

Group Bigfoot: Final Pac-man Game

Swamy Reddi (team lead)
Frank Feng (tech lead)
Nicole Du (doc lead)
Anand Gadsing (dev)
Daniel Kuo (dev)



BigFoot's Pac-Man

Agenda:

- History of Pacman
- Sneak peek to our game
- Overview of the Technical details
- API End-points
- Demo



src:

<https://thelogicalindian.com/technology/pac-man-274>
23

History & About PacMan

- Pac-Man was originally Called “Puck-Man” (Puck from Hockey)
- Created by Toru Iwatani and his team in 1980.
- The inspiration for the Pac-Man character was the image of a pizza with a slice removed. Further, Iwatani has said he also rounded out the Japanese character for mouth, **kuchi** (Japanese: 口).
- Each of the Ghosts have specific nature
- The Goal of this game was to attract not only teenage boys and grown-up men to play his game, but an entire family, comprising of women and children as well, as games at that time mostly had some theme related to shooting or sports.



src: <http://nickcassway.com/designblog/?p=2443>



Sneak Peek BigFoot's Pac-Man

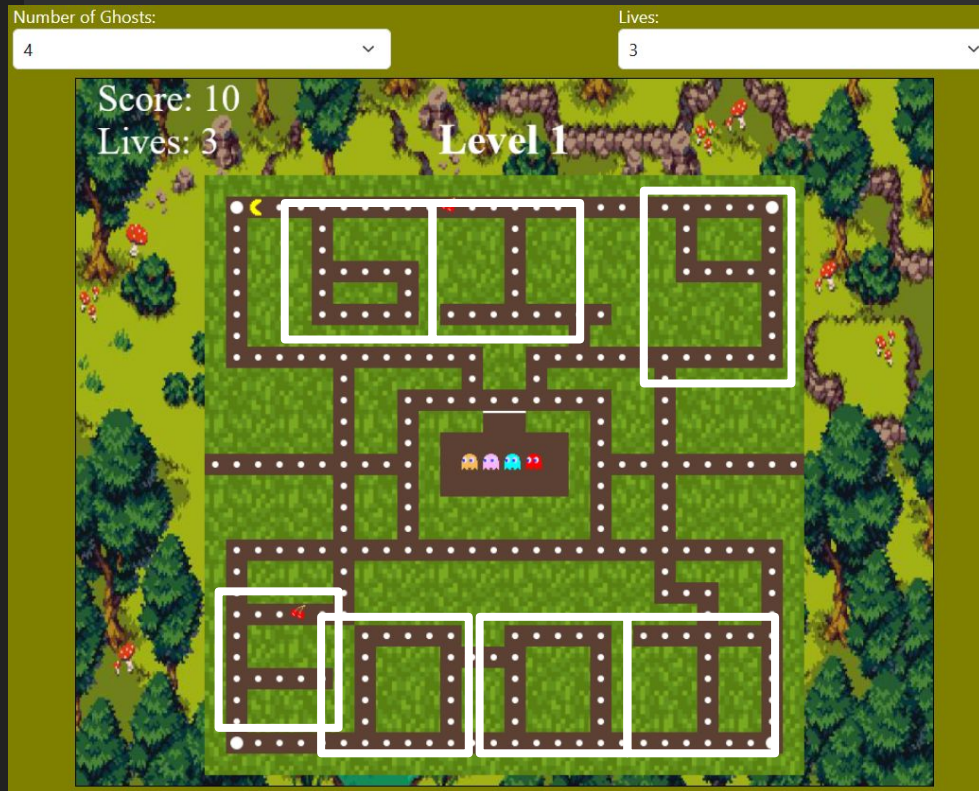
- Two game level, user can set levels
- Map layout designed with “bigfoot”
- Extra points from big foot icon like fruit



