



REPÚBLICA DE CHILE
ILUSTRE MUNICIPALIDAD DE CONCÓN
DIRECCIÓN DE CULTURA

ORD. N° 341 /2024.-

ANT. :

MAT. Informe y boleta de
Mauricio Jaramillo.

CONCÓN, 25 de junio de 2024

DE: SRTA. MARCELA SANTIBÁÑEZ TORRES
DIRECTORA DE CULTURA

A: SRTA. VICTORIA VALENCIA VÉLIZ
JEFA DE CONTABILIDAD Y PRESUPUESTO

A través del presente, luego de saludar a usted, adjunto remito Informe Mensual y Boleta de Honorario mes de junio del siguiente funcionario:

NOMBRE	N° BOLETA	N° DECRETO
MAURICIO ANTONIO JARAMILLO BECERRA	28	0388 /25-01-2024

Se deja constancia que se han recepcionado conformes los trabajos realizados y se autoriza el pago correspondiente.

Sin otro particular, saluda atentamente a usted,



MARCELA SANTIBÁÑEZ TORRES
DIRECTORA DE CULTURA

MST/atb

Distribución:

- 1.- Finanzas
- 2.- RRHH
- 3.- Dirección de Cultura

Fecha: 01 de julio de 2024

**MAURICIO ANTONIO
JARAMILLO BECERRA**

Rut: [REDACTED]

Giro(s):

OTRAS ACTIVIDADES CREATIVAS,
ARTISTICAS Y DE ENTRETENIMIENTO N.C.P.

Dirección:

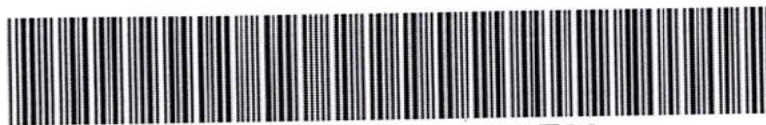
**Señor(es): I MUNICIPALIDAD
DE CONCON**

Rut: 73.568.600-3



Por atencion profesional	Monto
Serv Técnico en soporte realidad virtual para Prog Centro Innovación Tecnologica mes JUNIO2024	897.827
Total Honorario \$	897.827
13,75% Impto. retenido	123.451
Total	774.376

El contribuyente receptor de esta boleta debe retener el 13,75%



19185345000001B41E89

Res. Ex. N° 83 de 30/08/2004
Verifique este documento en www.sii.cl

"Certifico la recepción conforme de todos los Bienes, Servicios y Honorarios descritos en el presente documento tributario, dando cumplimiento a cabalidad lo solicitado, procediendo su pago respectivo".

Fecha: 28/6/24 Nombre, Firma y Timbre Responsable

Norelia Santibonéz

Programa de Servicio Técnico en Soporte de Realidad Virtual para Centro Innovacion Tecnológica Cultural Citec JUNIO 2024

