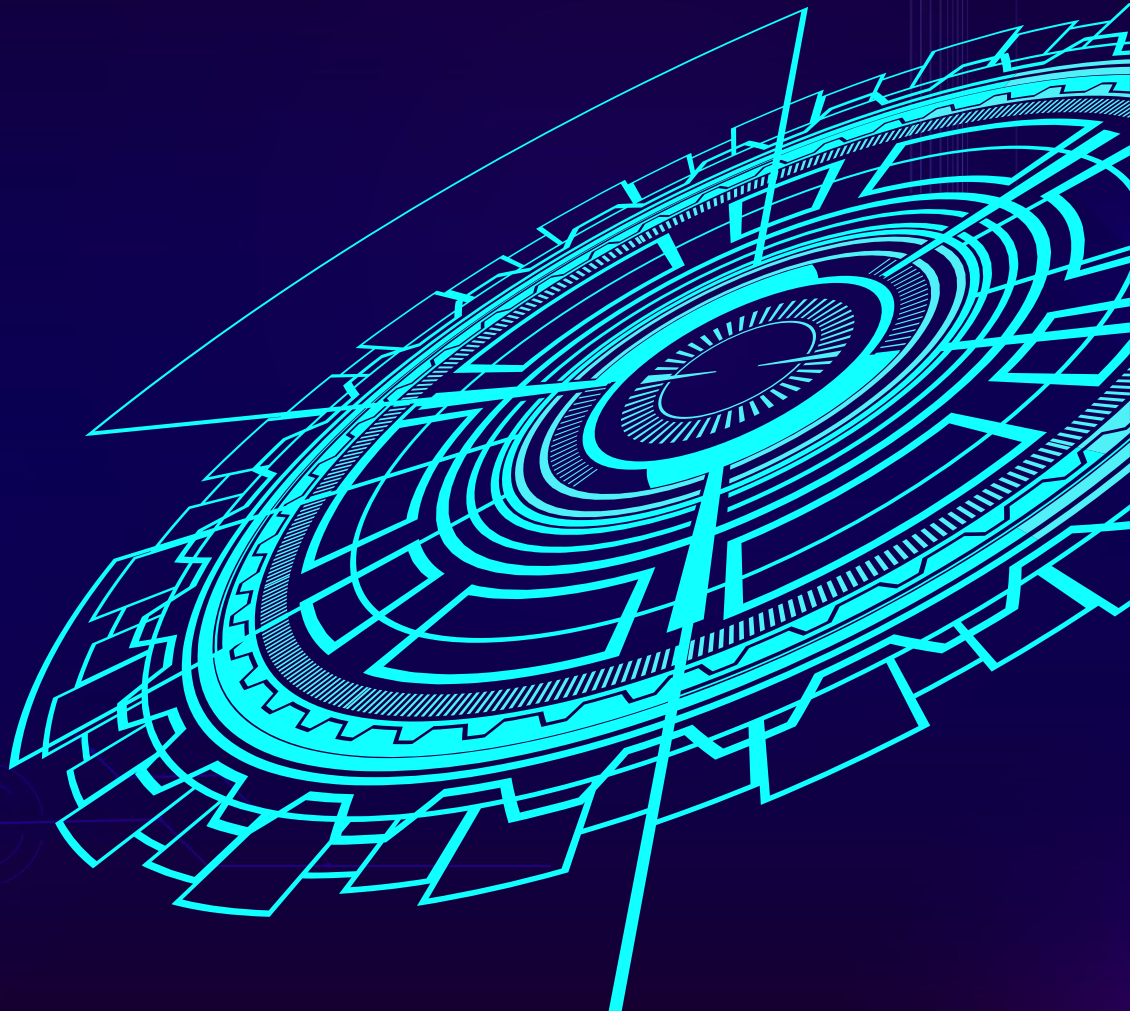
Pac-Man Game

By Zoom Coders

Final Presentation
Rice University, COMP 504 Fall 20



The Team

Alex Liapis

Project Team Lead

Mi Yang

Technical Lead

Howard Wang

Documentation Lead

Jiamin Liu

Developer

Minyu Jiang

Developer

Dingjia Li

Developer

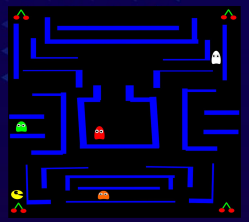
Use Cases

***Pac-man*:**

1. Will move in a direction based on directional keys.
2. pac-man will appear at the other side of the board after going through tunnel.
3. When pac-man eats large dots, the ghosts turn dark blue and then start flashing (blue and white) for a period of time.
4. If pac-man collides with a flashing ghost, the ghosts become two eyes and travel quickly to the square box.
5. If pac-man collides with a non dark blue or non-flashing ghost, Pac-Man loses 1 life, pac-man has 3 lives.

***Ghost*:**

1. Ghosts starts in the middle and will use some behavior to move toward or away from pac-man.
2. Ghosts will have different behavior, different colors and velocities.



***Board*:**

1. There are “tunnels” on sides of the wall where if the pac-man goes in, it comes out of the other side.
2. Each board will have approximately 240 small dots, 4 big dots in corners.
3. Periodically a fruit worth 100 points will appear on the screen.
4. There will be info like a score, dots consumed, lives left in the board.
5. Game ends and will display “game over” on the board when the pac-man loses 3 lives, player wins when eats all dots in 3 levels
6. 2 levels (different board) when pac-man eats all dots.
7. level 2 (board) becomes more difficult.

API SPEC

ACharacter (abstract class):
An abstract character of the pac-man game. Could be a ghost or pac-man.

	Member fields
vel (Point)	The velocity of the character.
updateStrategy (IUpdateStrategy)	The movement strategy of the character.