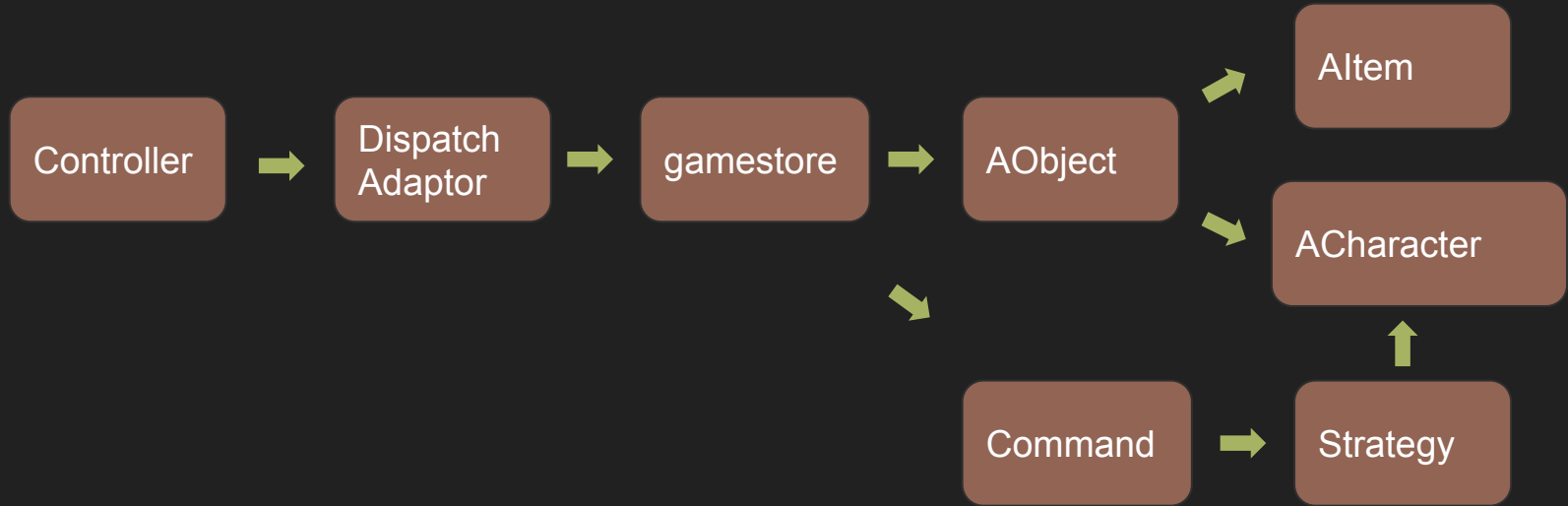
- Two behavior for ghosts

Up Next:

- Overview of the Technical details



Structure Overview



GameStore

ACharacter
Altem

- Maintain state of pacman world
 - Work with ACharacter and Altem abstract classes (property change listener) to manage the update of pacman world
 - Use 2D array to represent the map
-
- Pacman / Ghost movement
 - Pacman / Ghost interactions
 - Pacman / items interaction
 - Score / lives updated

Use cases

GameStore		
int	eatenDots	
int	updatePeriod	
ACharacter	pacman	
int	ghostScore	
int	ghostFlashingTime	
int	maxLives	
int	timeElapsed	
int	numberOfFruits	
int	endTime	
int	maxGhosts	
int	numberOfGhosts	
PropertyChangeSupport	pcs	
int[]	portals	
int	levelCount	
int[][]	layout	
int	highestScore	
List<ACharacter>	ghosts	
int	currentScore	
void	setGhostScore(int)	
void	removeAll()	
int	getNumberOfFruits()	
int	getGhostScore()	
void	setCurrentScore(int)	
void	removeDot(Altem, boolean)	
boolean	isGameOver()	
GameStore	updateStore(String)	
ACharacter	getPacman()	
void	setHighestScore(int)	
List<ACharacter>	getGhosts()	
void	removeFruit(Altem, boolean)	
int	getLives()	
int	getCurrentScore()	
void	setGameParameters(int, int, int)	
void	init()	
int	getHighestScore()	
void	removeBigDot(Altem, boolean)	

