

Inteligencia Artificial Proyecto 2

Camilo Provoste
cprovoste22@alumnos.ugal.cl

Rodrigo Soto
rsoto22@alumnos.ugal.cl

Ian Pereira
ipereira22@alumnos.ugal.cl

Diego Morales
diemorales22@alumnos.ugal.cl

EXPLICACIÓN DE LAS IMPLEMENTACIONES

I. BLINKY

Para la implementación de blinky se usó un Finite State Machine base, no se agregó otro estado para evitar la duplicación de estados y dependencia entre transiciones.

Chase State: El chase se modificó para que en vez de solo perseguir a Pac-Man, se comporte como un acechador que espera la oportunidad para acorralarlo, manteniendo una distancia a este, considerando la posición de otros fantasmas para intentar hacer una “encerrona” y si puede, emboscar al Pac-Man basándose en la dirección predicha de este, por último acercándose cuando Pac-Man este cerca de una Power Pill, para interceptarlo.

ScatterState y FrighteredState: se mantuvieron igual, el Scatter hace que blinky vaya a su esquina y el Frightered se activa cuando pacman come una Power Pill.

Transiciones: entre Chase y Scatter cambian de estado constantemente por un periodo de tiempo, se puede cambiar a Frightered desde cualquiera de los dos estados, siempre y cuando Blinky este Azul.

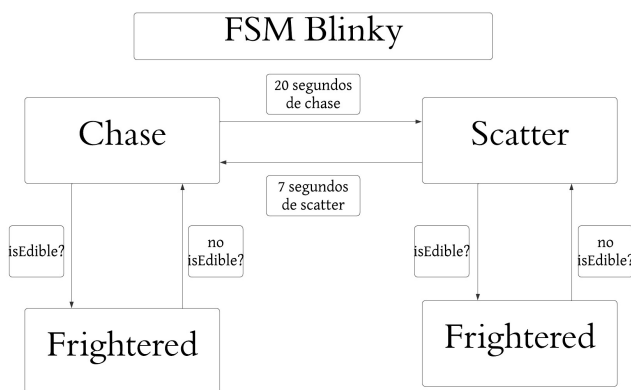


Fig. 1. Diagrama de Blinky

II. PINKY

La implementación que se agregó a PinkyController fue un FSM con los siguientes estados:

Chase: persigue al Pacman con la condición de que vaya 5 bloques de distancia a la dirección por donde el este mirando, pero el empezara a seguir sin la condición si es que Pinky este muy cerca de el.

Fright: escapa del Pacman tomando el camino mas lejano de el, pero si es que esta muy lejos de Pacman se acercara un poco hacia el.

InitialPoint: va hacia una posición del tablero en un inicio para distanciarse de sus compañeros.

Scatter: se resguarda hacia la esquina superior izquierda.

Transiciones:

Chase: Mientras que el fantasma no este en **isEdible**, haya llegado al punto establecido y que aun no hayan pasado los veinte segundos, estará en el estado Chase.

InitialPoint: Mientras aun no haya llegado al punto establecido, estará en estado InitialPoint.

Fright: Mientras que el fantasma este en **isEdible**, estará en estado Fright.

Scatter: Mientras que el fantasma no este en **isEdible**, y que este entre los segundos veinte y veintisiete, estará en el estado Chase.

PINKY FSM

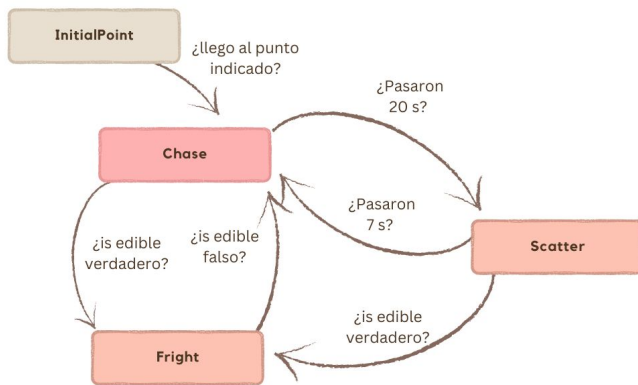


Fig. 2. Diagrama de Pinky

III. INKY

Los cambios sobre InkyController fue un Behavior Tree con los siguientes estados:

Home: mientras el puntaje de Pacman sea menor a 300 el fantasma se dirige a la esquina inferior derecha.

Fright: se escapa de Pacman por el camino mas largo, pero solo considera 2 movimientos validos y no todos.

ForceMovement: revisa una lista de posibles movimientos e intenta acorralar a Pacman por la derecha o abajo.

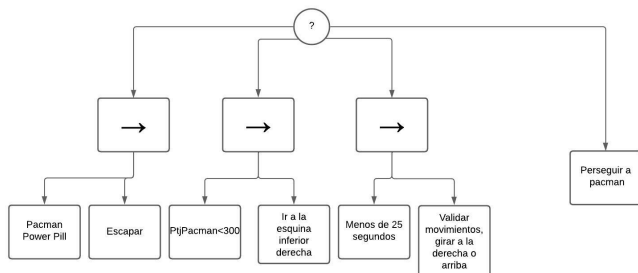


Fig. 3. Diagrama de Inky

IV. SUE

Para SueController se hizo con un Behavior Tree con las siguientes implementaciones:

SueIfPill: Comprueba que queden PowerPills en el mapa, si es así, va a SueGoToPill.

SueGoToPill: Busca la PowerPill más cercana a Pacman y se dirige hacia ella.

El comportamiento de Frightened fue separado en las secuencias frightenedFarther y frightenedCloser, con el objetivo de no estar muy lejos de Pacman una vez sea vulnerable nuevamente.

FrightenedFarther: Comprueba si Sue es comible, y luego si está a menos de 500 de distancia, si se cumplen ambas, se aleja de Pacman.

FrightenedClose: Comprueba si Sue es comible, y luego si está más de 500 de distancia, si se cumplen ambas se acerca a Pacman.

SueChase funciona como un chase normal, persiguiendo a Pacman.

Sue Behavior Tree

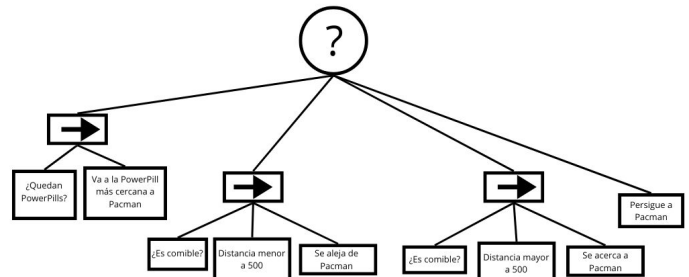


Fig. 4. Diagrama de Sue