



ugr

Universidad
de Granada

PROGRAMACIÓN GRÁFICA DE VIDEOJUEGOS

Game Design Document

Autores

Bailón Robles, Rafael
Bowen, Kieran Michael
Martín Jáimez, José Adrián
Urrea Echeverry, Juan Carlos
Vendrell Molina, Rafael



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍAS INFORMÁTICA Y DE TELECOMUNICACIÓN

Granada, 10 de noviembre de 2019

Índice

1. Ideas Iniciales	3
1.1. Temática	3
1.2. Mecánicas del Juego	3
1.2.1. Acciones	4
1.2.2. Interacciones	5
1.2.3. Recursos	5
1.2.4. Verbos	5
2. Entidades y Atributos	7
2.1. Descripción	7
2.2. Estados S-T	14
3. Acciones, Interacciones y Reglas	16
3.1. Acciones	16
3.1.1. Pop-Up Actions	16
3.1.2. HUD Actions	19
3.1.3. Research Actions	22
3.1.4. Map Actions	22
3.1.5. Difficulty Actions	22
3.2. Interacciones	22
3.3. Reglas	23

1. Ideas Iniciales

1.1. Temática

La conquista de un nuevo planeta siempre es dura. Protege tu base, de monstruos gigantes, construyendo torretas. En "The Last Shelter", la humanidad ha llegado a un nuevo planeta habitado por monstruos que no piensan compartir el territorio. Para lidiar con esta situación, y defenderte de las hordas de enemigos, construye tus torres y defiende tus asentamientos. Obtén los recursos que necesitas y defiende tu base, culmina la conquista del nuevo mundo.

El juego se desarrolla en un espacio continuo y 3D. Las posiciones de las torretas están en espacios discretos 2D, con posiciones fijas en las que se puede colocar, y el camino de las hordas enemigas también será 2D.

Hay distintas perspectivas para que el jugador perciba el mundo según la situación:

- **Vista Ortográfica (Proyección Ortogonal).** Esta vista permite al jugador construir la estrategia de defensa (posicionar las torres). El jugador puede colocar, mejorar, vender y proteger las torres defensivas durante el transcurso del juego.
- **Vista Isométrica.** Esta vista es la otra cámara principal junto con la Vista Ortográfica. Permite realizar las mismas acciones, con una perspectiva diferente.
- **Vista de Torre Defensiva.** Esta vista es sólo para ver lo que la torre defensiva ve. Sirve para observar como llegan los enemigos. La cámara se sitúa sobre las armas de la torre, de forma que cuando se mueva para apuntar, la cámara se mueve también, acompañando el movimiento.
- **Vista de Enemigos.** Esta vista es sólo para ver el terreno desde el punto de vista de los enemigos. Se utiliza para ver como avanzan los enemigos hacia las torres. La cámara se sitúa en primera persona y sigue el movimiento del enemigo seleccionado.

El jugador es algo exterior al juego, no tiene avatar pero puede crear y manipular distintas torres de varias formas para lograr el objetivo. El jugador controla las torres que defienden la base. Las torres se enfrentan a las hordas de enemigos resultando en la eliminación de los enemigos o en la destrucción de la torre. Otra interacción es el uso de habilidades especiales que afectan al mundo del juego.

Fuera de los niveles, el jugador puede conseguir mejoras gastando los puntos de experiencia. También fuera de los niveles, el jugador puede elegir el nivel que quiere jugar.

Además, el jugador, puede tomar el control de las armas de la base para utilizarlas como apoyo para las torres defensivas. Es la base la que posee las armas especiales.

1.2. Mecánicas del Juego

El juego tiene una serie de objetivos. Para completar el objetivo principal de cada nivel el jugador debe sobrevivir a un número determinado de oleadas de enemigos evitando que estos destruyan la base.

Para cumplir los objetivos secundarios el jugador debe cumplir ciertos requisitos al terminar el nivel:

- Completar el nivel sin recibir daño en la base.
- Completar el nivel sin perder más de un número de terminado de torres.
- Completar el nivel sin usar una determinada torre.
- Completar el nivel sin perder ninguna torre.
- Completar el nivel sin usar habilidades especiales.

El principal reto al que se enfrenta el jugador son los enemigos. Debe usar las torres para acabar con ellos y evitar que destruyan la base. También debe lidiar con la posible falta de recursos para construir o mejorar sus torres.

