

```
public float jumpForce = 3f;
```

14. Uruchomić grę

15. Save Scene

16. Dodać zmienną do PlayerController

```
private bool isOnGround;
```

17. Dodać na początek FixedUpdate()

```
isOnGround = collider.IsTouchingLayers();
```

18. Zmienić if'a na

```
if (Input.GetButtonDown("Jump") && isOnGround)
```

19. Uruchomić grę – pokazać, że działa, ale jest bug ze ścianami

20. Save Scene

**Commit: Add first script**

**Slajdy o warstwach i tagach**

NAPRAWIENIE SKAKANIA

1. Dodać layer „Platform” do Platform

2. Dodać do PlayerController

```
private int layer;
```

3. Dodać do Start() w PlayerController

```
layer = LayerMask.GetMask("Platform");
```

4. Zmienić na

```
isOnGround = collider.IsTouchingLayers(layer);
```

5. Uruchomić grę

6. Save Scene

**Commit: Fix jumping**

**Slajd o przeszkodach, prefabach**

DODANIE PRZESZKÓD, DODANIE JAKO PREFAB

1. Dodać spike

2. Reset spike

3. Zmiana nazwy spike na Spike

4. Spike position Y = -2.5
5. Dodać Polygon Collider 2D
6. Rozwinąć Points i Paths
7. Size = 1
8. Rozwinąć Element 0
9. Size = 5
10. Frame Selected na Spike
11. Edit Collider
12. Dodać tag „Spike” do Spike
13. Dodać do PlayerController

```
void OnCollisionEnter2D(Collision2D other)
{
    if (other.gameObject.CompareTag("Spike"))
    {
        Destroy(gameObject);
    }
}
```

14. Uruchomić grę
15. Stworzyć folder Prefabs
16. Przeciągnąć Spike do folderu Prefabs
17. Spike prefab position y = 0
18. Platform scale x = 1
19. Przeciągnąć Platform do folderu Prefabs
20. Reset Platform prefab
21. Save Scene

**Commit: Add spike, add prefabs**

## Slajd o poruszaniu kamery

### DODANIE PORSZUANIA KAMERY

1. Dodać nowy GameObject
2. Reset GameObject
3. Zmienić nazwę GameObject na Moving Parts
4. Ustawić Main Camera jako dziecko Moving Parts
5. Dodać CameraMover z poziomu folderu Scripts
6. Dodać CameraMover do Moving Parts
7. Dodać do CameraMover

```
public float speed = 0.75f;
```

8. Dodać

```
void Update () {
    transform.position = new Vector2(transform.position.x,
    transform.position.y - speed * Time.deltaTime);
}
```

9. Uruchomić grę
10. Dodać Left Wall, Right Wall oraz Background Sprite do Moving Parts
11. Uruchomić grę
12. Save Scene

## Commit: Add camera movement

### Slajdy o spawnerze

#### DODANIE SPAWNERA

1. Dodać nowy GameObject do Moving Parts
2. Zmienić nazwę GameObject na Spawner
3. Reset Spawner
4. Ustawić ikonę Spawner na zieloną
5. Spawner position y = -10
6. Dodać nowy GameObject jako dziecko Spawner
7. Zmienić nazwę GameObject na Max Left
8. Ustawić ikonę Max Left na turkusową
9. Reset Max Left
10. Max Left position x = -6.5
11. Zduplicować Max Left
12. Zmienić nazwę Max Left (1) na Max Right
13. Max Right position x = 6.5
14. Dodać skrypt SpawnController do Spawner
15. Przenieść SpawnController do folderu Scripts
16. Dodać zmienne

```
public Transform[] boundaries;
public GameObject platformPrefab;
public GameObject spikePrefab;

public float minScaleX = 0.5f;
public float maxScaleX = 3f;
public float probabilityOfSpike = 0.33f;

public int spikesInRowLimit = 2;

private int spikesInRow;
```

17. Dodać funkcję Spawn()

```
void Spawn()
{
    GameObject objectToSpawn;
```

```

    float scaleX = Random.Range(minScaleX, maxScaleX);
    float scaleY;

    float positionX = Random.Range(boundaries[0].position.x,
boundaries[1].position.x);

    if (spikesInRow < spikesInRowLimit && Random.value <
probabilityOfSpike)
    {
        ++spikesInRow;
        objectToSpawn = Instantiate(spikePrefab);
        scaleY = scaleX;
    }
    else
    {
        objectToSpawn = Instantiate(platformPrefab);
        scaleY = objectToSpawn.transform.localScale.y;
        spikesInRow = 0;
    }

    objectToSpawn.transform.position = new Vector2(positionX,
transform.position.y);

    objectToSpawn.transform.localScale = new Vector2(scaleX, scaleY);
}

```

#### 18. Wywołać funkcję z Start

```

void Start () {
    Spawn();
}

```

#### 19. Dodać MaxLeft i MaxRight do Boundaries

#### 20. Dodać Platform Prefab oraz Spike Prefab

#### 21. Uruchomić grę

#### 22. Save Scene

#### 23. Dodać zmienne

```

public float minDistance = 1f;
public float maxDistance = 5f;

private float positionY;
private float nextSpawnPosition;

```

#### 24. Dodać FixedUpdate

```

void FixedUpdate () {
    positionY = Mathf.Abs(transform.position.y);
    if (positionY >= nextSpawnPosition)
    {
        nextSpawnPosition += Random.Range(minDistance, maxDistance);
        Spawn();
    }
}

