

Tiny Alien

Game Design Document

Written by
David Martinez,
Mario Henao,
Juan Gonzalez

September 11th, 2023

Index

[Index](#)

[Overview](#)

[Elevator Pitch](#)

[Design Pillars](#)

[Crecer](#)

[Comer](#)

[Sobrevivir](#)

[Summary](#)

[Brief Description](#)

[Descripción](#)

[Theme/Setting/Genre](#)

[Main Objective](#)

[Aesthetics/Dynamics/Mechanics](#)

[Influences](#)

[Kirby](#)

[Hole.io](#)

[Slither.io](#)

[Vampire survivors](#)

[What Sets This Game Apart?](#)

[Target Audience](#)

[Target Rating](#)

[Gameplay](#)

[Gameplay Summary](#)

[Brief](#)

[Detailed](#)

[Modes](#)

[Victory Conditions](#)

[Core Game Loop](#)

[- Eat](#)

[- Grow](#)

[- Evade](#)

Core Gameplay Mechanics

Eat

- Details

- Implementation

Evade

- Details

- Implementation

Grow

- Details

- Implementation

Story

Brief Summary

Detailed Outline

Characters

Character #1

Level Design

Unico Nivel

Conveyance Mechanisms

Energía

Comer

Evadir

Crecer

Recibir daño

Cinematics

Technical

Screens

Title Screen

In-Game

Barra de energía

Escenario

Personaje

Enemigos

Objetos comestibles

Botón de pausa

Options Menu

Credits

Game Engine

Controls

Development

Classes

- PlayerController
- EnergyBar
- ObjectController
- EnemyController
- EdibleObject
- SpawnManager

Visual Art

Style

UI/HUD

Assets Needed

Characters

Terrain/Environment

Objects

Promotional Material

Audio

Style

Music

Sound Effects

Project Scope

Time Scale

Anticipated Gameplay Time

Cost

Target Platforms

Team

Juan David Gonzalez Dimaté

Mario Henao

David Martinez

Schedule

Prototipo básico - Martes 12 de Septiembre

Prototipo básico, funciones adicionales - Miércoles 13 de Septiembre

Producto Alfa - Jueves 14 de Septiembre

Producto beta - Viernes 15 de Septiembre - Sábado 16 de Septiembre.

Version History

V1.0 - Inicialización

V1.1 - Fixing Bugs

Overview

Elevator Pitch

¿Qué comen los aliens en este planeta? Pues el propio planeta.

Design Pillars

Crecer

El mundo debe volverse pequeño para que sea coherente con el tema de la Game Jam. Por esto decidimos que lo principal es que el personaje crezca para cumplir con este objetivo.

Comer

Es el medio por el cual el personaje debe crecer. Conforme más crece, más cosas disponibles tiene para comer.

Sobrevivir

El personaje debe tener los medios para cumplir con el objetivo de crecer, y por esto hay ciertas mecánicas como la interacción con los enemigos y/o sus propias habilidades que lo facilitan o lo impiden.

Summary

Resumen general

Brief Description

Un alien pequeño llegó al planeta tierra luego de que un satélite enviado por los humanos se chocó contra su casa justo antes de la hora de la cena, por lo que quiere ir al planeta para que se lo compensen.

Descripción

El alien cayó a la tierra después de mucho tiempo, y como no había cenado empezó a comer todo lo que se le pasara por el frente. En un principio, solo absorbía la comida que estuviera cerca suyo, pero como su especie crece cada vez que come algo que no sea de su propio planeta, empezó a sentirse cada vez más hambriento, por lo que empezó a comer personas y animales, luego vehículos, y luego edificios.

Mientras tanto, los científicos de la NASA detectaron un agujero negro en miniatura en una ciudad cercana, por lo que dieron la orden de destruirlo lo más rápido posible, antes de que sea demasiado tarde.

Theme/Setting/Genre

Top-down en 3D, con elementos de survival y recolección de recursos. Transcurre en una ciudad cercana al centro de investigación de la NASA, lo que facilita la identificación y tratamiento de la amenaza.

Comic/cartoonie

Main Objective

El personaje principal debe comer y crecer hasta ser capaz de comerse al mundo.