1.2.1. Acciones

Acciones del Main Mode:

- **Jugador**
 - Ataque Especial
 - Construir Torre y Torre de Energía
 - Controlar Base
 - Disparar
 - Localizar Objetivo
 - Mejorar Torre y Torre de Energía
 - Poner Cámara en Torre
 - Poner Escudo en Torre
 - Reparar Torre
 - Seleccionar Torre
 - Vender Torre
- **Enemigos**
 - Atacar Torre
 - Avanzar
 - Colisión de Enemigos con Torres
 - Colisión de Enemigos en Distintas Rutas
 - Colisión de Enemigos en una Misma Ruta
 - Orientar Enemigo
 - Poner Cámara en Enemigo

Acciones del Mapa:

- Acceder al Research Mode
- Seleccionar Nivel
- Volver al Menú Principal

Acciones del Research Mode

- Cancelar Mejora
- Confirmar Mejora
- Elegir Nivel de Mejora

Acciones de Configuración:

- Acelerar
- Ralentizar
- Reanudar
- Reiniciar
- Salir

1.2.2. Interacciones

- La derrota de las hordas de enemigos otorgan recursos.
- La aparición de un enemigo en el radio de acción de la torre la activa. Las torres tienen enemigos predilectos a los que atacar.
- Orientar Arma torre hacia el enemigo seleccionado.
- Localizar Objetivo al que disparar por parte de las torres.
- Al inicio del nivel se reciben recursos.
- Las hordas enemigas se desplazan siguiendo una trayectoria.
- Los enemigos atacan a las torres en su trayectoria hasta morir o destruirlas.
- La venta de una torre otorga recursos.
- Cambio de cámara actual a cámara en la nuke al dispararla.

1.2.3. Recursos

- **Energía:** se incrementa con torres de energía y eliminando enemigos y al comienzo de una oleada además de vender torretas, se gasta en construcción de torretas, mejoras de torretas, uso de habilidades tanto de torreta como propia. No es acumulable para otros niveles.
- **Experiencia:** al cumplir objetivos secundarios y completando niveles, se gasta en mejoras.

1.2.4. Verbos

Verbos del Jugador:

- Cambio de Cámara
- Comenzar Oleada
- Construir
- Defender
- Mejorar
- Proteger
- Reparar
- Seleccionar
- Vender

Verbos de las Torres:

- Destruir
- Disparar
- Matar
- Recolectar

Verbos de los Enemigos:

- Atacar
- Avanzar
- Destruir
- Morir

2. Entidades y Atributos

2.1. Descripción

Torre

Entidad encargada de dañar a los enemigos.

- **MAX_HEALTH {INTEGER}**
 - f(Upgrade, Research.Improvements, Type)
 - Vida Máxima de la entidad Tower.
- **Defensa {INTEGER}**
 - f(Upgrade, Research.Improvements, Type)
 - Defensa de la entidad Torre.
- **Health {0, MAX_HEALTH}**
 - f(LVL, Research.Improvements, Type)
 - Vida actual de la entidad Tower.
- **Upgrade {1, 2, 3}**
 - Actualización de la entidad Tower.
- **Type {T1, T2, T3}**
 - Tipo de la entidad torre T1 (ligera), T2 (pesada), T3 (zona).
- **Range {1, MAX_RANGE}**
 - f(LVL, Research.Improvements, Type)
 - Distancia a la cual se localiza al objetivo mediante colisión.
- **Area of Effect {INTERGER}**
 - f(Type)
 - Área de daño de los disparos.
- **Damage {1, MAX_DMG}**
 - f(LVL, Type)
 - Daño que afecta a la entidad enemigo modificando el valor health del mismo.
- **Rate of fire {T_MIN, T_MAX}**
 - f(LVL, Research.Improvements, Type)
 - Velocidad con la cual la entidad torre ejecuta la acción de atacar.
- **Appearance {VISUAL_MODEL}**
 - f(LVL, Type)
 - Diseño gráfico de la entidad Tower.

- **Sell Price {INTEGER}**
 - $f(\text{LVL}, \text{Research.Improvements}, \text{Type})$
 - Cantidad de recursos que da la venta el objeto Torre.
- **Buy price {INTEGER}**
 - $f(\text{Type})$
 - Cantidad de recursos que cuesta colocar la torre.
- **Repair price {INTEGER}**
 - $f(\text{Type})$
 - Cantidad de recursos que cuesta reparar la torre.
- **Upgrade Price {INTEGER}**
 - $f(\text{Upgrade}, \text{Type})$
 - Cantidad de recursos que cuesta el upgrade de Tower.
- **Camera {ON, OFF}**
 - Dato que Indica si la entidad Camera está activa.
- **Preferred_Target {Bicho.Type}**
 - Dato que indicara el objetivo preferente del objeto Tower.

Cámara

Entidad encargada de la vista desde distintos objetos del juego, solo uno a la vez.

- **Coordenadas {X, Y, Z}**
 - Valores que indican la posición del objeto Camera en el mundo del juego.
- **Dirección**
 - Valor que indica hacia donde observa el objeto Camera en el mundo del juego.
- **Up Vector**
 - Valor para diferenciar arriba/abajo en la cámara.

Torre Generadora

Entidad encargada de generar recursos periódicamente.

- **MAX_HEALTH {INTEGER}**
 - Vida máxima de la entidad Torre Generadora.
- **HEALTH {0, MAX_HEALTH}**
 - Vida actual de la entidad Torre Generadora.
- **Sell Price {MAX_BENEFITS}**
 - $f(\text{LVL}, \text{Research.Improvements})$
 - Cantidad de recursos que da la venta del objeto Torre Generadora.

