Introduction to Artificial Intelligence in Games

Lecture 1

Assessment

Total: 100 marks

- Final Exam: 60 marks
- Mid-term: 20 marks
- Project+Oral : 20 marks

Motivations for AI in Games

- Many applications for AI
  - Computer vision, natural language processing, speech recognition, …
  - But games are some of the more interesting.
  - Opponents that are challenging, or allies that are helpful.
- No need for other players:
  - If there are no other players to play with, the game can add AIs to play as the other players.
  - Also, in team-based games, teams can be filled up with AIs for example to balance the number of players.
- PVE (Player vs. Environment):
  - In some multiplayer games, the players are not competed against each other but against the game itself or the “environment”.
  - In this case, the AI can take control of the enemies in the environment.

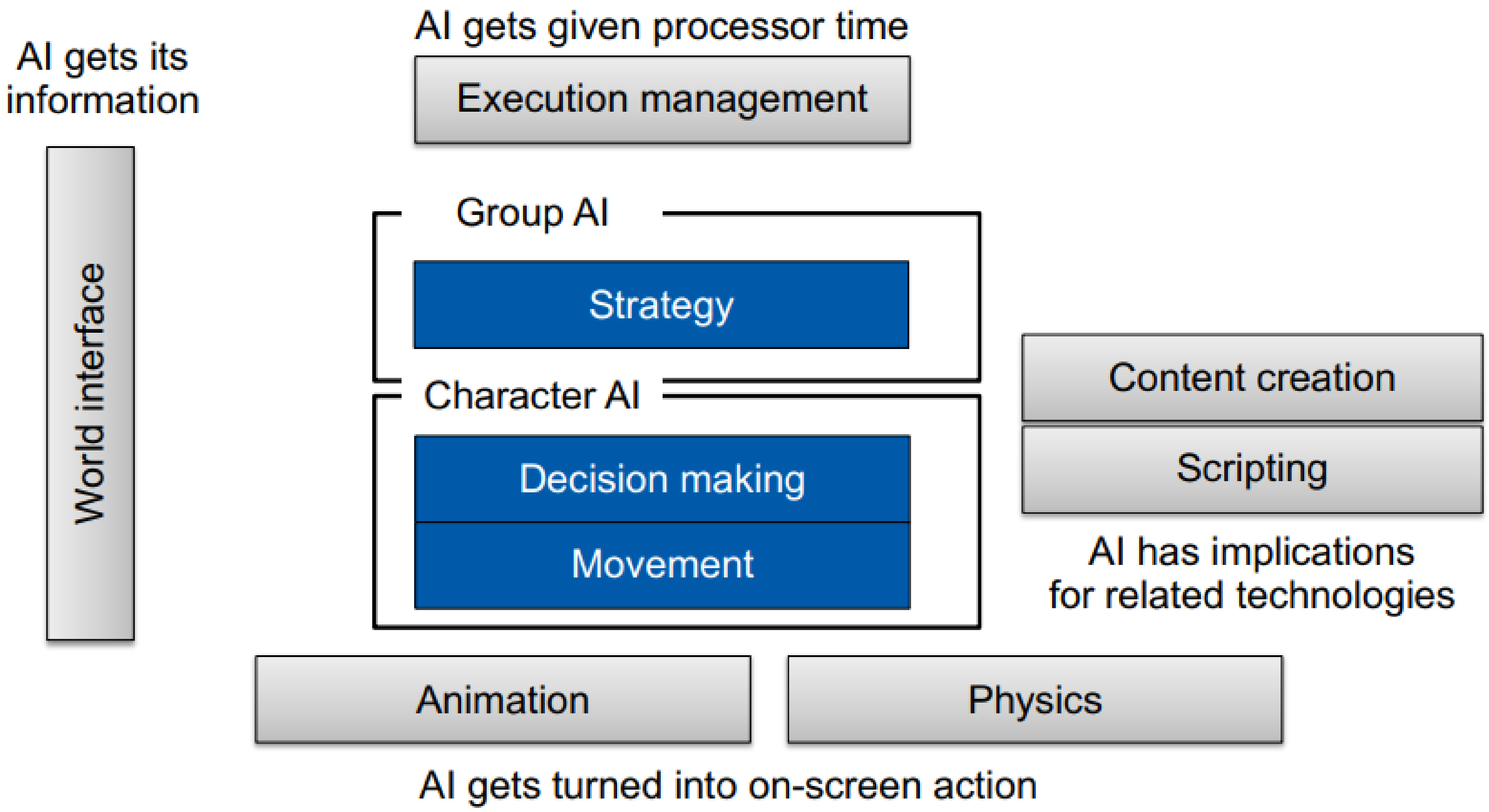
Motivations for AI in Games (cont.)