Aesthetics/Dynamics/Mechanics

Este es un juego de supervivencia con elementos roguelike en donde el jugador debe apoyar al personaje principal a no sentir más hambre. El jugador debe sentir que el mundo está en su contra y que la única manera de revertir la situación es comiéndose a todo el mundo. Entre más coma el personaje, más va creciendo y más cosas puede comer. Cada vez que coma, obtendrá energía, la cual afecta a su tamaño (entre más energía tenga, más grande será). En un principio, el personaje será tan pequeño como un conejo, pero crecerá conforme absorba elementos en el escenario. El obstáculo principal son los misiles que se despacharon para acabar con él. Cada misil reduce la energía que acumula, y por lo tanto, su tamaño. El jugador pierde cuando su energía cae a 0. Para evitar este escenario, el personaje podrá gastar parte de su energía haciendo movimientos rápidos y explosivos que le permitan esquivar esas amenazas. Conforme crezca el personaje, podrá absorber más objetos del escenario. Cuando el jugador llegue al tope de energía, podrá devorar al mundo entero.

Influences

Kirby

- Como el personaje de la franquicia de Kirby se la pasa comiendo cosas como su principal mecánica de juego, tomamos inspiración en este personaje para nuestro jugador.

Hole.io

- En este juego, el jugador debe mover un agujero negro que se va haciendo más grande conforme absorbe cosas del escenario. Consideramos que este concepto está muy relacionado con la temática de la Game Jam.

Slither.io

- Similar a lo que sucede en Hole.io, el jugador va creciendo conforme absorbe las partículas del escenario. En este caso nos enfocamos en la mecánica en la cual el jugador puede usar un movimiento especial (el “turbo”) a costa de parte de su longitud total.

Vampire survivors

- Tomamos como inspiración la perspectiva de este juego y la mecánica sobre la cual se generan de manera aleatoria los elementos en el escenario, tanto enemigos como objetos consumibles.

What Sets This Game Apart?

- El juego hace sentir al jugador que todo lo que no sea comida es un enemigo, lo que otorga una perspectiva única.
- El jugador no solo debe preocuparse por comer y crecer, sino que también debe asegurarse de gestionar bien el recurso que estas acciones generan, ya que habrá veces en las que tendrá que decidir entre comer o sobrevivir.

Target Audience

Debido a que el juego maneja una estética simple y no contiene temas muy oscuros o que puedan perturbar al espectador, el público objetivo está orientado a personas de todas las edades.

Target Rating

El juego lo pueden jugar personas de todas las edades, ya que no tiene contenido sexual, violento u oscuro.

Github Repository

<https://github.com/jdgonzalezd1/TinyAlien>

Gameplay

Gameplay Summary

Brief

El jugador debe comer elementos en el escenario con el fin de crecer, y cada vez que come algo, aumenta la energía que posee. Entre más energía tenga, más grande será. Los obstáculos en el escenario son los misiles que se dirigen al jugador. El jugador puede utilizar la energía acumulada para evadir los proyectiles. El juego termina cuando: el jugador almacena la energía máxima o su energía cae a 0.

Detailed

El jugador empieza como un objeto de un tamaño aproximado al de un conejo en el escenario. En el escenario se puede encontrar comida y objetos pequeños que el jugador puede consumir. El jugador empieza con una cantidad fija de energía, la cual es el recurso principal del juego. Cada objeto consumido otorga energía. Entre más energía tenga el jugador, más crecerá. Los obstáculos que tiene que evitar el jugador son los misiles teledirigidos de la NASA. El jugador puede utilizar una cierta cantidad de energía para utilizar un “dash”, lo que le facilita esquivar ataques enemigos. Cuando el jugador adquiere suficiente energía, evoluciona a un tamaño mediano el cual le permite absorber más objetos en el escenario y aumenta la frecuencia con la cual se lanzan los misiles. Cuando el jugador adquiere suficiente energía en su tamaño mediano, evoluciona a un tamaño gigante el cual le permite absorber a las estructuras de la ciudad. Se aumenta la frecuencia de lanzamiento de los misiles aún más. El juego termina si al jugador en su etapa pequeña se le reduce la energía a 0.

Modes

Hay solo un modo de juego.

Victory Conditions

El jugador obtiene la victoria en la partida cuando llega al límite máximo de la barra de energía.