- **Resources {INTEGER}**
 - f(Research.Improvements)
 - Valor que indica la cantidad de recursos que genera el objeto Torre Generadora.
- **Rate_of_Resources {T_MIN, T_MAX}**
 - f(LVL, Research.Improvements)
 - Velocidad con la cual la entidad Torre Generadora ejecuta la acción de dar recursos.
- **Appearance {VISUAL_MODEL}**
 - Diseño gráfico de la entidad Torre generadora.

Base

Entidad que el jugador debe proteger para completar el nivel. También puede controlarla para dañar a los enemigos.

- **MAX_HEALTH {INTEGER}**
 - Vida máxima de la entidad Base.
- **HEALTH {0, MAX_HEALTH}**
 - Vida actual de la entidad Base.
- **Tower_Play_Mode {ON, OFF}**
 - Variable que indica si está activo el modo de control de la base.
- **Range {1, MAX_RANGE}**
 - Alcance máximo de la base en modo controlado.
- **DAMAGE {1, MAX_DMG}**
 - Daño de la base en modo controlado.
- **Rate_of_fire {T_MIN, T_MAX}**
 - Cadencia de fuego de la base en modo controlado.
- **Camera {ON, OFF}**
 - f(Play_Mode).
 - Variable que indica si está activada la cámara de la base.
- **Appearance {VISUAL_MODEL}**
 - Diseño gráfico de la entidad Torre generadora.

Bicho

Entidad de la que el jugador debe defender la base.

- **MAX_HEALTH {INTEGER}**
 - Vida máxima de la entidad Bicho.

- **HEALTH {0, MAX_HEALTH}**
 - Vida actual de la entidad Bicho.
- **Resources {INTEGER}**
 - f(TYPE)
 - Cantidad de recursos que aporta el bicho al ser eliminado.
- **Damage {MIN_DMG, MAX_DMG}**
 - f(TYPE)
 - Daño que inflige el bicho a las unidades aliadas.
- **Speed {MIN_SPEED, MAX_SPEED}**
 - f(TYPE)
 - Velocidad de desplazamiento del bicho.
- **TYPE {T1, T2, T3, SB}**
 - Variable que indica el tipo de bicho.
- **Appearance {VISUAL_MODEL}**
 - Diseño gráfico de la entidad Bicho.

Mejoras

Mejoras disponibles para las torres.

- **Effect {EFFECT}**
 - Efecto que tiene la mejora sobre la torre.
- **Cost {1, MAX_COST}**
 - Cantidad de recursos que cuesta la mejora.
- **Appearance {VISUAL_MODEL}**
 - Diseño gráfico de la mejora.

LVL

- **Init_resources {1, MAX_RESOURCES}**
 - Cantidad de recursos con la que empieza el juego.
- **Pause {TRUE, FALSE}**
 - Atributo que indica si el tiempo del juego está detenido.
- **Accelerate {TRUE, FALSE}**
 - Atributo que indica si el tiempo del juego está acelerado.
- **Resources {0, MAX_RESOURCES}**
 - Cantidad de recursos que tiene el jugador.

- **Enemy_spawn_point {ListX, Y}**
 - Lista de puntos de spawn posibles para los enemigos.
- **Tower_spawn_point {ListX, Y}**
 - Lista de puntos de spawn posibles para las torres.
- **Collector_spawn_point {ListX, Y}**
 - Lista de puntos de spawn posibles para las torres recolectoras.
- **Wave_number {1, MAX_WAVES}**
 - Número de oleadas necesarias para completar el nivel.
- **Wave_started {TRUE, FALSE}**
 - Indica si hay una oleada activa.

Wave

Entidad que contiene y gestiona a una oleada particular de bichos.

- **Enemigos {List(Enemigos, Spawn Enemigo, Tiempo)}**
 - f(Level.Wave_Number, Level.Enemy_Spawn_Point)
 - Una lista que contiene información sobre cada enemigo que aparecerá

Research

Entidad que contiene las mejoras del jugador.

- **EXP {INTEGER}**
 - f(NIVEL)
 - La experiencia actual que tiene el jugador
- **Improvements {List(Tipo, Nivel, EXP)}**
 - Una lista con las mejoras que podría comprar el jugador.

Partida

Entidad que contiene el progreso que ha hecho el jugador en el juego.

- **Nivel desbloqueado {1,2,3}**
 - El nivel hasta el que se ha llegado.
- **Logros {List(Nivel, Logro)}**
 - Una lista que contiene información sobre los logros que se han obtenido en cada nivel.
- **Nivel superado {List(dificultad)}**
 - Una lista que contiene información sobre la dificultad en la que se ha superado cada nivel.

HUD

Todos los atributos de la IU.