- Realistic world:
  - AIs are not only employed for controlling humanoid characters, but can also control other entities in the world.
  - Examples are animals or flocks of birds that populate the world and make it more believable.
- Challenge:
  - AI can also provide a challenge to players that a human might not be able to provide.
- Testing:
  - an AI can allow a single developer to test a game without getting other developers to play with him.

AI for Games different than AI for CS

- Must be smart,
  - Play in a fun and challenging way
- No unintended weaknesses
  - No “golden path” to defeat
  - Must not look dumb
- Must perform in real time (CPU)
- “Amount” and type of AI for game can vary

Model of Game AI



- It splits the AI task into three sections: movement, decision making, and strategy.
- The first two sections contain algorithms that work on a character-by-character basis.
- the last section operates on a whole team or side.
- Around these three AI elements is a whole set of additional infrastructure.

Model of Game AI

- Not all game applications require all levels of AI.
- Board games like Chess require only the strategy level; the characters in the game don’t make their own decisions and don’t need to worry about how to move.
- On the other hand, some games might not have a strategy.
- These games are purely reactive, making their own simple decisions and acting on them.

World Interface (Sensing)

- Gather current world state: barriers, opponents, NPC, objects, …
- Needs limitations: vision, hearing, range, etc.

  Vision:

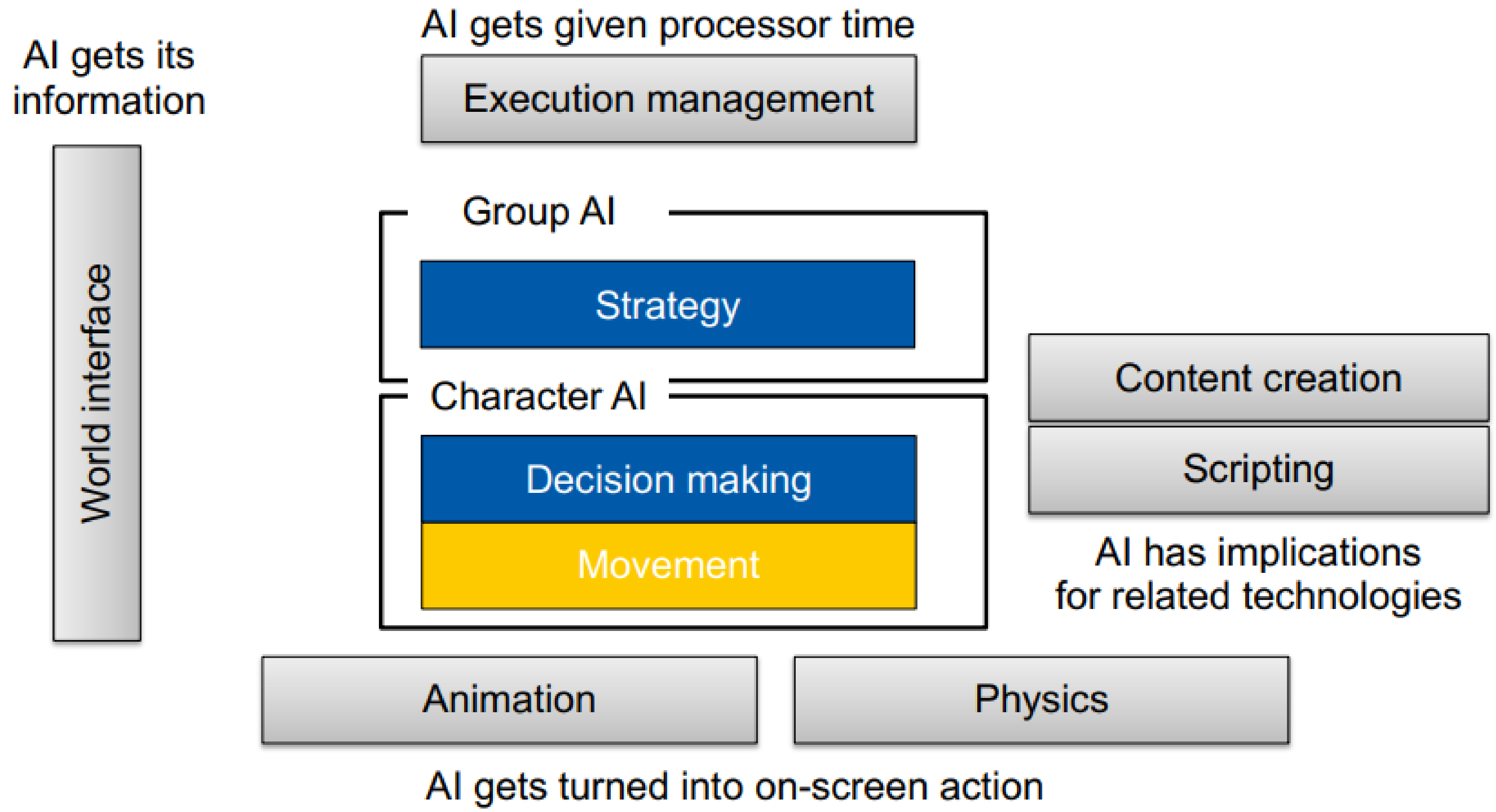
  - Check magnitude (ie- is it too far away?)

