HW7: Pac-Man API Specification
COMP 504 Chaos

1. Class AEntity
   a. edu.rice.comp504.chaos.model.entities.AEntity
   b. All Implemented Interfaces: java.io.Serializable, java.lang.Cloneable
   c. Direct Known Subclasses: Ghost, Pacman
   d. public abstract class AEntity
      extends java.lang.Object
      implements java.lang.Cloneable, java.io.Serializable
   e. Summary: An entity which is in the game. It could be a Pac-Man or a ghost.
   f. Fields:

Modifier and Type	Field	Description
private Coordinate	coord	The coordinate in the maze.
private Direction	dir	The orientation, intended moving direction.
private Coordinate	loc	The location (px, px) on the canvas.
private int	remainLength	When the entity has arrived at a regular spot and try to make a decision, the distance it still has to move.
private int	size	The length of the entity.
private int	speed	The speed of moving.
private Direction	startDir	The orientation of it when the game starts.
private Coordinate	startLoc	The location of it when the game starts.

   g. Constructor:

Constructor	Parameters	Description
AEntity(Coordinate startLoc, int speed, int size, Direction dir)	startLoc - the initial Coordinate of the entity. speed - the speed of the entity. size - the size of the entity.	The constructor. To produce an entity with some specific information.

	dir - the initial direction.	

h. Methods:

Modifier and Type	Method	Parameters	Returns	Description
(package private) Coordinate	computeIntendedDestination()		the intended destination	Compute the intended destination according to the location, speed and direction.
Coordinate	getCoord()		the Coordinate	Get the Coordinate.
Direction	getDir()		the direction	Get the direction.
private int	getItemOnIntendedDirection()		the item	Get the item on the next intended position.
Coordinate	getLoc()		the location	Get the location.
(package private) int	getSpeed()		the speed	Get the speed of the entity.
(package private) Direction	getStartDir()			Get the start direction.
private Coordinate	getStartLoc()		the initial location	Get the initial location.
(package private) void	move(int length)	length - the length		Move a certain distance.
(package private) void	move (Coordinate destination)	destination - a specific destination		Move to a specific destination.
(package private) void	moveOnRegularSpot()			Move on the regular spot.
(package	moveToCoor			Move to a

private) void	d(Coordinate destination)			specific Coordinate.
void	resetLoc()			Rest the entity to its initial location.
(package private) void	setDirection (Direction dir)	dir - the intended direction		Set the intended direction of the entity.
private void	setLoc (Coordinate loc)	loc - the location		Set the location.
void	setSpeed(int speed)	speed - the speed		Set the speed of the entity.

2. Class AGhostPersonality
    a. edu.rice.comp504.chaos.model.personalities.AGhostPersonality
    b. Direct Known Subclasses: Ambusher, Bashful, Chaser, Pokey
    c. public abstract class AGhostPersonality
       extends java.lang.Object
    d. Summary: The personality of a ghost. There will be four specific personalities.
    e. Fields:

Modifier and Type	Field	Description
private Pacman	pm	The reference of the Pac-Man. The ghost will need the location and orientation of Pac-Man to make decisions.

    f. Constructor:

Constructor	Parameters	Description
AGhostPersonality (Pacman pm)	pm - the reference of the Pac-Man.	Constructor. No matter what the concrete personality is, the ghost must know the Coordinate of the Pac-Man.

    g. Methods:

Modifier and	Method	Parameters	Returns	Description

Type				
(package private) Coordinate	getPacmanCoord()		the Coordinate	Get the Coordinate of the Pac-Man.
(package private) Direction	getPacmanDirection()		the direction	Get the direction of the Pac-Man.
abstract IGhostStrategy	think(Ghost context)	context - the ghost	the strategy	Come up with a strategy, according to different personalities.

3. Interface IGhostStrategy
    a. All Known Implementing Classes: RandomStrategy, TargetStrategy
    b. public interface IGhostStrategy
    c. Summary: The strategy determines how ghosts will take actions.
    d. Methods:

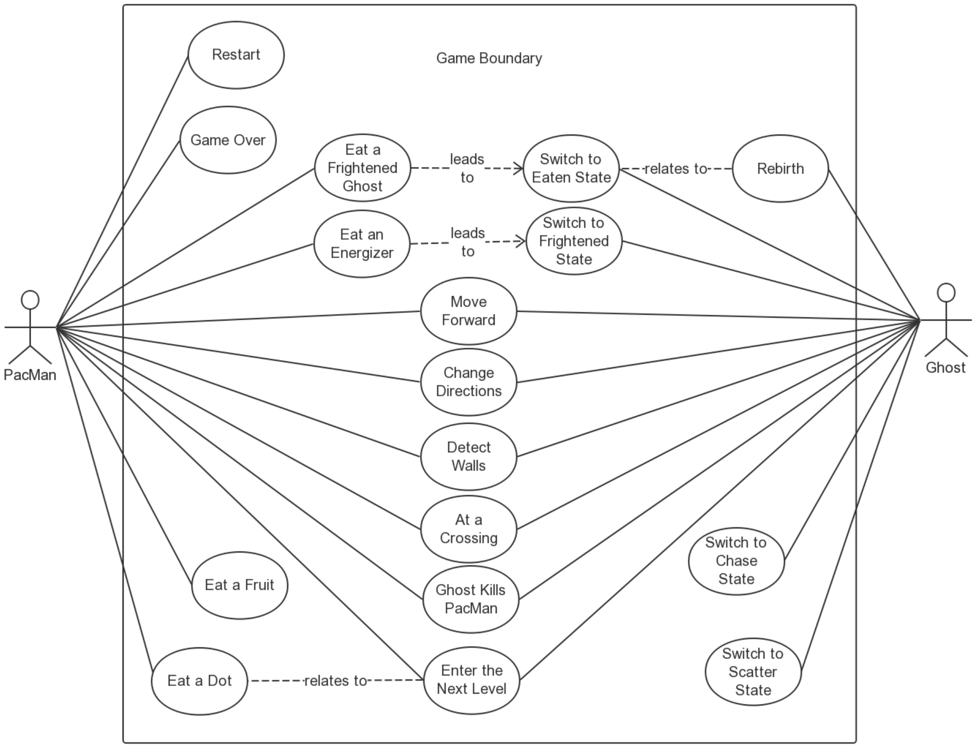
Modifier and Type	Method	Parameters	Returns	Description
Direction	choose (Coordinate current, java.util.List< Direction> availableDirections)	availableDirections - all available directions	the chosen one	Ghosts choose a direction from all available directions, according to different strategy.

4. Design Decisions
    a. User Extensibility
        i. The default mode is Single, and players could change it to Duo by clicking the button “Add Player” and there would be two Pac-Man on the map.
        ii. Players are able to choose their favorite map by clicking the drop-down menu and selecting it. There are two maps available right now: Google and Chaos.
    b. Design for Extensibility
        i. We store map and food information (dots, energizers, fruits) as matrix in the .txt files and when the game loads, the Game class would read them. So, in order to change the map, we just need to replace the map file; in order to change food information, we just need to replace the food map file.

ii. We use Object-Oriented-Programming, so that extra Pac-Man and ghosts could be added simply by calling their constructors.

5. Use Case

a. Use Case Diagram



b. Use Cases

Name	Move Forward
Identifier	UC01
Actors	Pac-Man, Ghost
Description	Entity moves towards its direction.
Triggers	Time.
Preconditions	The game is started. There is no wall ahead of the entity.
Post Conditions	The location of the entity changes.
Frequency	Once every 0.1 seconds.

Basic Flow	1. The entity keeps moving.
Alternates	N/A
Exceptions	When there is a wall ahead, the entity stops.

Name	Change Directions
Identifier	UC02
Actors	Pac-Man, Ghost
Description	Change the moving direction of the entity.
Triggers	N/A
Preconditions	The game is started. There is no wall in the new direction.
Post Conditions	The entity turns to a new direction.
Frequency	Around once every 5 seconds.
Basic Flow	1. If the player enters “↑”, the entity goes up. 2. If the player enters “↓”, the entity goes down. 3. If the player enters “←”, the entity goes left. 4. If the player enters “→”, the entity goes right.
	If there exists a second player, use “W” as up, “S” as down, “A” as left, “D” as right. For ghosts, the new direction is based on their moving states, their personalities, and the target.
Exceptions	When there exists a wall in the new direction, this operation does not work.

Name	Detect Walls
Identifier	UC03
Actors	Pac-Man, Ghost
Description	Detect if there is a wall ahead.
Triggers	N/A
Preconditions	The game is started.

Post Conditions	Return a Boolean value.
Frequency	Once the entity changes its location or direction.
Basic Flow	1. Get the direction and location of the entity. 2. Return whether there is a wall ahead.
Alternates	N/A
Exceptions	N/A

Name	At a Crossing
Identifier	UC04
Actors	Pac-Man, Ghost
Description	Decide whether the entity is at a crossing.
Triggers	Around once every 5 seconds.
Preconditions	The game is started.
Post Conditions	Return the crossing situation.
Frequency	Once the entity changes its location.
Basic Flow	1. Get the location of the entity. 2. Return the crossing situation.
Alternates	N/A
Exceptions	N/A

Name	Ghost Kills Pac-Man
Identifier	UC05
Actors	Pac-Man, Ghost
Description	The Ghost eats the Pac-Man.
Triggers	The Ghost and the Pac-Man overlap with each other.
Preconditions	The game is started. The ghost is in Scatter or Chase state.
Post Conditions	The Pac-Man dies.
Frequency	It depends.