■ Main Mode

- El HUD durante juego normal.
- **Resources**
 - f(Level.resources)
- **EXP**
 - f(Research.EXP)

■ Pop-Up Menu Mode

- Varios menus o iconos que aparecen superimpuestos sobre el juego.
- **Tipo {Torre, Nivel, Barra de vida}**
- (tipo == Torre)
 - **Tower_Menu**
 - ◊ Un menu con todas las acciones relativas a una torre disponibles.
- (tipo == Barra de vida)
 - **HP**
 - ◊ Una barra que muestra la vida actual respecto a la máxima de una torre/enemigo/base.
 - ◊ f(Torre.HP, Torre.MAX_HEALTH) || f(Base.HP, Base.MAX_HEALTH) || f(Enemigo.HP, Enemigo.MAX_HEALTH)
- (tipo == Nivel)
 - **Logros**
 - ◊ Los logros que se ha conseguido en cierto nivel.
 - ◊ f(Partida.Logros)
 - **Menu_dificultad**
 - ◊ Permite seleccionar los distintos niveles de dificultad, además de mostrar cuales se han superado.
 - ◊ f(Partida.Nivel_superado)

■ Research Mode

- El HUD durante la compra de mejoras entre niveles.
- **Mejoras_menu**
 - Las mejoras compradas y disponibles para comprar.
 - f(Research.Improvements)

■ Map Mode

- El HUD durante la selección de niveles.
- **Niveles**
 - Los niveles seleccionables.
 - f(Partida.Nivel_superado)

Ataque Especial

- **TYPE {Bomb, Nuke}**
 - El tipo de ataque especial.
- **Damage {(0, MAX_LEVEL)}**
 - $f(\text{Research.Improvements}, \text{TYPE})$
 - El daño que hará el ataque especial por defecto.
- **Appearance {VISUAL_MODEL}**
 - $f(\text{TYPE})$
 - La apariencia visual de la explosión.
- **Area Of Effect {INTEGER}**
 - $f(\text{TYPE})$
 - Área en el que hace daño el ataque.
- **Cooldown {INTEGER}**
 - $f(\text{Research.Improvements}, \text{TYPE})$
 - Cuánto tiempo ha de transcurrir antes de poder usar el ataque especial de nuevo.

2.2. Estados S-T

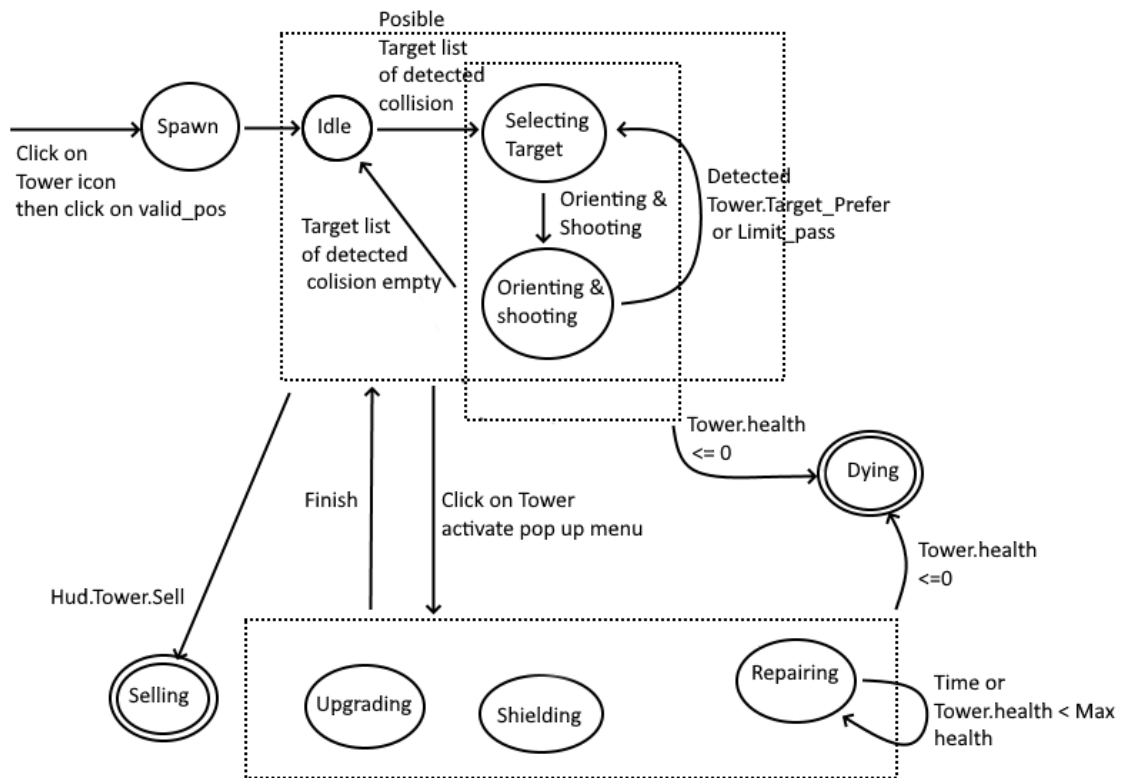
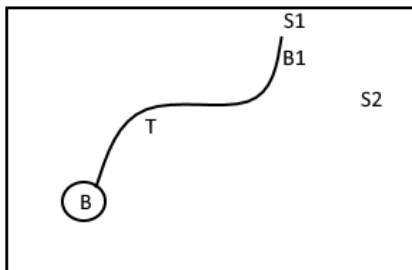