  - Check angle (ie- within 120° viewing angle?)
- Communication between players

  - Model sensing data from other agents

An NPC (non-player character) is a character in a game that is not controlled by the person playing the game, nor by any sort of AI. They just be there to fill the world. For example, in games with large cities, NPCs are used as citizens on the street, simply walking in a single direction or running away from you. They help make the game's world feel more realistic and alive.

Movement



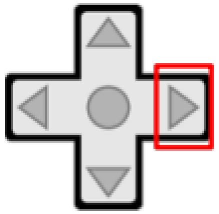
- Movement is the lowest level in the AI model.

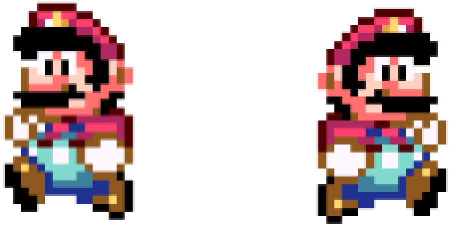
Movement

- Movement refers to algorithms that turn decisions into some kind of motion.
- The basic idea of this layer is to turn the character into a certain direction and giving it a speed.
- Often, the animation system (for playing the correct animation) and the physics system (to tell if obstacles have been hit and to handle the acceleration of the character) are involved in this step.
- There are two types of movement:

  - Kinematic: update the velocity (direction and speed) of a character immediately.