INFORME DE GESTIÓN MES JUNIO 2024

Nombre: Mauricio Antonio Jaramillo Becerra

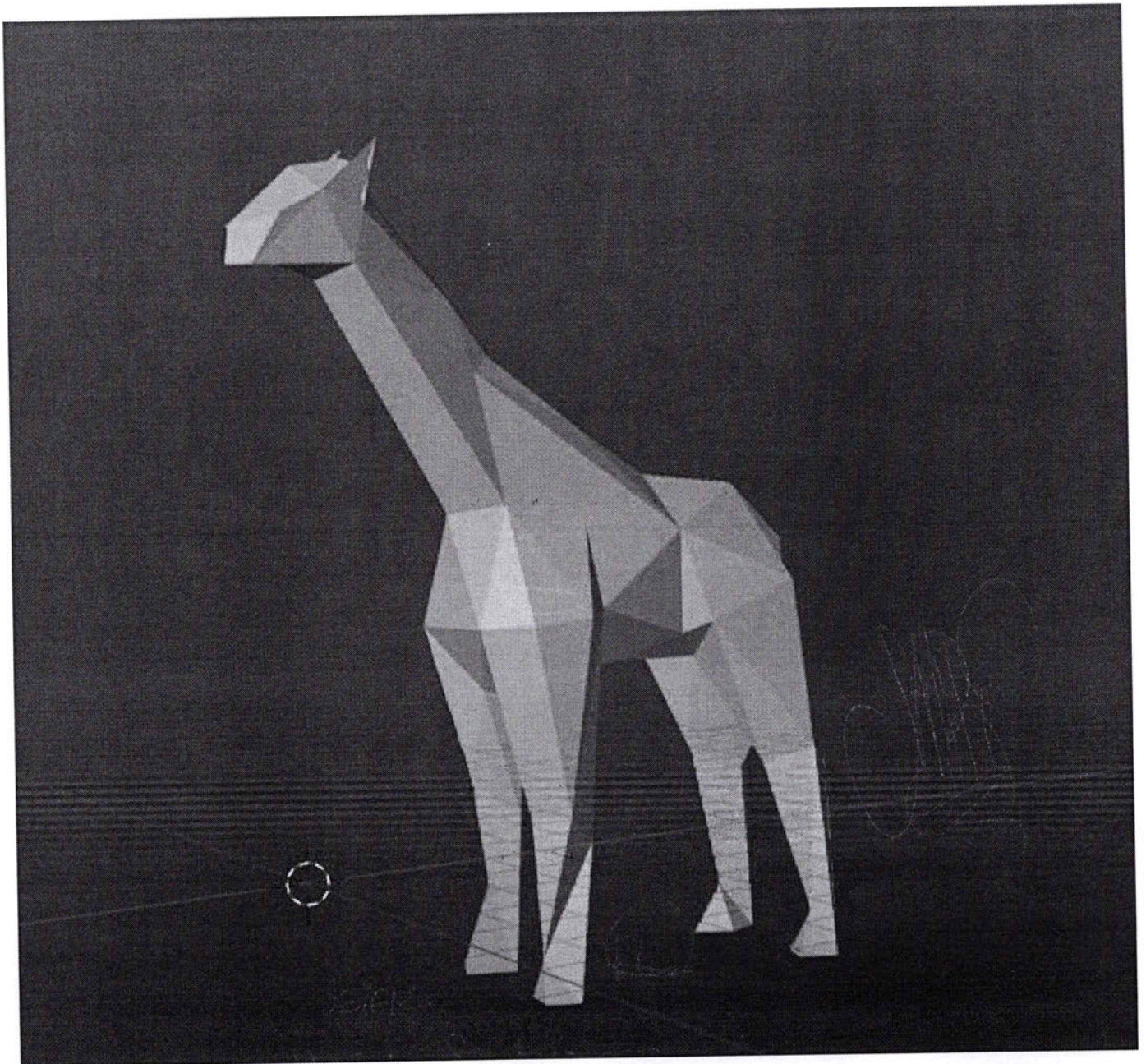
Cargo: Técnico Soporte Vr , Programa Centro de Innovación Tecnológica Cultural y Social, I.Municipalidad de Concón