```

#### 25. Zmienić Start

```

void Start () {
    nextSpawnPosition = Mathf.Abs(transform.position.y);
}

```

26. Uruchomić grę
27. W Left Wall oraz Right Wall ustawić Order in Layer = 1
28. Save Scene

**Commit: Add spawn controller**

**Slajdy nt. borderów**

## DODANIE BORDERÓW

1. Dodać nowy GameObject jako dziecko Moving Parts
2. Zmienić nazwę GameObject na Borders
3. Reset Borders
4. Dodać nowy GameObject jako dziecko Borders
5. Zmienić nazwę GameObject na Top Border
6. Reset Top Border
7. Top Border position y = 10
8. Dodać Box Collider 2D do Top Border
9. Box Collider 2D size x = 20 w Top Border
10. Box Collider 2D Is Trigger = true
11. Dodać Rigidbody 2D do Top Border
12. Rigidbody 2D Body Type = Kinematic na Top Border
13. Wyjaśnić po co Rigidbody 2D
14. Dodać skrypt TopBorderController do TopBorder
15. Przenieść TopBorderController do Scripts
16. Dodać do TopBorderController

```
private void OnTriggerEnter2D(Collider2D other)
{
    if (!other.gameObject.CompareTag("Wall"))
    {
        Destroy(other.gameObject);
    }
}
```

17. Zduplikować Top Border
18. Zmienić nazwę Top Border (1) na Bottom Border
19. Bottom Border position y = -7
20. Usunąć TopBorderController z Bottom Border
21. Usunąć Rigidbody 2D z Bottom Border
22. Dodać BottomBorderController do Bottom Border
23. Przenieść BottomBorderController do folderu Scripts
24. Dodać do BottomBorderController

```
private void OnTriggerEnter2D(Collider2D other)
{
    if (other.gameObject.CompareTag("Player"))
    {
        Destroy(other.gameObject);
    }
}
```

25. Pokazać użycie konsoli
26. Dodać tag „Player” do Player
27. Dodać tag „Wall” do Left Wall oraz Right Wall
28. Uruchomić grę
29. Save Scene

### Commit: Add borders

### Slajdy nt. tła i jak będzie scrollowane

#### DODANIE SCROLLOWANEGO TŁA

1. Dodać nowy Quad jako dziecko Moving Parts
2. Zmienić nazwę Quad na Background
3. Reset Background
4. Usunąć Mesh Collider z Background
5. Background position z = 3
6. Background scale x = 20, y = 32
7. Usunąć Background Sprite
8. Przeciągnąć background (z folderu Sprites) na Background (w Scene View)
9. Background Shader = Sprites/Default
10. Zduplikować Background
11. Ustawić Background (1) jako dziecko Background
12. Ustawić Background (1) za Background (za pomocą V)
13. Dodać skrypt BackgroundScroller do Background
14. Przenieść BackgroundScroller do folderu Scripts
15. Stworzyć zmienne w BackgroundScroller

```
public float scrollingSpeed = 0.75f;

private float lenght;
private Vector3 startingPosition;
```

16. Dodać do BackgroundScroller

```
void Start ()
{
    lenght = transform.localScale.y / 2;
```



```
startingPosition = transform.localPosition;  
}
```

## 17. Dodać do BackgroundScroller

```
void Update ()  
{  
    float newPosition = Mathf.Repeat(Time.time * scrollingSpeed,  
length);  
  
    transform.localPosition = startingPosition + Vector3.up *  
newPosition;  
}
```

## 18. Uruchomić grę

## 19. Save Scene

## Commit: Add scrolling background

## Slajdy nt. UI

### DODNIE UI I GAME CONTROLLER

1. Dodać nowy UI Text
2. Zmienić nazwę Text na Score
3. Reset Score
4. Ustawić placeholder Score na „Score: 0 m”
5. Ustawić wszystko na stretch top
6. Font Size = 40
7. Height = 50
8. Left = 60
9. Pos y = -10
10. Color = 420798FF
11. Zduplikować Score
12. Zmienić nazwę Score (1) na Restart
13. Ustawić wszystko na stretch middle
14. Ustawić Alignment na center
15. Color = 890000FF
16. Ustawić placeholder Restart na „Press ‘R’ to restart”
17. Disable Restart
18. Dodać nowy GameObject
19. Zmienić nazwę GameObject na Game Controller
20. Reset Game Controller
21. Dodać skrypt GameController do Game Controller
22. Przeciągnąć GameController do folderu Scripts
23. Dodać zmienne do GameController

```
public Text score;
public Text restart;

private bool isRestartEnabled;
```

24. Przeciągnąć score oraz restart na odpowiednie pola w Game Controller

25. Dodać do GameController

```
public void UpdateScore(float y)
{
    score.text = "Score: " + Mathf.RoundToInt(Mathf.Abs(y)) + " m";
}
```

26. Dodać do PlayerController

```
private GameController gameController;
```

27. Dodać do Start w PlayerController

```
gameController =
GameObject.FindWithTag("GameController").GetComponent<GameController>
();
```

28. Dodać tag „GameController” do GameController

29. Dodać do PlayerController

```
void Update()
{
    gameController.UpdateScore(transform.position.y);
}
```

30. Dodać do GameController

```
public void EnableRestart()
{
    isRestartEnabled = true;
    restart.gameObject.SetActive(true);
}
```

31. Dodać do PlayerController

```
void OnDestroy()
{
    gameController.EnableRestart();
}
```

32. Dodać do GameController

```
void Update()
{
    if (isRestartEnabled && Input.GetButtonDown("Restart"))
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
```

33. Ustawić przycisk 'R' na restart Edit -> Project Settings -> Input -> Axes

34. Size = Size + 1

35. Zmienić nazwę Cancel na Restart

36. Positive Button = r

37. Usunąć Alt Positive Button

38. Dodać początkowe platformy i spike

- 39. Uruchomić grę
- 40. Dostosować wartości gameplay'owe (np. spawner)
- 41. Save Scene

**Commit: Add UI, game controller, polish gameplay**