  - Dynamic: take into account velocity and direction change via accelerations

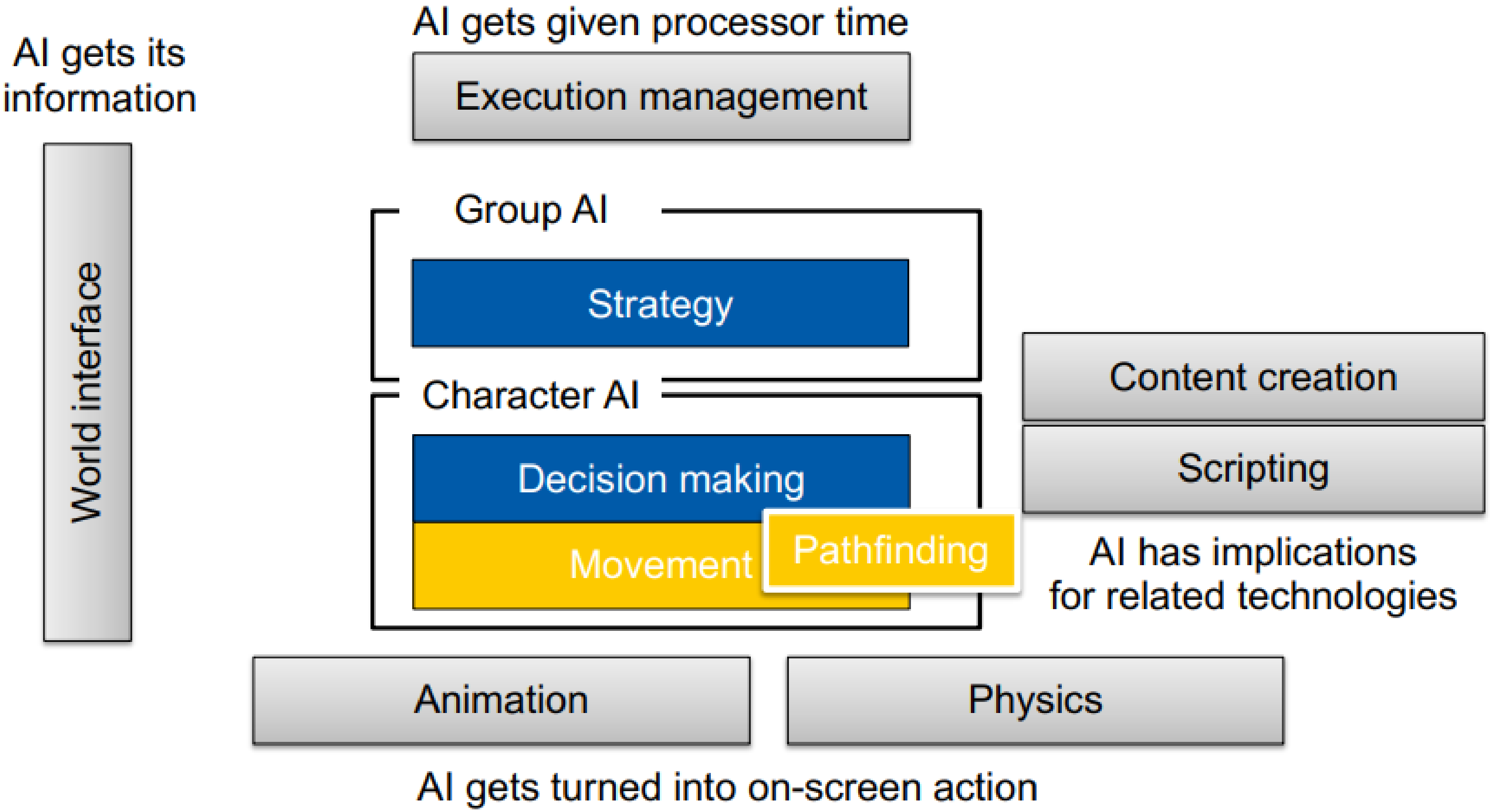




Kinematic

Dynamic

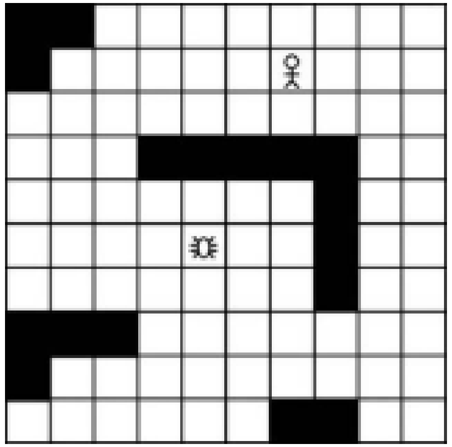
Pathfinding