collideStrategy (ICollideStrategy)	The collision strategy of the character.
color (String)	The color of the character.
direction (int)	The direction the character is facing
originalLoc (Point)	The original location of the ACharacter.
size (int)	The size of the ACharacter.

AItem (abstract class):
An abstract item in the game. Could be a dot, big dot, fruit, or a wall.

	Member fields
color (String)	The color of the item.
score (int)	How much score the item gives.
isEaten (boolean)	Whether the item has been eaten.

GameLevel:
A game level in the pac-man game.

	Member fields
levelCount (int)	The number of the level.
walls (List<Walls>)	The list of walls of the game board for the level.
fruits (List<fruits>)	The list of fruits for the level.
dots (List<Dot>)	The list of dots for the level.
bigDots (List<bigDots>)	The list of big dots for the level.
zoom (Zoom)	A special zoom item.

PaintObj (abstract class):
An object that will be drawn in the pac-man game world.

	Member fields
loc (Point)	Location of the paint object.
name (String)	Name of the paint object.

ICollideStrategy (interface):
Interface for ACharacter strategies that determine the collision behavior between characters in the game.

	Methods
getName()	Get the name of the collision strategy. Return: String name.
collideState(APaintObj other)	Update the state of collision for the character. Parameters: APaintObj other.

IUpdateStrategy (interface):
IUpdateStrategy interface is used to determine the behavior of a paint object in the canvas over time.

	Methods
getName()	Get the update strategy name. Return: String name.
updateState(ACharacter context)	Update the state of the ACharacter. Parameters: ACharacter context.
updateState(ACharacter pacman, ACharacter context)	Update the state of the ACharacter. Context is the character to apply strategy to.