Figura 1: Diagrama Entidad-Relacion Torre



Entidad Bicho:
Atributo a modelar: Movimiento

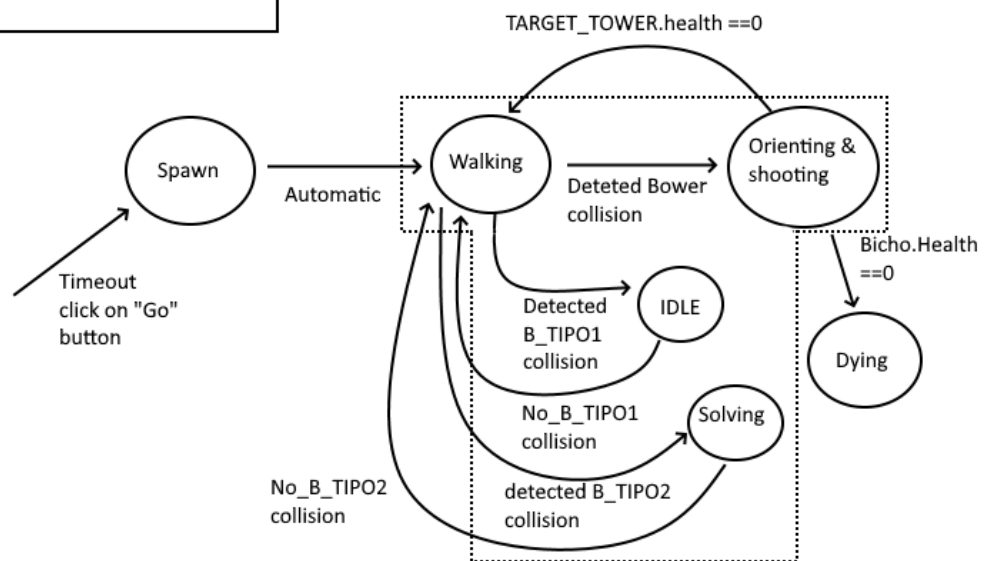


Figura 2: Diagrama Entidad-Relacion Enemigos

3. Acciones, Interacciones y Reglas

3.1. Acciones

3.1.1. Pop-Up Actions

■ Vender

- Limitaciones: vida torre mayor que 0 (no destruida)
- Consecuencias: espacio libre, cambio visual (torre desaparece), aumentan recursos

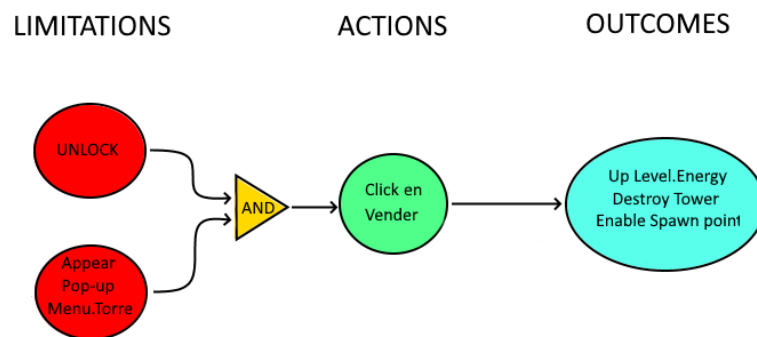


Figura 3: Diagrama de la Acción Vender

■ Reparar

- Limitaciones: vida torre mayor que 0 y menor que vida maxima, recursos suficientes
- Consecuencias: aumento vida torre, disminución de recursos

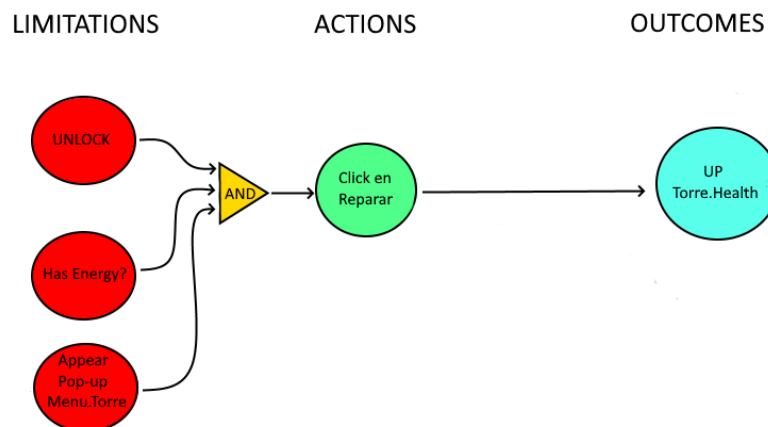


Figura 4: Diagrama de la Acción Reparar

■ Mejorar

- Limitaciones: tener Torre.Nivel menor que Torre.MAX_LEVEL, recursos suficientes, mejora desbloqueada
- Consecuencias: aumento de Torre.Health, disminución de recursos