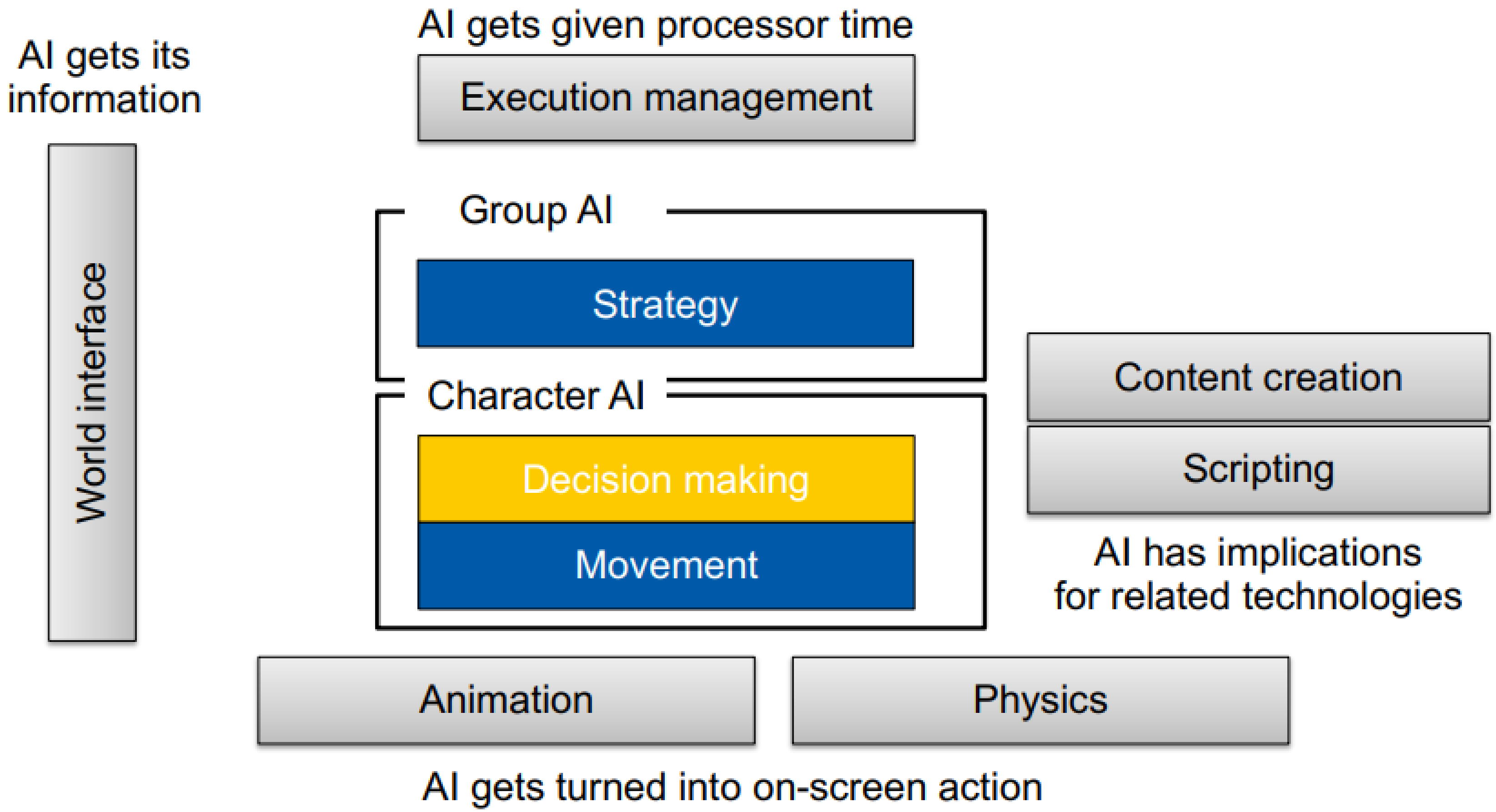
- Pathfinding is the next level in the AI model: it uses global information about the map to find a valid path from the current and target locations.

