CRC Cards for Ms. Pacman

Group: Breakfast

GameObject	
Interface that descries the common essential behaviors of all characters in the game. This includes draw() and update()	MovingObject StationaryObject

StationaryObject	
The parent class for all game object which can not move.	NormalPellet PowerPellet

ArcadeList	
A list of Arcade objects. In a pacman game lifecycle, there may be more than one arcade used. This list manages to control and update each individual Arcade.	Arcade

Arcade	
Hold 1 Arcade object that describes the layout of the map. The Arcade object provides a coordination system and constraintes so that the game objects such as the pacman, pellets, and cherries can be hold upon.	ArcadeBlock

ArcadeBlock	
The ArcadeBlock is most fundamental building block to an Arcade. The ArcadeBlocks are of the same shape but with different coordinates and properties.	N/A

MovingObject	
The parent class for all game object which can not move.	NormalPellet PowerPellet

Pacman	
Save all 4 Pacman faces/directions into an array and be able to return the correct face with animation based on the direction argument passed be the caller object. Eat Pellets.	Ghost PelletCell PowerPelletCell BitmapDivider

GhostList	
It stores and renders all 4 ghosts,	Arcade Ghost BitmapDivider

Ghost	
ramdomized Ghost's movement without user input. Try to kill Pacman by Collision	Pacman CollisionDetector

PacmanActivity	
initialize the game and set up all screen configuration (fill sceen, and maybe landscape mode). Handle pause and resume situations.	PacmanGame

PacmanGame	
initialize the all images, such as the Pacman, ghosts, obstacles, ect. Run the game until all 3 lives of the pacman have been consumed. be able to move the Pacman. Be able to detect collision.	Pacman Ghost ScoreSystem PlayerInput GhostControl

GameObjectCollection	
The place to collect and udpate all GameObjects related to one Arcade	Arcade

PlayerInput	
Be able to register the user input through swipes (up, down, left, and right) and return the command back to the caller object (PacmanGame)	N/A

PelletList	
Has the collection of pellets, i.e. pelleteCell, powerpelletCell. Initialze the location of each cell, and their points. Visibility states.	PelletCell PowerPelletCell ArcadeList PackmanGame TwoTuple

PelletCell	
Each cell has type of pellets, State: uneaten/visible, eaten/unvisible. Initializes with points, if gets eaten then decrement the points	GameObject

PowerPelletCell	
Inherites the pelletcell, but has different points, look and locations.	PelletCell

Cake	
Independent class for dispalying the Cake onto the screen. Set a timer fot the when to be displayed and for how long.	MotionInArcade TwoTuple Arcade

ScoreSystem	
Seperate class for keeping the scores of the pacman game, will be initialized in Pacman Game, Keep tract of the scores that the pacman is gaining. Based on the different pellets that pacman eats, score different score will be added. Also have a boolean value for cherry, if cherry is eaten, everything gets eaten after will counts double of its original points.	pacmanGame Pellets PowerPelletCell Cake

UserInput	
This objects works as a listener to the touch events on the screen. It extract essential information from each touch event and keep the information so that functtions in other threads can use it.	listener (Java util)

JsonParser (ArcadeDecoder)	
The Aarcade information is kept in a JSON file. This object reads the file and calls the construtor of ArcadeList	JsonReader (Android IO)

BitmapDivider	
Divide one bitmap into multiple based on row and column	N/A

WelcomeView	
This is the first class that gets displayed onto the player's display, and allows the player to choose the game mode (earsy, normal, and hard)	N/A

CollisionDetector	
This object takes 2 Obstacle objects and determine if they have collided into each other.	Obstacle

Obstacle	
This object takes a game object and transorm it into a obstacle. Every GameObject can be considered as an Obstacle. This class will assist collision detection. The reason we want this class is to calculate the dimension of the bounding box of a moving or stationary GameObject.	N/A

GameMode	
It changes the speed of the entire game based on the player initial input at WelcomeView page. Options are: easy (slow), noraml (faster), and hard (fast).	WelcomeView

NextMotionInfo	
Use this class to return updated motion information. if the motion information is not valid, the gameObject recive this motion info can not change its motion status.	TwoTuple

MotionInArcade	
Detect the motion in arcade is valid or not. Project current motion onto the arcade block. Current motion is not only decided by current position but also current direction.	TwoTuple Arcade

NavigationButtons	
This is the pacman contorl which based on the input of player's touch on the button to control the pacman	N/A

UserInput	
we take user input and process all other stuff based on this object.	N/A

TwoTuple	
contains the X and Y coordinate.	N/A

ConsoleReader	
This is an utility class that reads console input (keyboard input). The reason we do this is to use keyboard to control Pacman.	N/A

GhostBehavior	
We have 4 ghost in the game, and each of the ghost acting differently. (i.e. the red ghost will trace the pacman.)	Ghost

ChaseBehavior	
target directly on the pacman	

ChaseFrontBehavior	
target a distance infront of the pacman	

EscapeBehavior	
escape from the pacman	

KilledBehaviour	
go back to the reborn point	

PredictAndChaseBehaviour	
target on the extension line from the red ghost to the pacman	

MotionInfo	
describes the current motion status	TwoTuple

StaticInfo	
describes the current location status	TwoTuple

Records	
Modify local JSON to keep the highest records	Record

Record	
A record that keeps User name in string and score in int	

Description

CRC (class, responsibilities, and collaboration) cards are a visual brainstorming technique often used in the early stages of system development. Used in Extreme Programming, this brainstorming method can help you understand the different objects that your team will be creating and how these will interact with other people, places, things, or functions. CRC cards quickly help to uncover potential problems or weaknesses in the system.

To use this template, fill in the cards by first putting the class name in the header. A class represents a collection of similar objects (an object being defined as any person, place, thing, event, etc. relevant to the system). Next, list that class's responsibilities (i.e. anything that class knows or does) in the left-hand column. List collaborators (other classes) this class needs to complete its responsibilities (because it either doesn't know or can't do it itself) in the right-hand column.

Move the cards around and place those that are closely related next to each other. You can also add color to further clarify close relationships.

Tutorials

(Hold Shift + ⌘ or Ctrl, then click)

Watch Lucidchart basic tutorials

See an example

(Hold Shift + ⌘ or Ctrl, then click)