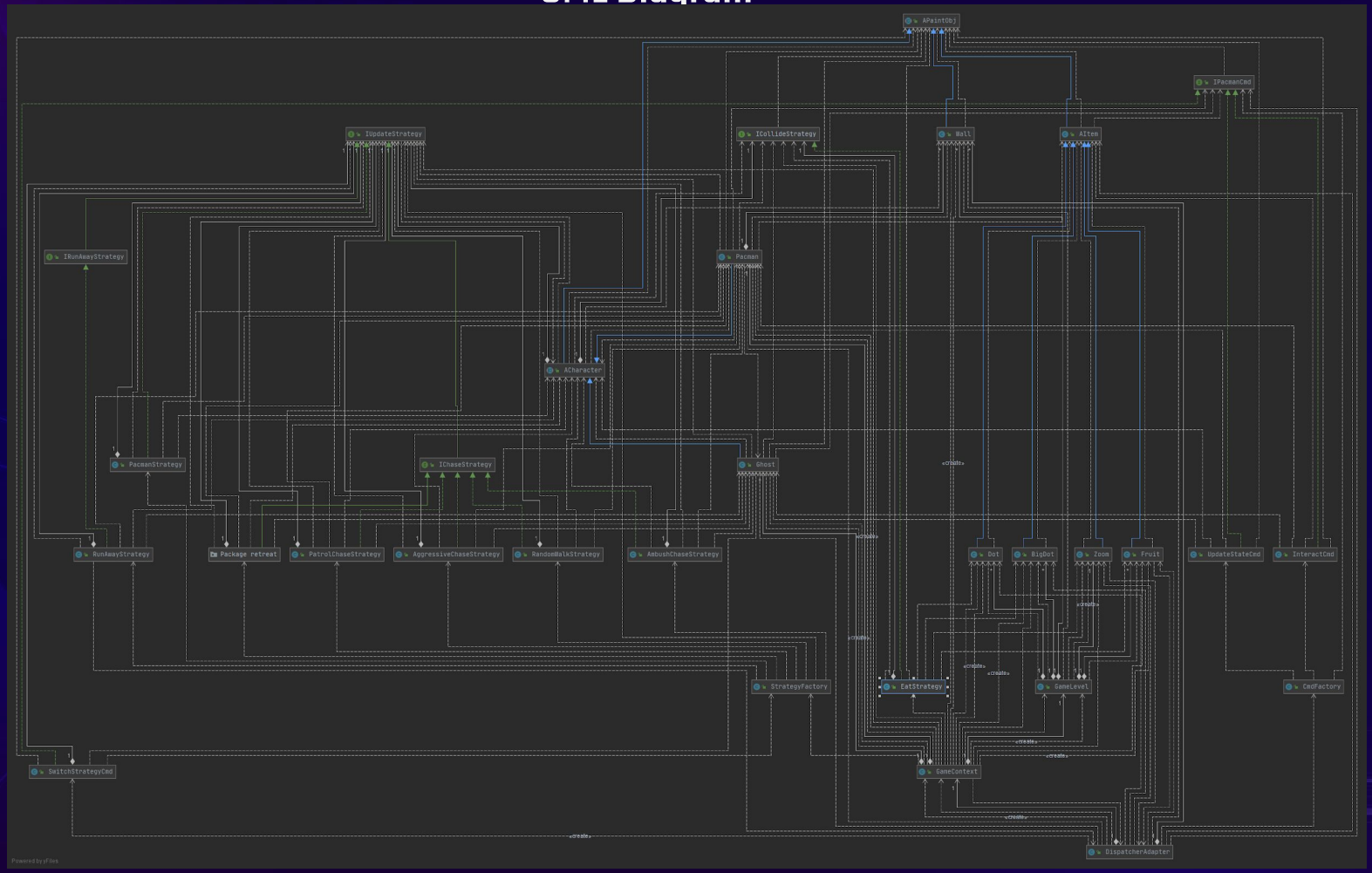
GameContext:
The pac-man context and game board for the game.

	Member fields
levelInstance (GameLevel)	Level instance either 1 or 2.
pacman (Pacman)	The pacman object.
Ghosts (List<Ghost>)	The list of the ghosts for the game.
Status (int)	The status of the game, whether it's paused.
maxLives (int)	The maximum number of lives for the pac-man.
currentScore (int)	The current score of the game.
highestScore (int)	The highest score of the game.
levelCount (int)	Total counts of the levels.
numberOfGhosts (int)	The number of ghosts.
numberOfFruits (int)	The number of fruits.
isZoomAvailable (boolean)	Whether the zoom item is available.
ghostScore (int)	The ghost score.
nextFruitIndex (int)	The index of the next fruit.
fruitsActivated (int)	The number of fruits activated so far.
fruitActiveTime (int)	How long fruits are active for.
zoomEffectTime (int)	How long zoom items are active for.
inEffect (boolean)	Whether the item is in effect.
effectLeftTime (int)	Time left on the effect
scoreFactor (int)	The multiply factor for the score.
ghostFlashingTime (int)	How long ghosts keep flashing.
maxGhosts	Maximum number of ghosts.