Pathfinding

- Generate a path for an AI to follow
- Based on a graph:
  - Basic connectivity: point B can be reached from A if there is a connection between them.
  - Costs: How “hard“ is it to get from A to B?
  - Directions: In which direction is the connection? (E.g. jumping down an unclimbable ledge)
- Ex. A* algorithm



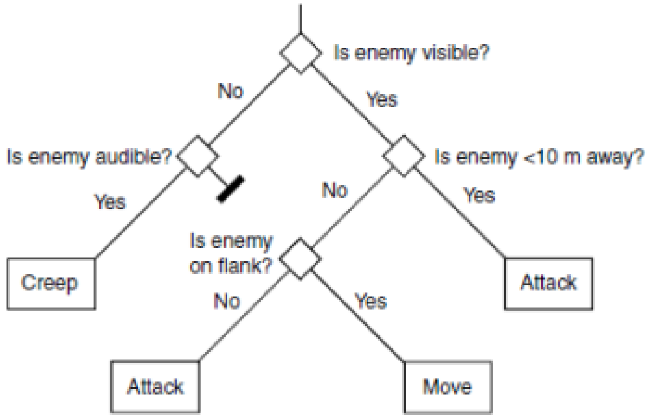
Decision Making



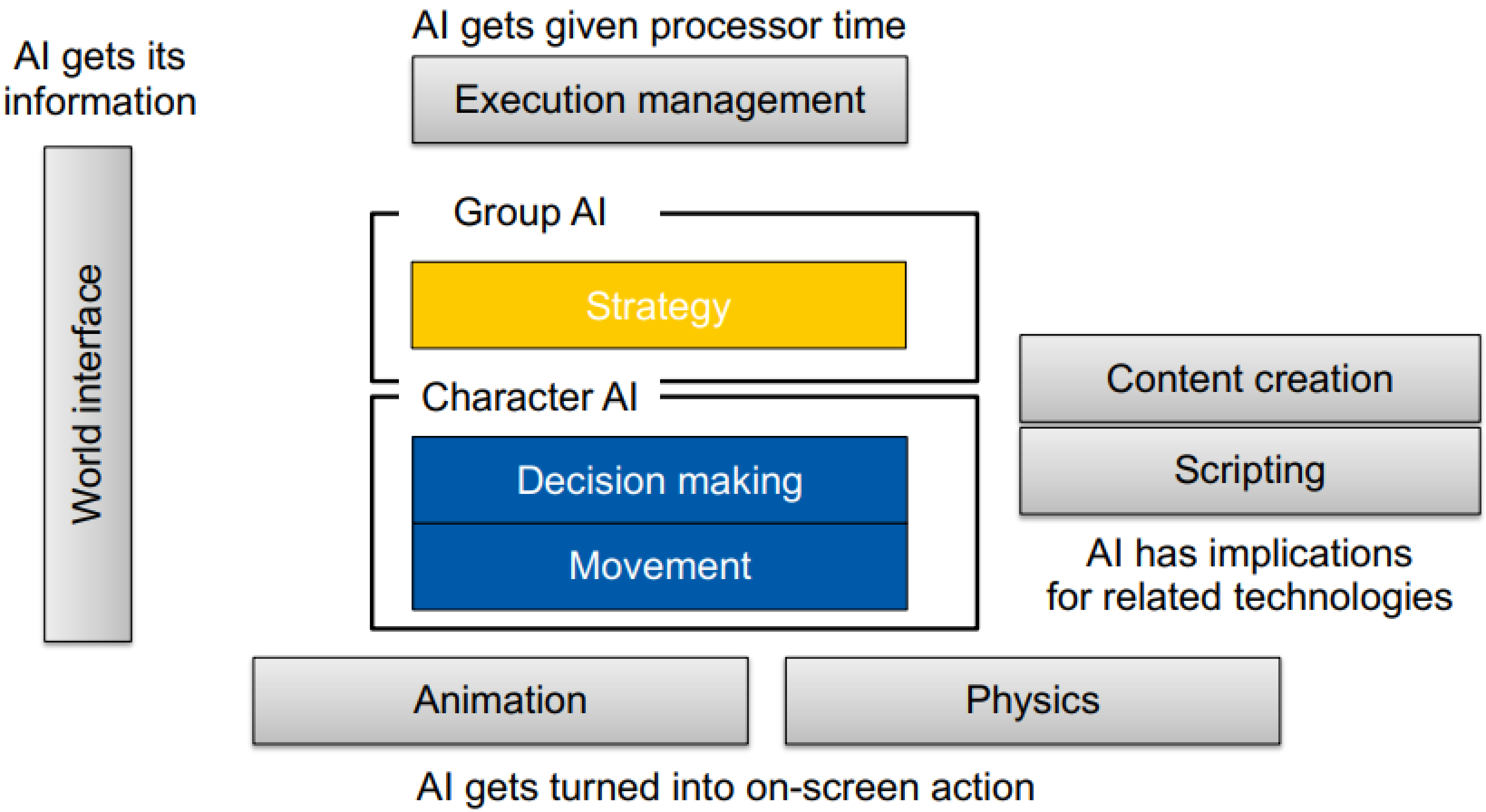
Decision Making

- This layer of AI refers to how individual AIs make their decisions (what to do next, where to move to, etc.)
- Each character has a range of different behaviors that they could choose to perform: attacking, standing still, hiding, exploring, patrolling, and so on.