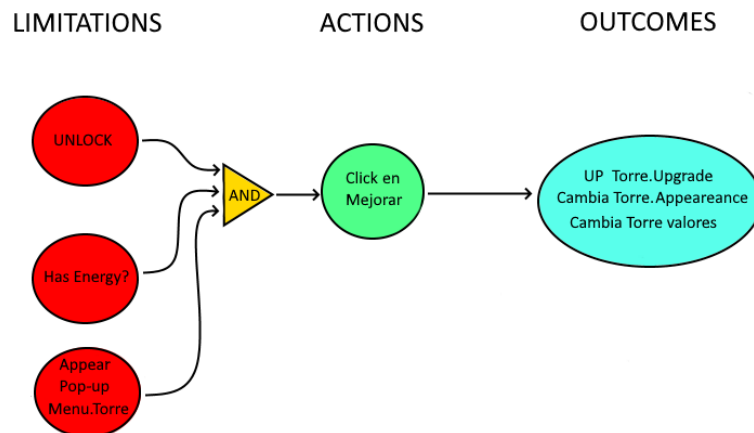


Figura 5: Diagrama de la Acción Mejorar

■ Escudo

- Limitaciones: recursos suficientes, habilidad desbloqueada
- Consecuencias: reducción de daño sobre torre, disminución de recursos

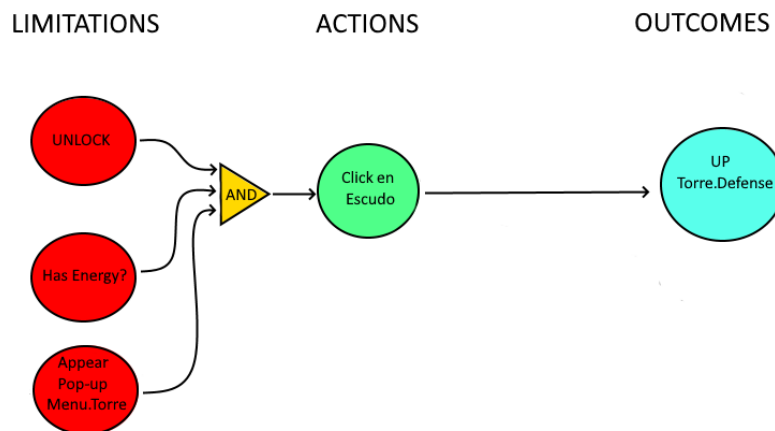


Figura 6: Diagrama de la Acción Escudo

- **Controlar Base**

- Limitaciones: sin limitaciones
- Consecuencias: activa el modo de disparo de la base o Tower_Play_Mode

ACTIONS

OUTCOMES

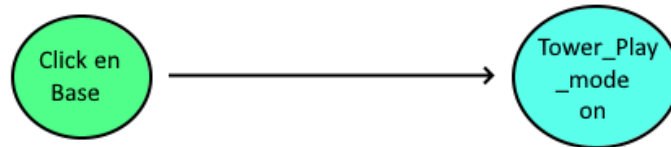


Figura 7: Diagrama de la Acción Control Base

LIMITATIONS

ACTIONS

OUTCOMES

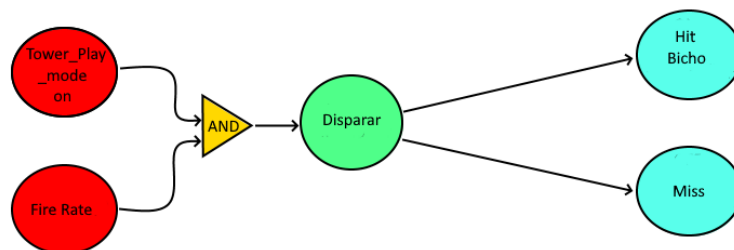


Figura 8: Diagrama de la Acción Disparar

3.1.2. HUD Actions

■ Construir Torre (0 Recolector)

- Limitaciones: recursos suficientes, espacio libre, torre desbloqueada
- Consecuencias: cambio visual (nueva torre), espacio ocupado, reducción de recursos