Core Game Loop

- Eat
- Grow
- Evade

Core Gameplay Mechanics

Eat

- **Details**
 - El jugador se acerca a un objeto y lo absorbe oprimiendo una tecla.
- **Implementation**
 - Cuando el jugador se acerca a un objeto puede presionar una tecla que desencadene el evento de comerse a ese objeto.

Evade

- **Details**
 - El jugador realiza un “dash” hacia donde se esté moviendo oprimiendo una tecla.
- **Implementation**
 - Cuando el jugador requiera evadir a un enemigo o un ataque, podrá moverse a una dirección y si oprime la tecla de evasión, realizará un “dash” hacia esa dirección.

Grow

- **Details**
 - El jugador aumentará su tamaño cada vez que absorba un objeto.
- **Implementation**
 - Cuando el jugador absorba un objeto, su barra de energía crecerá, y cada vez que su barra de energía crezca, se desencadenará un evento en el cual el jugador crece una determinada cantidad.

Story

Brief Summary

Un alienígena estaba a punto de cenar cuando un satélite cae en su casa. El alien rastrea al satélite y se dirige a su punto de origen para que los responsables le compensen su cena. Luego de un viaje muy largo, el alienígena llega a la Tierra con mucha hambre, así que decide comer todo lo que se le atraviese.

Detailed Outline

En una galaxia lejana, existe un planeta con una especie un tanto particular. Esta especie posee un micro agujero negro en su boca, lo que provoca que tengan un hambre atroz. Afortunadamente, las frutas que crecen en su planeta son lo suficientemente nutritivas como para saciar su hambre por lo menos 1 mes espacial (50 años terrestres), pero es complicado obtenerlas de manera constante. Un alien de esta especie particular logró obtener una de estas frutas y se estaba preparando para cenar en su hogar, cuando de repente un satélite se estrelló sobre su casa. El choque destruyó la fruta que tanto le costó conseguir, por lo que decidió rastrear el origen del objeto y solicitarle a la persona que lo lanzó una compensación. Luego de un viaje de 15 días espaciales, el alienígena llegó a la Tierra, pero el no haber cenado en tanto tiempo hizo que se estrellara cerca de una ciudad. Ahora, con un hambre atroz, decide que la compensación por su cena será todo lo que pueda comer.

Characters

Character #1

Tiny Alien

- Un alienígena de un planeta lejano. Busca saciar su hambre devorando todo lo que esté a su paso. Cada vez que come, crece, permitiéndole comer cosas cada vez más grandes.

Level Design

Unico Nivel

- Goal: Llenar la barra de energía, con todo lo que se pueda comer antes de que los misiles le quiten toda la energía.
- Mood/Theme: Es un juego casual, con estilo de low poly
- Objects/Enemies/NPCs/etc: Comestibles(Hamburguesas, periódicos, pizzas,arbustos, carros, edificios) Enemigos son los Missiles

Conveyance Mechanisms

Energía

Es el recurso principal del juego. Tiene la forma de una barra de vida, en donde se muestra el recurso que tiene el jugador cada vez que come o que evade ataques. Está dividido en tres segmentos, en donde cada segmento representa la etapa en la que el jugador se encuentra. La barra de energía se llena cada vez que el jugador se come un objeto y se reduce cada vez que utiliza su evasión. El primer segmento de la barra indica que el jugador se encuentra en la etapa pequeña, lo que define los objetos que puede comer y los objetos que lo pueden dañar. Cuando la barra se llena y pasa de un segmento a otro, el jugador “evoluciona” y pasa a la etapa mediana, lo que altera los objetos con los que interactúa, y lo mismo cuando el jugador pasa al último segmento.

Comer

Es la acción principal del juego. El jugador puede comer objetos al acercarse a ellos, y cada vez que lo hace, la barra de energía crece.

Evadir

Es una acción secundaria que puede ejecutar el jugador. El jugador puede hacer un “dash” para evadir enemigos. Esta acción gasta energía.

Creecer

Es la mecánica principal del juego. El tamaño del jugador es proporcional a su barra de energía.

Recibir daño

Es el obstáculo principal del juego. Cada vez que el jugador recibe daño, se reduce su barra de energía. Las condición de derrota es si la energía llega a 0.

Cinematics

No hay cinemáticas en el juego.

Technical

Screens

Title Screen

Pantalla de inicio. Contiene:

- Nombre del juego.
- Escenario de fondo.
- Botón de play.
- Botón de créditos.
- Botón de como jugar.

