

“Hoe een haas een koe vangt”

Week 2

Opdracht2: Finite State Machine (FSM)

1. Maak een standalone (.exe) applicatie, die op een windowsmachine probleemloos start.	Een .exe applicatie wordt gemaakt (in SDLFramework\Debug) wanneer de Visual Studio applicatie gedebugd wordt.
2. Implementeer een FSM volgens H2. Zorg er daarbij voor dat:...	
3. ... Je (een) toestandsdiagram (men) maakt die overeen komt met jouw applicatie.	Een photo van de toestandsdiagram is te vinden in de ZIP bestand.
4. ... Jouw toestandsdiagram minstens van dezelfde moeilijkheidsgraad is als die voor de <i>Voorbeeldcase</i> (zie onder de opdracht).	CowChaseRabbitState.cpp, CowSearchForPillState.cpp, CowSearchForWeaponState.cpp, CowWanderingState.cpp, RabbitFleeingState.cpp, RabbitWanderingState.cpp, StateMachine.h
5. ... We ingame@ altijd grafisch kunnen zien in welke toestand een Entity zich bevindt (Zie T&T).	Dashboard.cpp
6. ... De start state van de Koe “Wandering” is en die van de Haas “Wandering”.	Cow.cpp, Rabbit.cpp
7. ... In elk geval de Koe van toestand kan veranderen.	Cow.cpp, CowChaseRabbitState.cpp, CowSearchForPillState.cpp, CowSearchForWeaponState.cpp, CowWanderingState.cpp,
8. ... Als de koe achter de haas aan gaat, zij dit doet dmv A* (zoals bij opdracht 1).	CowChaseRabbitState.cpp, AStar.cpp
9. Zorg dat de volgende Gameltems verspreid worden over de waypoints:	
- Pill	Pill.cpp,
- Weapon	Weapon.cpp
10. Zorg dat zowel de “Pill” als de “Weapon” upgrade de toestand van (één van) de dieren aanpast. Het is dus bijvoorbeeld prima als de Pill alleen effect heeft op de Koe en de Weapon alleen op de Haas, dat is aan jou. Als ze maar allebei in de applicatie zitten.	Pill.cpp, Weapon.cpp
11. Zorg dat ook de Haas <i>zinnig</i> van toestand kan veranderen.	Rabbit.cpp, RabbitFleeingState.cpp RabbitWanderingState.cpp