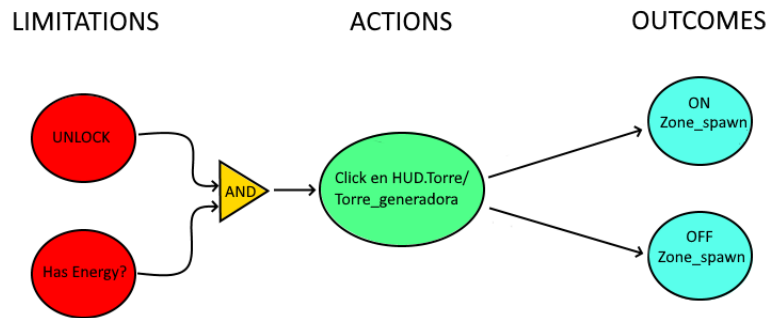


Figura 9: Diagrama de la Acción Click en Icono de Torre

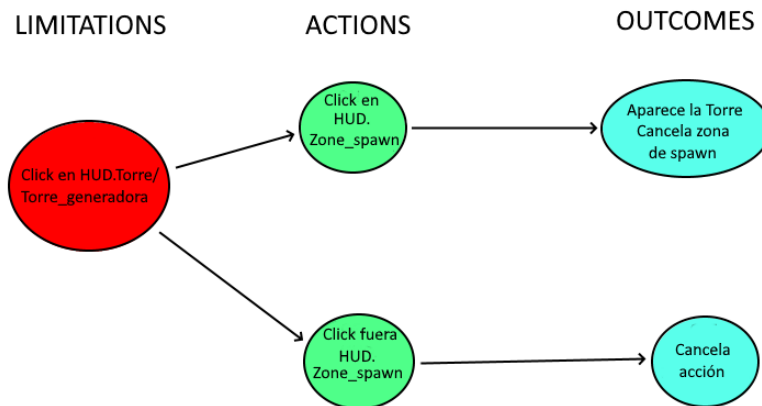


Figura 10: Diagrama de la Acción Soltar torre

■ Seleccionar Torre

- Limitaciones: torre seleccionable
- Consecuencias: activa menú pop-up

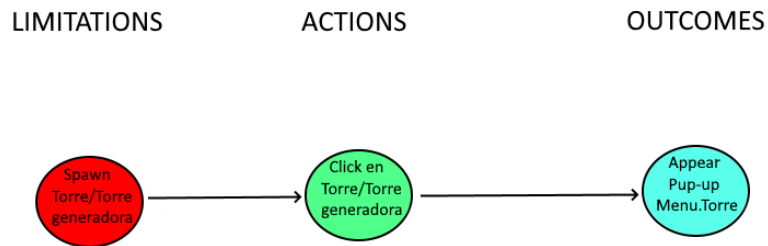


Figura 11: Diagrama de la Acción Click en Torre

■ Cámara en Torre o Enemigo

- Limitaciones: torre o enemigo seleccionable
- Consecuencias: cambia la posición de la cámara

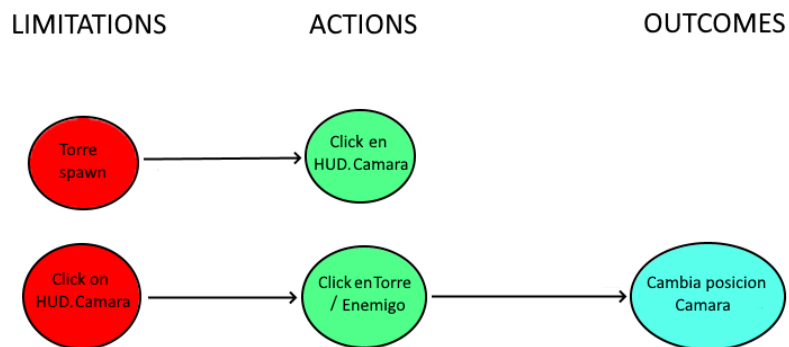


Figura 12: Diagrama de la Acción Camara

■ Ataque Especial

- Limitaciones: ataque activo (no en tiempo de recarga), ataque desbloqueado
- Consecuencias: disminución de la vida de los enemigos, activar tiempo de recarga

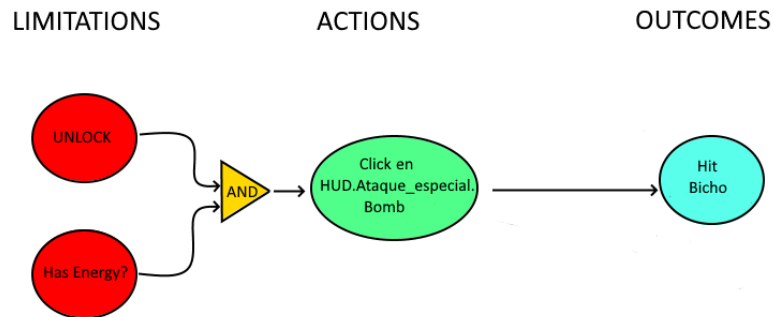


Figura 13: Diagrama de la Acción Bomba

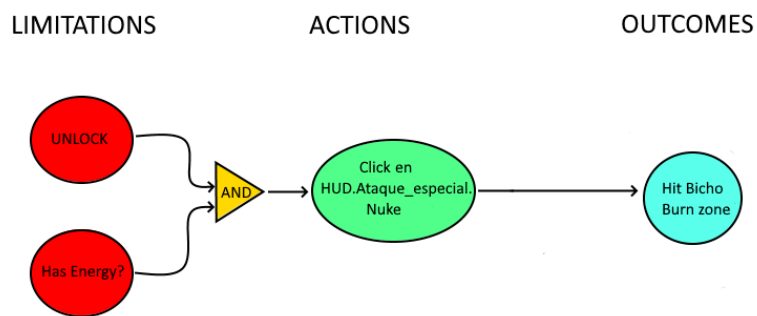


Figura 14: Diagrama de la Acción Nuke

■ Pausar el juego

- Limitaciones: juego activo
- Consecuencias: juego pausado

■ Reanudar Juego

- Limitaciones: juego pausado
- Consecuencias: juego reanudado

■ Acelerar Juego

- Limitaciones: juego en velocidad inferior a la que se active
- Consecuencias: juego acelerado