- Structure decisions as a series of conditions and actions as leaves
- The decision making system needs to work out which of these behaviors is the most appropriate at each moment of the game.
- The chosen behavior can then be executed using movement AI
- Algorithms for Decision-Making:

  - MinMax Algorithm



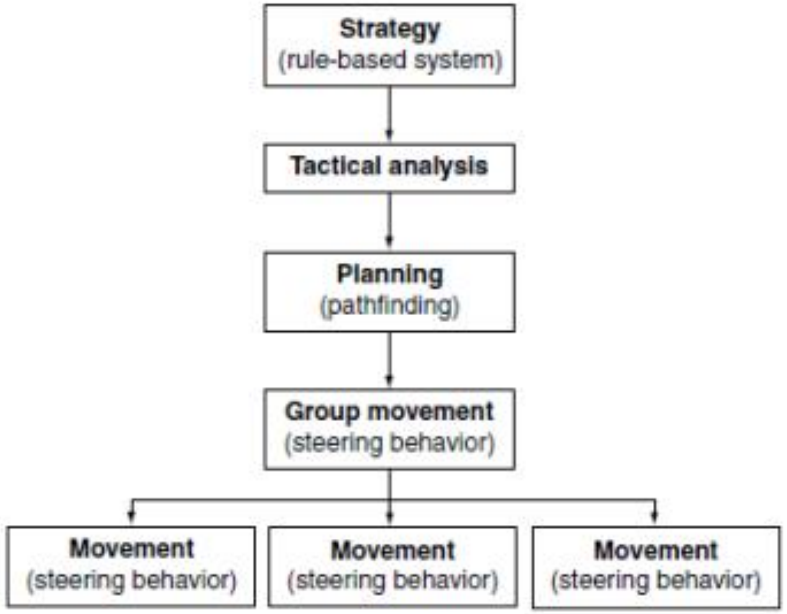
Strategy


AI gets its information
World interface
AI gets given processor time
Execution management
Group AI
Strategy
Character AI
Decision making
Movement
Content creation
Scripting
AI has implications for related technologies
Animation
Physics
AI gets turned into on-screen action