Design Decisions

- Our game will be rendered inside a canvas, with buttons and menus to set the level, reset, etc.
- The canvas will update according to the user settings at start. Highest score, current lives, etc will be updated as those events occur on the canvas.
- Pacman & ghosts are both subclasses from ACharacter.
- Character direction is decoupled from velocity so that the magnitude of the character speed can be separately changed from the directions. And velocity can be used to store the next moving direction.
- Pacman & ghosts have two versions of the same sprite in order to create the animation of movement. These animation frames are cycled between in the view.
- Previous location data & redraw parameters are used to ensure that old ghost positions are cleared & ghosts do not erase previous dot, big dot, & fruit drawings on the canvas when passing over those items.
- Each ghost has its own unique default chasing strategy. This strategy is changed to a run away strategy when pacman eats a big dot or return to base strategy when eaten.
- GameContext class holds the game information: level, score, etc. The Dispatch Adapter ties these all together for each game session.
- Collisions between objects are assigned a specific command according to the type of collision.
- A general update state command is used to refresh the character locations & parameters.

UML Diagram


AVaintObj
IPacmanCmd
IUpdateStrategy
ICollideStrategy
Wall
AItem
IRunAwayStrategy
Pacman
ACharacter
PacmanStrategy
IChaseStrategy
Ghost
«create»
RunAwayStrategy
Package retreat
PatrolChaseStrategy
AggressiveChaseStrategy
RandomWalkStrategy
AmbushChaseStrategy
Dot
BigDot
Zoom
Fruit
UpdateStateCmd
InteractCmd
StrategyFactory
EatStrategy
GameLevel
CmdFactory
SwitchStrategyCmd
GameContext
DispatcherAdapter