Basic Flow	1. The Pac-Man and the ghost overlap with each other. 2. The Pac-Man dies.
Alternates	N/A
Exceptions	N/A

Name	Enter the Next Level
Identifier	UC06
Actors	Pac-Man, Ghost
Description	The game enters the next level, which is more difficult.
Triggers	All dots are eaten up by the Pac-Man
Preconditions	The game is started.
Post Conditions	The game enters the next level.
Frequency	It depends.
Basic Flow	1. When all dots are cleared, the game goes to the next level.
Alternates	N/A
Exceptions	N/A

Name	Eat a Dot
Identifier	UC07
Actors	Pac-Man
Description	Move the Pac-Man and eat a dot.
Triggers	The Pac-Man and the dot are overlap with each other.
Preconditions	The game is started.
Post Conditions	When all dots are cleared, the game goes to the next level.
Frequency	No more than 240 times each level.
Basic Flow	1. Pac-Man moves. 2. When the Pac-Man is over the dot, the dot is

	eaten. 3. When all dots are cleared, the game goes to the next level
Alternates	N/A
Exceptions	N/A

Name	Eat a Fruit
Identifier	UC08
Actors	Pac-Man
Description	The Pac-Man eats a fruit.
Triggers	The Pac-Man and the fruit are overlap with each other.
Preconditions	The game is started. There are some fruits on the map.
Post Conditions	The Pac-Man gets some bonus credits.
Frequency	It depends.
Basic Flow	1. Pac-Man moves. 2. When the Pac-Man is over the fruit, the fruit is eaten.
Alternates	N/A
Exceptions	N/A

Name	Eat a Frightened Ghost
Identifier	UC09
Actors	Pac-Man
Description	Pac-Man eats the ghosts and changes ghosts into eaten mode.
Triggers	The Pac-Man and the ghost are overlap with each other.
Preconditions	The game is started. Ghosts are in the frightened state.
Post Conditions	Frightened ghosts turn into the eaten state and run back to the ghost home. Pac-Man gets additional points.

Frequency	It depends.
Basic Flow	1. Move the Pac-Man and overlap itself with the frightened ghosts. 2. Frightened ghosts which are eaten turn into the eaten state.
Alternates	N/A
Exceptions	N/A

Name	Eat an Energizer
Identifier	UC10
Actors	Pac-Man
Description	Move the Pac-Man and eat an energizer.
Triggers	The Pac-Man and the energizer are overlap with each other.
Preconditions	The game is started. There are some energizers left on the map.
Post Conditions	All ghosts turn into the frightened state.
Frequency	No more than the-number-of-energizers times each level.
Basic Flow	1. Pac-Man moves. 2. When the Pac-Man is over the energizer, the energizer is eaten.
Alternates	N/A
Exceptions	N/A

Name	Game Over
Identifier	UC11
Actors	Pac-Man
Description	The game is over.
Triggers	UC05
Preconditions	There is no life left for the Pac-Man.
Post Conditions	Words “Game Over” appears.

Frequency	It depends.
Basic Flow	1. Words “Game Over” appears.
Alternates	N/A
Exceptions	N/A

Name	Restart
Identifier	UC12
Actors	Pac-Man
Description	Restart the game.
Triggers	The Pac-Man is eaten by a ghost.
Preconditions	There are still lives left.
Post Conditions	Pac-Man goes back to its birth point. Ghosts go back to their home. The number of lives minus 1
Frequency	It depends.
Basic Flow	1. Pac-Man goes back to its birth point. 2. Ghosts go back to their home. 3. The number of lives minus 1
Alternates	N/A
Exceptions	N/A

Name	Switch to Scatter State
Identifier	UC13
Actors	Ghost
Description	Ghost switches to Scatter state.
Triggers	It is the scatter time period now.
Preconditions	The game is started.
Post Conditions	Ghost switches to Scatter state.
Frequency	About once every 10 seconds.

Basic Flow	1. The ghost changes its target.
Alternates	N/A
Exceptions	N/A

Name	Switch to Chase State
Identifier	UC14
Actors	Ghost
Description	Ghost switches to Chase state.
Triggers	It is the Chase time period now.
Preconditions	The game is started.
Post Conditions	Ghost switches to Chase state.
Frequency	About once every 10 seconds.
Basic Flow	1. The ghost changes its target
Alternates	N/A
Exceptions	N/A

Name	Switch to Frightened State
Identifier	UC15
Actors	Ghost
Description	Ghost switches to Frightened state.
Triggers	UC10
Preconditions	The game is started.
Post Conditions	Ghost switches to Frightened state.
Frequency	It depends.
Basic Flow	1. The ghost changes its appearance. 2. The ghost changes its speed.
Alternates	N/A
Exceptions	N/A

Name	Switch to Eaten State
Identifier	UC16
Actors	Ghost
Description	Ghost switches to Eaten state.
Triggers	UC 09
Preconditions	The game is started.
Post Conditions	Ghost switches to Eaten state.
Frequency	It depends.
Basic Flow	1. The ghost changes its appearance. 2. The ghost goes back to ghost home.
Alternates	N/A
Exceptions	N/A

Name	Rebirth
Identifier	UC17
Actors	Ghost
Description	The Eaten ghost recovers in the ghost home.
Triggers	The ghost is in the ghost home.
Preconditions	The ghost is in the Eaten state.
Post Conditions	The ghost recovers.
Frequency	Once the ghost is eaten.
Basic Flow	1. The ghost changes back to its original appearance.
Alternates	N/A
Exceptions	N/A