EXAMEN PARCIAL BLOQUE 2 PLANSYS2

PREGUNTA 1

¿Qué es un sistema de planificación?

a. Es uno o varios algoritmos de planificación que resuelven planes especificados en PDDL.

b. Es una aplicación que permite resolver misiones en robot en un escenario complejo específico usando planificación.

c. Es un framework para la creación y ejecución de planes en robótica.

d. Es un estándar que define cómo especificar problemas y dominios PDDL.

PREGUNTA 2

¿En qué parte de PlanSys2 se cargan los plugins de los planificadores (POPF, Optics, …) para resolver planes?

a. Planner Expert

b. Problem Expert

c. Domain Expert

d. Executor

e. En ningún sitio de PlanSys2. No existen plugins de planificadores.

PREGUNTA 3

Una aplicación que usa PlanSys2 está compuesta por …

a. Un dominio PDDL y la implementación de sus acciones

b. Un controller (obligatorio) y la implementación de las acciones

c. Un controller, la terminal de PlanSys2, y un dominio PDDL y la implementación de sus acciones

d. Un controller (opcional), un dominio PDDL y la implementación de sus acciones

PREGUNTA 4

La terminal de PlanSys2 …

a. Es desde donde despegan la implementación de las acciones

b. Es una aplicación que te permite interactuar con PlanSys2 para obtener información y hacer peticiones a cada uno de sus componentes, usando comandos de texto

c. Es el API de cada uno de los módulos de PlanSys2, codificados en el interfaz del que heredan cada uno de sus clientes.

d. Es el estado final en el que queda PlanSys2 tras resolver y ejecutar un plan

PREGUNTA 5

get problem predicates

a. No me dice nada

b. Permite ver los predicados que hay definidos en el Domain Expert

c. Permite ver los argumentos (nombre y tipo) de un predicado

d. Permite ver los predicados que hay definidos en el Problem Expert

PREGUNTA 6

¿Dónde se usan Behavior Trees en PlanSys2?

a. El Executor codifica los planes como Behavior Trees para ejecutarlos. Las acciones también pueden ser implementadas como Behavior Trees.

b. Las acciones también pueden ser implementadas como Behavior Trees.

c. El Executor codifica los planes como Behavior Trees para ejecutarlos. Las acciones también pueden ser implementadas como Behavior Trees. Nav2 usa internamente Behavior Trees para definir cómo navegar.

d. El Executor codifica los planes como Behavior Trees para ejecutarlos.

e. Realmente no se usan.

PREGUNTA 7

¿Cómo se implementan las acciones que contiene un plan, conforme a un dominio PDDL?

a. Los Action performers son LifeCycle Nodes que se activan cuando le toca ejecutar la acción que implementan, recibiendo los argumentos que figuran en el plan.

b. Ninguna es correcta

c. Un Controller contiene la implementación de las acciones de una aplicación

d. Los Action performers son programas que se ejecutan cuando le toca ejecutar la acción que implementan.

PREGUNTA 8

¿Qué hace un Controller?

a. Inicializa el conocimiento, implementa las acciones del dominio, establece los goals, pide planes y controla la ejecución de estos planes.

b. Inicializa el conocimiento, establece los goals, pide planes y controla la ejecución de estos planes, replanificando si es necesario.

c. Inicializa el conocimiento, establece los goals, pide planes y controla la ejecución de estos planes.

d. Implementa las skills de actuación que usan las acciones.

PREGUNTA 9

¿Quién ejecuta una acción?

a. El primer Action Performer que responde al Executor cuando solicita la ejecución de una acción en el /actions_hub

b. Ninguna es correcta

c. El Action Performer que especifica el Executor el /actions_hub

d. El mejor Action Performer que responde al Executor cuando solicita la ejecución de una acción en el /actions_hub

PREGUNTA 10

¿Qué es esto?

```
class MoveAction : public plansys2::ActionExecutorClient
{
public:
  Qodo Gen: Options | Test this method
  MoveAction()
  : plansys2::ActionExecutorClient("move", 250ms)
  {
    progress_ = 0.0;
  }

private:
  Qodo Gen: Options | Test this method
  void do_work()
  {
    if (progress_ < 1.0) {
      progress_ += 0.02;
      send_feedback(progress_, "Move running");
    } else {
      finish(true, 1.0, "Move completed");

      progress_ = 0.0;
      std::cout << std::endl;
    }

    std::cout << "\r\e[K" << std::flush;
    std::cout << "Moving ... [" << std::min(100.0, progress_ * 100.0) << "%]  " <<
      std::flush;
  }

  float progress_;
};
```

a. Parte de un Controller

b. La implementación de un Action Performer

c. Esto no tiene relación con PlanSys2

d. Parte de un plugin de un planificador

PREGUNTA 11

¿Qué es “arg3” en esto?

```
<root BTCPP_format="4" main_tree_to_execute = "MainTree" >
    <BehaviorTree ID="MainTree">
        <Sequence name="root_sequence">
            <OpenGripper     name="open_gripper"/>
            <ApproachObject name="approach_object"/>
            <CloseGripper   name="close_gripper"/>
            <Move     name="move" goal="{arg3}"/>
            <OpenGripper     name="open_gripper"/>
        </Sequence>
    </BehaviorTree>
</root>
```

a. El tercer argumento del nodo Move

b. No significa nada. Es incorrecto

c. El tercer argumento de la acción que se implementa con este Behavior Tree

d. El tercer argumento con el que se llama a este controlador

SOLUCIONES

1. c
2. a
3. d
4. b
5. d
6. a
7. a
8. b
9. a
10. b
11. c