■ Ralentizar Juego

- Limitaciones: juego en velocidad superior a la que se active
- Consecuencias: se ralentiza el juego

3.1.3. Research Actions

- **Elegir Mejora**
 - Limitaciones: tener XP, tener el nivel previo de la mejora si lo hay
 - Consecuencias: se selecciona la mejora, sin hacerse efectiva
- **Confirmar Mejora**
 - Limitaciones: haber seleccionado la mejora, pulsar el botón de aceptación
 - Consecuencias: se agrega la mejora, se vuelve a la pantalla de mapa
- **Cancelar Mejora**
 - Limitaciones: pulsar el botón de cancelación
 - Consecuencias: se vuelve a la pantalla de mapa, no se confirman las mejoras elegidas

3.1.4. Map Actions

- **Selección de Nivel**
 - Limitaciones: nivel desbloqueado
 - Consecuencias: menú de selección de dificultad
- **Volver al Menú**
 - Limitaciones: no tiene
 - Consecuencias: vuelve al menú principal
- **Acceder al Research**
 - Limitaciones: no tiene
 - Consecuencias: abre el menú research

3.1.5. Difficulty Actions

- **Seleccionar Dificultad**
 - Limitaciones: haber seleccionado un nivel
 - Consecuencias: accede a la partida en esa dificultad

3.2. Interacciones

- **Colisión enemigo-torre**
 - Consecuencias: El enemigo se orienta hacia la torre y comienza a atacar. Mientras ataca, la torre pierde health. Si la torre se queda sin health, muere.
- **Colisión enemigo-base**
 - Consecuencias: El enemigo se orienta hacia la base y comienza a atacar. Mientras ataca, la base pierde health. Si la base se queda sin health, el jugador pierde.

- **Colisión enemigo-enemigo**

- Consecuencias: El enemigo de ID menor tiene prioridad, tanto en la misma vía como en vías distintas.

- **Entrada de un enemigo en la LOS de una torre**

- Consecuencias: La torre, si está disparando ya, evalúa de nuevo a qué enemigo debería estar disparando. Si no, simplemente se pone a disparar al enemigo.

- **Colisión bala/misil/explosión-enemigo**

- Consecuencias: El enemigo pierde health. Si el enemigo se queda sin health, muere y da una cantidad de recursos al jugador.

3.3. Reglas

- **Colocar torre**

- Si hay suficientes recursos y no hay ninguna otra torre en el punto seleccionado, coloca una torre.
- IF tower.buy_price < level.resources AND tower_spawn_point.is_empty() AND tower_spawn_point.isSelected() THEN tower.spawn(tower_spawn_point)

- **Reparar las torres**

- Si una torre está dañada y el jugador tiene la suficiente energía, se usa energía para restaurar la salud de la torre.
- IF(tower.HEALTH < tower.MAX_HEALTH) AND (level.resources >= tower.repair_price) THEN (level.resources -= tower.repair_price) AND (tower.HEALTH = tower.MAX_HEALTH)

- **Mejorar las torres**

- Si la experiencia es suficiente, se puede aumentar el nivel de la torre.
- IF (research.XP >= tower.UPGRADE_COST) THEN tower.UPGRADE += 1

- **Uso de habilidades**

- Si la habilidad está disponible, el jugador puede utilizar la habilidad.
- IF special_attack.isSelected() AND special_attack.cooldown == 0 AND level.pointClicked(point) THEN special_attack.launch()

- **Eliminar las torres para recuperar recursos**

- Si la vida de la torre no es 0 y la torre no se está reparando, el jugador podrá eliminar la torre y recuperar recursos.
- IF(tower.HEALTH > 0 AND NOT tower.repairing()) THEN (tower.HEALTH = 0) AND (level.resources += tower.COST) AND (tower.destroy())

- **Saquear enemigo**

- Si un bicho muere y la energía no están al máximo, entonces el jugador gana energía según el tipo de bicho.
- IF (bicho.HEALTH == 0) AND (level.resources < MAX_RESOURCES) THEN
level.resources += bicho.resources

- **Activar torre**

- Si un bicho entra en el radio de acción de una torre y esta no está atacando o el objetivo es prioritario y el objetivo actual no lo es, entonces esta se activa y ataca al bicho.
- IF collision(tower, bicho) THEN IF ((bicho.tipo == tower.preferred_target AND tower.target != tower.preferred_target) OR tower.target == NULL) THEN
tower.attack(bicho)

- **Iniciar recursos**

- La partida comienza con una determinada cantidad de energía.
- level.resources = level.init_resources

- **Bicho ataca**

- Si una torre está en el rango de acción de un bicho, este la ataca hasta que la torre es destruida o el bicho muere.
- IF collision(bicho, tower) THEN WHILE tower.HEALTH > 0 AND bicho.HEALTH > 0
DO attack(tower)