

# Desarrollo de un Bot para Dota 2

Josue Rolando Naranjo Sieiro

Noel Pérez Calvo

July 18, 2024

## Contents

### 1 Introducción

En este informe se presenta el desarrollo de un bot para el videojuego Dota 2. El objetivo es implementar un bot que se mueva al centro del mapa y gire en círculos. A continuación, se describen los pasos seguidos y el código utilizado para lograr esta funcionalidad.

### 2 Arquitectura del Bot

El bot se compone de varios scripts en Lua, cada uno con una función específica:

- **bot\_generic.lua:** Este script es el punto de entrada principal que llama a la función de pensamiento del bot personalizado.
- **bot\_think.lua:** Contiene la lógica básica de movimiento y ataque del bot.
- **botinit.lua:** Inicializa el bot genérico.
- **custom\_bot.lua:** Define el comportamiento personalizado del bot, en este caso, moverse en círculos alrededor del centro del mapa.

## 3 Implementación

### 3.1 bot\_generic.lua

Este archivo incluye el script personalizado del bot y llama a su función de pensamiento en un bucle continuo.

```
1 -- Incluir el script personalizado del bot
2 require("bots/custom_bot")
3
4 -- Funci n principal que se ejecuta repetidamente
5 function Think()
6     -- Llama a la funci n Think del script personalizado
7     customBotThink()
8 end
```

Listing 1: bot\_generic.lua

### 3.2 bot\_think.lua

Aquí se define la lógica básica del bot, como moverse al carril medio y atacar creeps enemigos si están cerca.

```
1 function BotThink()
2     local npcBot = GetBot()
3
4     -- Si el bot est  muerto, no hacer nada
5     if npcBot:IsAlive() == false then
6         return
7     end
8
9     -- Si el bot est  en la fuente, moverlo al carril medio
10    if npcBot:GetLocation() == npcBot:GetTeamFountain() then
11        npcBot:Action_MoveToLocation(Vector(0, 0, 0))
12        return
13    end
14
15    -- Si hay creeps enemigos cerca, atacarlos
16    local enemyCreeps = npcBot:GetNearbyCreeps(800, true)
17    if #enemyCreeps > 0 then
18        npcBot:Action_AttackUnit(enemyCreeps[1], true)
19        return
20    end
21
22    -- Si no hay creeps enemigos, mover al bot hacia adelante
23    en el carril
    npcBot:Action_MoveToLocation(Vector(0, 0, 0))
```

```
24 end
```

Listing 2: bot\_think.lua

### 3.3 botinit.lua

Este script inicializa el bot genérico y llama a su función de pensamiento repetidamente.

```
1 -- Incluir el script genérico del bot
2 require("bots/bot_generic")
3
4 -- Función que se llama repetidamente para pensar
5 function BotThink()
6     Think() -- Llama a la función Think de bot_generic.lua
7 end
```

Listing 3: botinit.lua

### 3.4 custom\_bot.lua

Este archivo contiene la lógica personalizada del bot, en este caso, moverse en círculos alrededor del centro del mapa.

```
1     -- Variables para el movimiento circular
2     local centerX = 0
3     local centerY = 0
4     local radius = 1000
5     local angle = 0
6     local angleStep = 0.05
7
8     -- Función Think del bot personalizado
9     function customBotThink()
10         -- Calcula la nueva posición
11         local posX = centerX + radius * math.cos(angle)
12         local posY = centerY + radius * math.sin(angle)
13
14         -- Mueve al bot a la nueva posición
15         MoveToLocation(posX, posY)
16
17         -- Incrementa el ángulo
18         angle = angle + angleStep
19     end
20
21     -- Función para mover al bot a una ubicación
22     -- específica
23     function MoveToLocation(x, y)
24         -- Obtiene el bot actual
```

```

24         local bot = GetBot()
25         local botLoc = bot:GetLocation()
26
27         -- Mueve el bot a la nueva ubicaci n
28         bot:Action_MoveToLocation(Vector(x, y, botLoc.z))
29     end
30

```

Listing 4: custom\_bot.lua

## 4 Recursos adicionales

### 4.1 Documentación de Dota Bot Scripting

Para obtener más información sobre cómo crear bots en Dota 2, puedes visitar la página oficial de [Dota Bot Scripting](#). Esta página contiene todas las funciones implementadas por Valve que se pueden utilizar para el desarrollo de bots.

### 4.2 Bot de ejemplo: Little Rockie Bot

El bot [Little Rockie Bot](#) contiene bots ya hechos con los que se puede probar e ir viendo el funcionamiento. Este bot facilita la comprensión de cómo se pueden probar los bots dentro del juego.

### 4.3 Comandos para probar bots

Para probar los bots creados, puedes usar los siguientes comandos en la consola de Dota 2:

- **map dota gamemode 1:** Este comando crea una partida de Dota en el modo All Pick, donde se pueden seleccionar todos los héroes.
- **dota bot populate:** Llena ambos equipos con los bots creados.

## 5 Conclusión

Este informe describe la implementación de un bot de Dota 2 que se mueve al centro del mapa y gira en círculos. La estructura modular del código permite añadir fácilmente nuevas funcionalidades y comportamientos al bot en el futuro. Esta implementación básica puede servir como punto de partida para desarrollar bots más complejos y sofisticados en el entorno de Dota 2.