In-Game

Escenario del juego. Contiene:

Barra de energía

- Barra de vida dividida en 3 segmentos.

Escenario

- Ciudad con un parque.

Personaje

- Personaje principal con una cámara top down enfocándose en él.

Enemigos

- Misiles.

Objetos comestibles

- Hamburguesas, pizza, donas, basura, botes de basura, contenedores de basura, hidrantes, postes de luz, vehículos y edificios.

Botón de pausa

- Botón de pausa en la esquina superior derecha. Muestra el menú de opciones.

Options Menu

Muestra las opciones de reanudar y salir.

Credits

Dependiendo de si el jugador gana o pierde aparece una imagen.

Game Engine

El Game Engine es Unity debido a su facilidad de uso y flexibilidad al momento de desarrollar un proyecto.

Controls

Los controles del juego soportan únicamente entradas de teclado. El movimiento por defecto será con las teclas WASD. y la barra espaciadora que se utiliza para realizar los dash para poder esquivar a los enemigos(Misiles).

Tecla W

- Mueve el personaje hacia adelante.

Tecla A

- Mueve el personaje hacia la izquierda.

Tecla S

- Mueve el personaje hacia atrás.

Tecla D

- Mueve el personaje hacia la derecha.

Barra espaciadora

- Realiza la evasión hacia la dirección a la que el jugador esté apuntando.

Development

Classes

- **PlayerController**

- PlayerController.cs
 - Attributes:
 - Float: energy
 - Int: Stage
 - Transform: Size
 -
 - Methods:
 - ModifySize
 - AbsorbObject
 - TakeDamage
 - ModifyStage
 - Movement
 - Dash
- CameraController.cs
 - Attributes:
 - PlayerController player;
 - CinemachineVirtualCamera virtualCamera;
 - CinemachineFramingTransposer transposer;
 - float scaleChange;
 - Methods:
 -

- **EnergyBar**

- EnergyBar.cs
 - References:
 - Player: Energy
 - GameObject winScreen
 - GameObject loseScreen
 - Methods:
 - DisplayEnergy
 - WinCondition
 - SetEnergy

- ObjectController

- ObjectController.cs
 - Attributes:
 - public int energyAmount;
 - PlayerController player;
 - References:
 - Player: stage
 - Methods:
 - CheckStage

- EnemyController

- EnemyController.cs
 - Attributes:
 - int damageTaken;
 - Controller playerController;
 - GameObject player;
 - NavMeshAgent agent;
 - AudioSource audioExplosion;
 - AudioClip explosion;
 - Methods:
 - Movement
 - DealDamage
 - Attack
 - DamageDealt

- EdibleObject

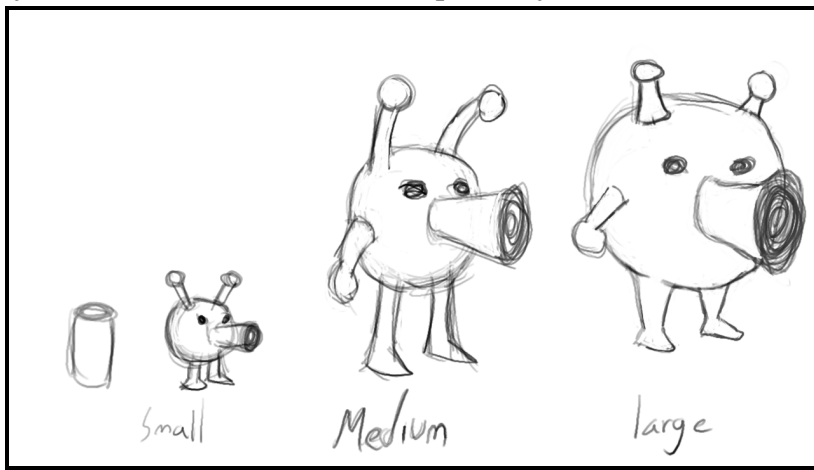
- SpawnManager

- SpawnManager.cs
 - Attributes:
 - Vector3 spawnStreet;
 - public int stageEnable;
 - public float spawnPosX, spawnPosZ;
 - public float spawnInterval = 4.0f;
 - float startDelay = 1.0f;
 - Methods:
 - SpawnMissile

Visual Art

Style

El juego va a ser en 3d con una cámara top-down, es un juego casual que se puede terminar en 5 o 10 minutos, el juego es en tercera persona con una cámara de vista aerea donde se va alejando con base en el tamaño del personaje.