Strategy

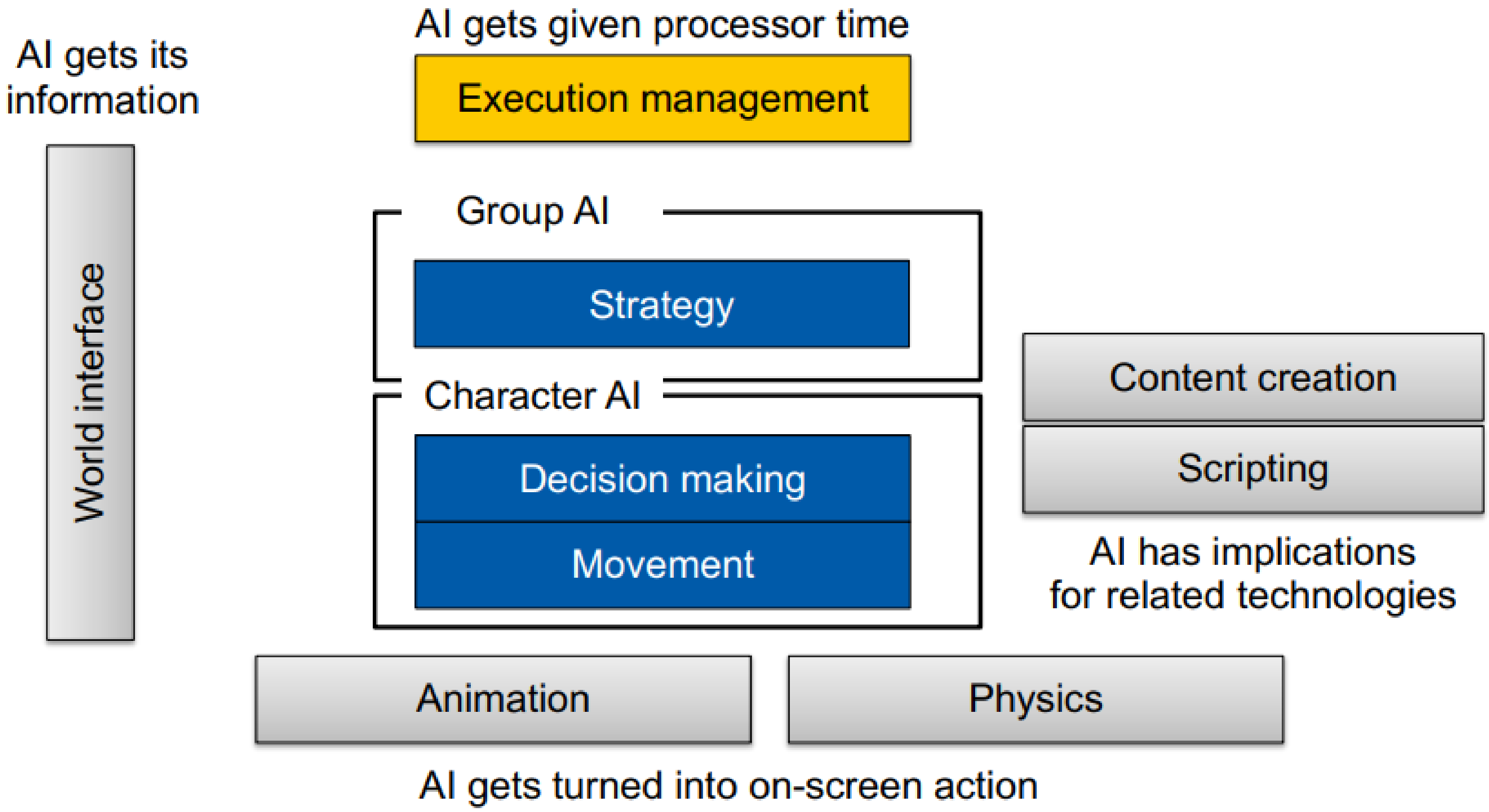
- Strategical AI is concerned with providing global information to AIs or with synchronizing the actions of several AIs.
- Analyze the game world:

  - How strong is the influence of each player?
- High-level commands passed downwards



Provides coordination between AIs

Execution Management



Execution Management

- It describes how the execution of AI tasks are scheduled.
- It manages tasks so that they are spread out over available frames instead of always slowing down every N’th frame.

Any Questions