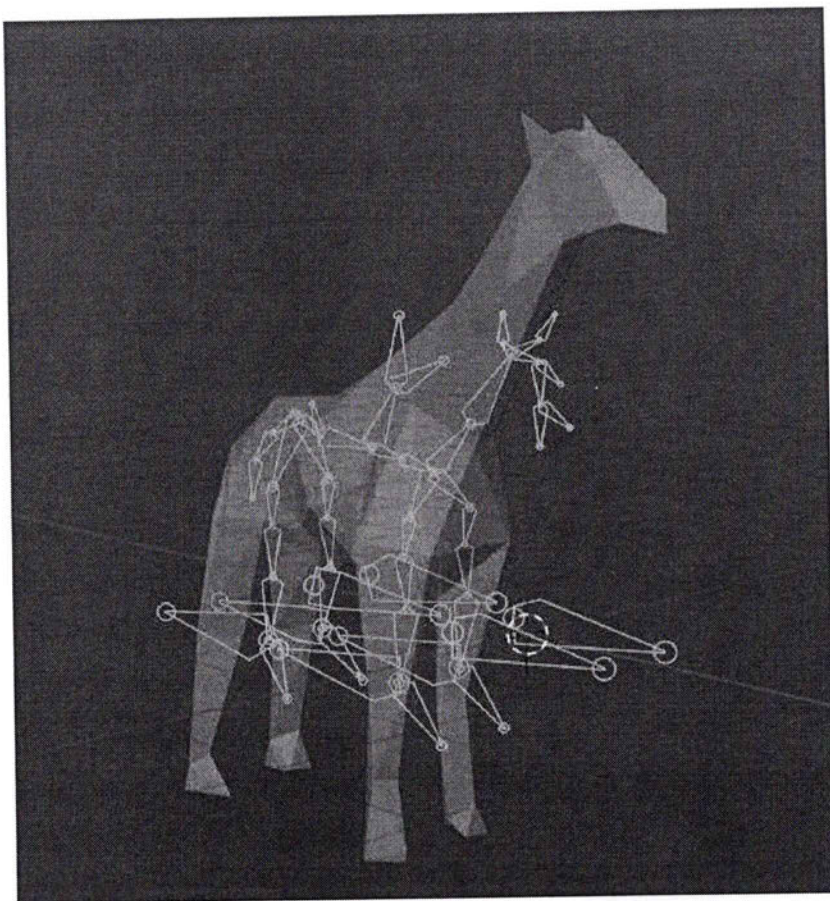
I. -Informe de actividades

Semana del 1 al 2 de Junio (SABADO-DOMINGO)

Semana del 3 al 7 de Junio

- Se comienza a trabajar en la integración de un personaje extra dentro del juego, la idea es generar un modelo 3D de un guanaco version lowpoly con el cual el niño bato pueda interactuar, en conjunto con la dinámica de recolección.
 - Se trabaja en el modelo 3D, esto presenta dificultades ya que, al momento de generar el modelo también hay que generar un rig del mismo, y generar las animaciones de forma individual.
 - Se comienza a trabajar en el código de este nuevo personaje utilizando Visual Studio.
- * Para esta versión del juego se esta usando el motor gráfico UNITY*





```

1 using UnityEngine;
2
3 public class CharacterMovement : MonoBehaviour
4 {
5     public float moveSpeed = 5f;
6     public float turnSpeed = 360f;
7
8     private Animator animator;
9     private Rigidbody rb;
10
11     void Start()
12     {
13         animator = GetComponent();
14         rb = GetComponent();
15     }
16
17     void Update()
18     {
19         float move = Input.GetAxis("Vertical");
20         float turn = Input.GetAxis("Horizontal");
21
22         Vector3 moveDirection = transform.forward * move * moveSpeed * Time.deltaTime;
23         rb.MovePosition(rb.position + moveDirection);
24
25         float turnAmount = turn * turnSpeed * Time.deltaTime;
26         Quaternion turnOffset = Quaternion.Euler(0, turnAmount, 0);
27         rb.MoveRotation(rb.rotation * turnOffset);
28
29         if (move != 0)
30         {
31             animator.SetBool("isWalking", true);
32         }
33         else
34         {
35             animator.SetBool("isWalking", false);
36         }
37     }
38 }

```

```

1 using UnityEngine;
2 using System.Collections.Generic;
3
4 public class Movement : MonoBehaviour
5 {
6     public float moveSpeed;
7     public float turnSpeed;
8     public float speedThreshold;
9     private Animator animator;
10     private int currentDirection = 0;
11
12     void Start()
13     {
14         animator = GetComponent();
15     }
16
17     void Update()
18     {
19         MoveAndTurn();
20     }
21
22     void MoveAndTurn()
23     {
24         if (Input.GetAxis("Vertical") != 0)
25         {
26             Transform targetPosition = animator.transform.position;
27             Vector3 direction = targetPosition - transform.position;
28             float distanceToTarget = Vector3.Distance(transform.position, targetPosition);
29
30             if (distanceToTarget > speedThreshold)
31             {
32                 Vector3 moveDirection = direction.normalized * moveSpeed * Time.deltaTime;
33                 transform.position += moveDirection;
34
35                 Quaternion targetRotation = Quaternion.LookRotation(direction);
36                 transform.rotation = Quaternion.RotateTowards(transform.rotation, targetRotation, turnSpeed * Time.deltaTime);
37
38                 animator.SetBool("isWalking", true);
39             }
40         }
41         else
42         {
43             transform.position = transform.position + 0.1f * moveSpeed * Time.deltaTime;
44             animator.SetBool("isWalking", false);
45         }
46     }
47 }

