Fishing Valley Documentation

Created by

Phattarathida Punyarit 6330406621
Punnapob Jindakul 6330320221

2110215 Programming Methodology
Semester 1 Year 2021
Chulalongkorn University

Fishing Valley

Introduction

Fishing Valley is a fishing simulator game. The objective of this game is to make as much money as you can from fishing. You can also use the money to upgrade your item or some abilities.

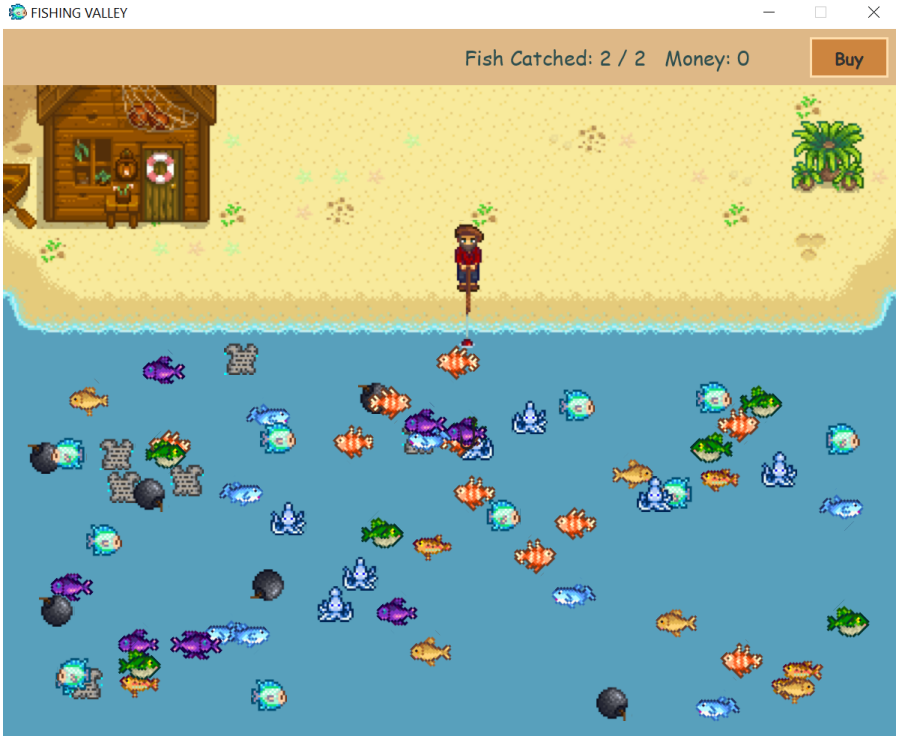
Rules

A player can start fishing by pressing the spacebar to set the hook and pressing S to reel the rod. When the fish is catched, press W to spin the rod back and then press E to sell it and get money. The fish can be catched more than one at a time. If a player catches a bomb, whatever are hooked will be gone. It is available for a player to turn right and left by pressing D and A respectively. Fish price, walking speed, hook speed and hook size can be upgraded by spending money in a shop.