ACharacter		
IUpdatePacmanStrategy	updateStrategy	
Point	originalLoc	
Point	vel	
int	direction	
int	size	
int	getSize()	
Point	getOriginalLoc()	
void	setVel(Point)	
void	setSize(int)	
Point	getVel()	
void	setDirection(int)	
void	setUpdateStrategy(IUpdatePacmanStrategy)	
boolean	detectCollisionWithWalls(int, int[])	
void	propertyChange(PropertyChangeEvent)	
IUpdatePacmanStrategy	getUpdateStrategy()	
int	getDirection()	

Altem		
boolean	isEaten	
int	score	
void	propertyChange(PropertyChangeEvent)	
void	setEaten(boolean)	
boolean	isEaten()	
int	getScore()	

ICharacter Command

Update Command

- Called by the property change methods in pacman and ghosts to update their position

Switch Command

- Called by the eatBigDot method in game store and will switch the moving strategy of ghosts

ICharacterCmd

```
m execute(ACharacter, ACharacter) void
m execute(ACharacter) void
```

CmdFactory

```
f singleton CmdFactory
m makeCmdFactory() CmdFactory
m makeCmd(String) ICharacterCmd
```

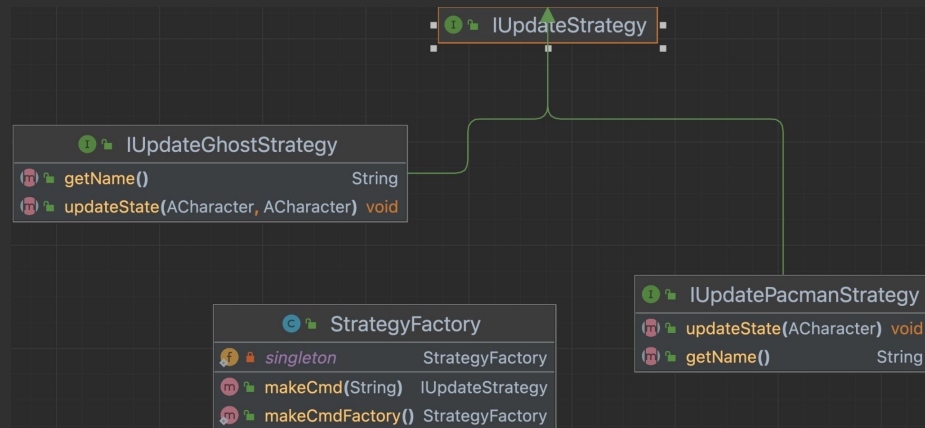
Update Strategy

Pac-man

- Move across the map given the arrow direction pressed by users
- Store the next direction so that it will turn once opportunity arises

Ghosts

- Two moving strategies
- Chase - aggressively chase pacman
- Random walk - move randomly across the map





Design Pattern

Command pattern

ICharacterCmd

Update State

Interact with items

Singleton pattern

StrategyFactory

CmdFactory

Strategy pattern

IUpdateStrategy

Pacman

Ghosts

Factory pattern

StrategyFactory

CmdFactory



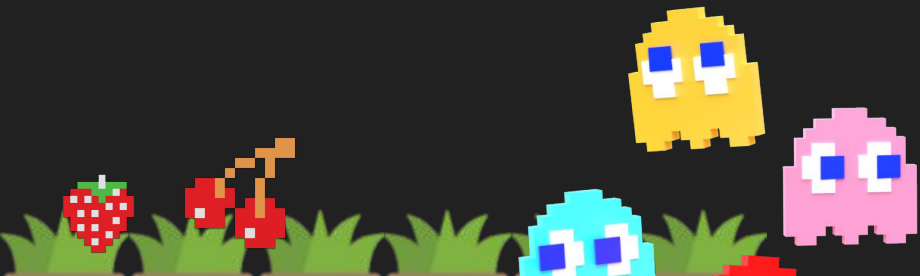
API endpoints

GET /initialize Initialize the pac-man world.

GET /clear Clear the pac-man world.

POST /update Update the pac-man world.

POST /setGameParameters Set the game parameters.





Frontend and backend challenges

Frontend



- Update user interface when an event occur
- Parse messages from backend

Backend



- Handle communication between frontend and backend
- Testing all the codes in model



GUI Demo



<https://pacman-api-team-bigfoot.herokuapp.com/>