```


Semana del 10 al 14 de Junio

- Se trabaja en generar una inteligencia artificial que actúe dentro del mismo juego para comandar acciones de otros personajes incluidos.
- Se continúa generando iteraciones de código (un escape básico, un escape con mayor aceleración y un escape aleatorio donde los parámetros se alteran dependiendo del entorno) para generar una dinámica de juego de las escondidas en el cual el modelo animado del guanaco debería actuar como un NPC (NON PLAYABLE CHARACTER) controlado por una inteligencia artificial dentro del juego que le haga escapar cuando el personaje niño bato se acerca al radio establecido.

```
1 using UnityEngine;
2
3 public class NPCEscape1 : MonoBehaviour
4 {
5     public Transform player;
6     public float detectionRange = 10f;
7     public float escapeSpeed = 5f;
8     public float turnSpeed = 300f;
9     public float escapeDistance = 15f;
10
11     private Animator animator;
12     private Rigidbody rb;
13
14     void Start()
15     {
16         animator = GetComponent<Animator>();
17         rb = GetComponent<Rigidbody>();
18     }
19
20     void Update()
21     {
22         float distanceToPlayer = Vector3.Distance(transform.position, player.position);
23
24         if (distanceToPlayer < detectionRange)
25         {
26             EscapeFromPlayer();
27         }
28         else
29         {
30             animator.SetBool("isWalking", false);
31         }
32     }
33
34     void EscapeFromPlayer()
35     {
36         Vector3 directionFromPlayer = (transform.position - player.position).normalized;
37         Vector3 escapeTarget = transform.position + directionFromPlayer * escapeDistance;
38         Vector3 directionToTarget = (escapeTarget - transform.position).normalized;
39
40         Vector3 moveDirection = directionToTarget * escapeSpeed * Time.deltaTime;
41         rb.MovePosition(rb.position + moveDirection);
42
43         Quaternion targetRotation = Quaternion.LookRotation(directionToTarget);
44         rb.MoveRotation(Quaternion.RotateTowards(transform.rotation, targetRotation, turnSpeed * Time.deltaTime));
45
46         animator.SetBool("isWalking", true);
47     }
48 }
49
```

```
1 using UnityEngine;
2
3 public class NPCEscape2 : MonoBehaviour
4 {
5     public Transform player;
6     public float detectionRange = 10f;
7     public float escapeSpeed = 4f;
8     public float turnSpeed = 300f;
9     public float escapeDistance = 10f;
10
11     private Animator animator;
12     private Rigidbody rb;
13
14     void Start()
15     {
16         animator = GetComponent<Animator>();
17         rb = GetComponent<Rigidbody>();
18     }
19
20     void Update()
21     {
22         float distanceToPlayer = Vector3.Distance(transform.position, player.position);
23
24         if (distanceToPlayer < detectionRange)
25         {
26             EscapeFromPlayer();
27         }
28         else
29         {
30             animator.SetBool("isWalking", false);
31         }
32     }
33
34     void EscapeFromPlayer()
35     {
36         Vector3 directionFromPlayer = (transform.position - player.position).normalized;
37         Vector3 randomDirection = (directionFromPlayer * Random.insideUnitSphere).normalized;
38         Vector3 escapeTarget = transform.position + randomDirection * escapeDistance;
39         Vector3 directionToTarget = (escapeTarget - transform.position).normalized;
40
41         Vector3 moveDirection = directionToTarget * escapeSpeed * Time.deltaTime;
42         rb.MovePosition(rb.position + moveDirection);
43
44         Quaternion targetRotation = Quaternion.LookRotation(directionToTarget);
45         rb.MoveRotation(Quaternion.RotateTowards(transform.rotation, targetRotation, turnSpeed * Time.deltaTime));
46
47         animator.SetBool("isWalking", true);
48     }
49 }

```



```

1 using UnityEngine;
2
3 public class NPCEscaper : MonoBehaviour
4 {
5     public Transform player;
6     public float detectionRange = 5f;
7     public float initialEscapeSpeed = 2f;
8     public float maxEscapeSpeed = 6f;
9     public float acceleration = 3f;
10    public float turnSpeed = 400f;
11    public float escapeDistance = 20f;
12
13    private Animator animator;
14    private Rigidbody rb;
15    private float currentEscapeSpeed;
16
17    void Start()
18    {
19        animator = GetComponent<Animator>();
20        rb = GetComponent<Rigidbody>();
21        currentEscapeSpeed = initialEscapeSpeed;
22    }
23
24    void Update()
25    {
26        float distanceToPlayer = Vector3.Distance(transform.position, player.position);
27
28        if (distanceToPlayer < detectionRange)
29        {
30            EscapeFromPlayer();
31        }
32        else
33        {
34            animator.SetBool("isWalking", false);
35            currentEscapeSpeed = initialEscapeSpeed; // Reset speed when player is not close
36        }
37    }
38
39    void EscapeFromPlayer()
40    {
41        Vector3 directionFromPlayer = (transform.position - player.position).normalized;
42        Vector3 escapeTarget = transform.position + directionFromPlayer * escapeDistance;
43        Vector3 directionToTarget = (escapeTarget - transform.position).normalized;
44
45        currentEscapeSpeed = Mathf.Min(currentEscapeSpeed + acceleration * Time.deltaTime, maxEscapeSpeed);
46
47        Vector3 moveDirection = directionToTarget * currentEscapeSpeed * Time.deltaTime;
48        rb.MovePosition(rb.position + moveDirection);
49
50        Quaternion targetRotation = Quaternion.LookRotation(directionToTarget);
51        rb.MoveRotation(Quaternion.RotateTowards(transform.rotation, targetRotation, turnSpeed * Time.deltaTime));
52
53        animator.SetBool("isWalking", true);
54    }
55 }