UI/HUD

Está la barra de energía o vida, que muestra el progreso del personaje, está la

- UI de pausa donde el jugador puede reanudar o volver al menú principal
- las UI de victoria o derrota en el juego principal
- El menú principal donde el jugador puede pensar el juego, ver los créditos usando un botón y otro de como jugar yendo allí usando un botón (Mostrado los controladores).

<Describe elements such as the user interface, heads up display, and quest selection menu if applicable. Note functionality, options, animations, etc. List out all widgets and their functionality. Ideally this should include concept art, flowcharts, and/or wireframes.>

Assets Needed

El formato del archivo es .FBX, el nombre de los archivos van a ser:
nameAsset_Small/Medium/Large, el programa con el que van a ser creados es Blender

1. Small
 - Pizza
 - Can
 - Newspaper
 - Hamburger
2. Medium
 - People
 - Bins
 - Mailboxes
 - Containers
3. Large
 - Cars
 - Building
 - Lamp post

Characters

- Aline (Idle, walk, eat)
- Missile

Terrain/Environment

- City
- Park in the city

Objects

1. Small
 - Pizza
 - Can
 - Newspaper
 - Hamburger
2. Medium
 - People
 - Bins
 - Mailboxes
 - Containers
3. Large
 - Cars
 - Building
 - Lamp post

Audio

Style

El estilo de la música debe tener elementos de jazz que incluyan el sentimiento de estar dentro de una ciudad, con un ambiente urbano. El estilo que queremos lograr es muy similar a la canción de fondo de Castelia City en Pokemon Black and White.

Se busca enfatizar el ambiente urbano de una ciudad, ya que es el principal escenario. También se busca que conforme el jugador crece, la música haga una transición entre la alegre ciudad y la destrucción que él está provocando.

Music

- Cityscape

Sound Effects

- Explosion para cuando el personaje es alcanzado por un misil
- Succión cuando el personaje come algo.

Project Scope

Time Scale

El proyecto debe estar finalizado para el viernes 15 de septiembre de 2023, con un posible retraso para el sábado 16 de septiembre, con el fin de lanzarlo ese mismo día y recibir la retroalimentación correspondiente.

Anticipated Gameplay Time

El tiempo de una sola partida debe ser de 5 a 10 minutos.

Target Platforms

En juego será lanzado en la plataforma de Itch.io en formato Web.

Team

Juan David Gonzalez Dimaté

- Game Designer: \$2'560.000

Mario Henao

- Technical Artist: \$2'480.000

David Martinez

- Lead Developer: \$2'360.000

Schedule

Prototipo básico - Martes 12 de Septiembre

1. Prototipo de escenario 1.
2. Prototipo de jugador. - Done
3. Prototipo de objetos en escenario.
4. Prototipo funciones básicas: -Done
 - a. Movimiento. -Done
 - b. Dash. -Done
 - c. Comer. -Done

Prototipo básico, funciones adicionales - Miércoles 13 de Septiembre

1. Prototipo de enemigos.
2. Prototipo funciones de enemigos.
3. Prototipo funciones avanzadas:
 - a. Crecer. -Done
 - b. Spawn de enemigos.
 - c. Barra de energía.
 - d. Patrulla enemigos.
 - e. Interacción con objetos.

Producto Alfa - Jueves 14 de Septiembre

1. Incluir estilo de arte en el escenario.
2. Incluir soundtrack.
3. Pruebas técnicas.
4. Pruebas de jugabilidad.
5. Arreglar Bugs

Producto beta - Viernes 15 de Septiembre - Sábado 16 de Septiembre.

1. Pruebas técnicas.
2. Pruebas de jugabilidad.
3. Lanzamiento.

Version History

V1.0 - Inicialización

- Se inició a trabajar en el documento, se definieron el diseño general del juego y las mecánicas principales. Se definió una idea general del audio y el arte. Se espera complementar el apartado de audio y arte en una futura modificación.

V1.1 - Fixing Bugs

- Añadimos audio para los enemigos y cuando el personaje se come algo, también la música para el inicio del juego, cambiamos el lugar de la cámara para que los jugadores vean más el escenario.