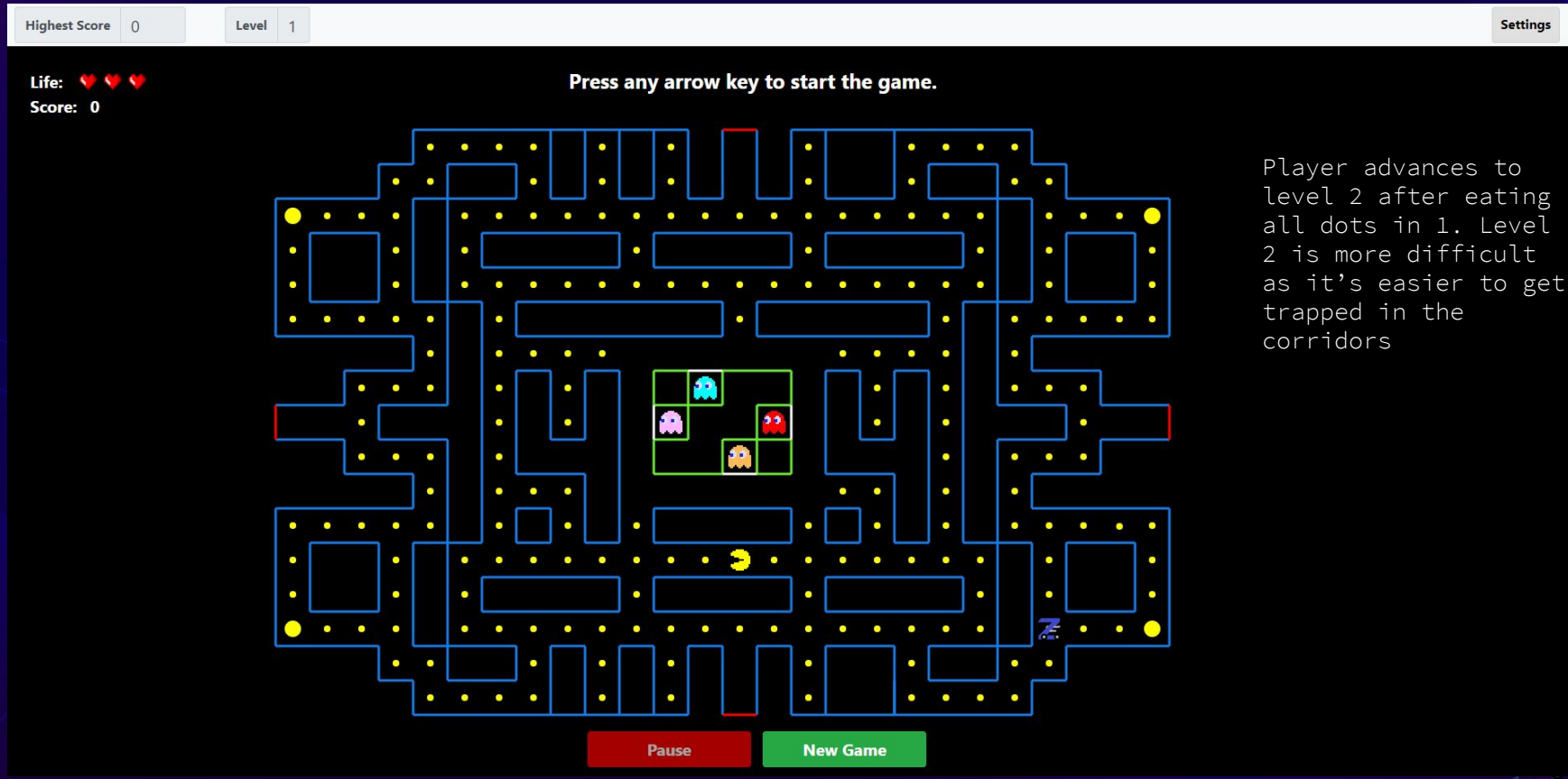
Level 1

Game UI

Will automatically advan
to level 2 when all dots
eaten

Level 2

Game UI



Player advances to level 2 after eating all dots in 1. Level 2 is more difficult as it's easier to get trapped in the corridors

Ghosts

Game UI

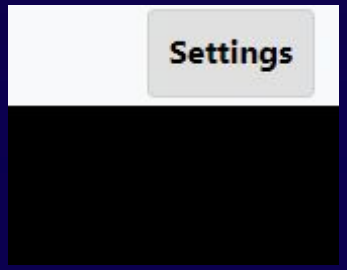
flashing ghosts

4 different types, each with different behaviors. Ghosts start in the middle box when game starts. When the pac-man dies, ghosts will also respawn from here.

Game UI

Game Statistics

Game Settings



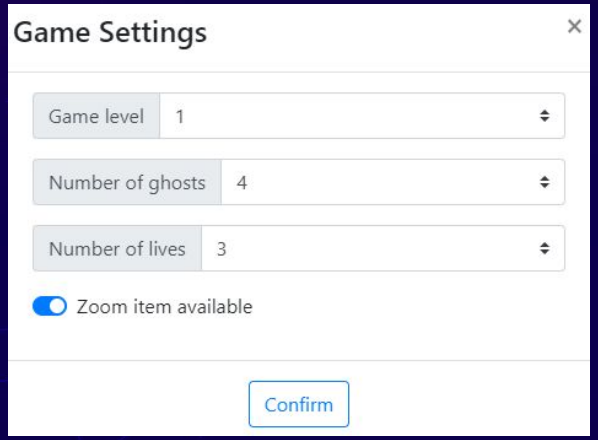
Game settings in upper right corner

Helpful notifications

Game Status Buttons

The user can pause, continue, or restart a new game

Game UI



There is a settings menu where the player can chose the level to start game at, number of ghosts, and number of lives.

The user can chose to include a “Zoom item”.

Special “Zoom item” where if the pac-man eats it, the score is doubled for 10 seconds.

ZOOM! ZOOM! ZOOM! Receive 2x score for next 10 seconds!

THANKS

We hope you enjoyed our presentation!

Enjoy our game in the link when it’s ready!

pacman-team-zoomcoders
.herokuapp.com