```

```

1 using UnityEngine;
2
3 public class NPCController : MonoBehaviour
4 {
5     public Transform player;
6     public float detectionRange = 10f;
7     public float escapeSpeed = 5f;
8     public float turnSpeed = 360f;
9     public float escapeDistance = 15f;
10    public float wanderRadius = 20f;
11    public float wanderTimer = 5f;
12
13    private Animator animator;
14    private Rigidbody rb;
15    private Vector3 wanderTarget;
16    private float timer;
17
18    void Start()
19    {
20        animator = GetComponent<Animator>();
21        rb = GetComponent<Rigidbody>();
22        timer = wanderTimer;
23        SetRandomWanderTarget();
24    }
25
26    void Update()
27    {
28        float distanceToPlayer = Vector3.Distance(transform.position, player.position);
29
30        if (distanceToPlayer < detectionRange)
31        {
32            EscapeFromPlayer();
33        }
34        else
35        {
36            Wander();
37        }
38    }
39
40    void EscapeFromPlayer()
41    {
42        Vector3 directionFromPlayer = (transform.position - player.position).normalized;
43        Vector3 escapeTarget = transform.position + directionFromPlayer * escapeDistance;
44        Vector3 directionToTarget = (escapeTarget - transform.position).normalized;
45
46        MoveAndRotate(directionToTarget, escapeSpeed);
47
48        animator.SetBool("isWalking", true);
49    }
50
51    void Wander()
52    {
53        timer -= Time.deltaTime;
54        if (timer < 0)
55        {
56            SetRandomWanderTarget();
57            timer = wanderTimer;
58        }
59    }
60
61    void SetRandomWanderTarget()
62    {
63        Vector3 wanderPosition = transform.position + Vector3.sphericalRandom(wanderRadius, 0, 0);
64        wanderTarget = wanderPosition;
65    }
66
67    void MoveAndRotate(Vector3 direction, float speed)
68    {
69        Vector3 moveDirection = direction * speed * Time.deltaTime;
70        rb.MovePosition(rb.position + moveDirection);
71
72        Quaternion targetRotation = Quaternion.LookRotation(direction);
73        rb.MoveRotation(Quaternion.RotateTowards(transform.rotation, targetRotation, turnSpeed * Time.deltaTime));
74    }
75 }

```

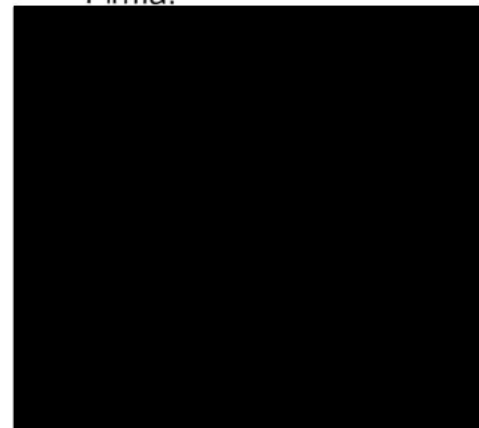
Semana del 17 al 21 de Junio (JUEVES 20 FERIADO)(vacaciones del 18 al 27)

- Se continua en la integración de la Inteligencia Artificial para comandar distintos NPC dentro del juego bato.
- Vacaciones.

Semana del 24 al 28 de Junio (vacaciones hasta el jueves 27)

- Vacaciones.
- Se prestará asistencia en evento de DOM los días Viernes, Sabado y Domingo.

Nombre: Mauricio Antonio Jaramillo Becerra
Cargo: Técnico de Soporte en Realidad Virtual
Institución: Ilustre Municipalidad de Concón
Firma:



A handwritten signature in blue ink, likely belonging to Mauricio Antonio Jaramillo Becerra.