Example

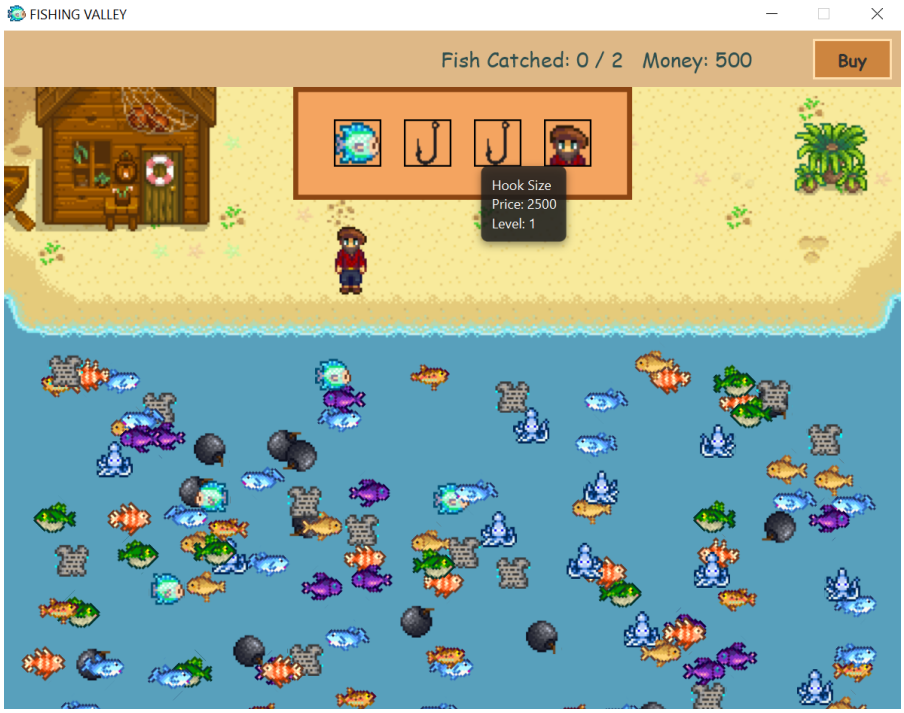
- The hook appears on the screen when pressing the spacebar.

- Press S to reel the rod and after the fish is hooked, the number of fish catched will be shown in the tab above the player.

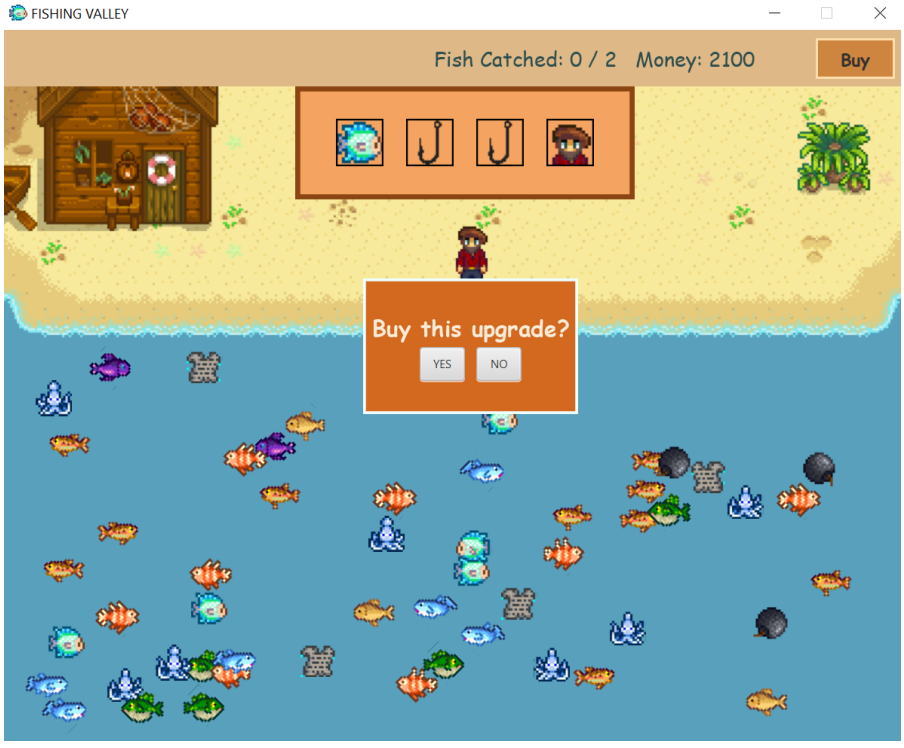


- Press W to spin the rod back and press E to sell the fish. After pressing E, the fish will disappear from the hook and the money from selling will be shown in the tab above the player. The player position can also be moved by pressing A (turn left) and pressing D (turn right).

- Click on the “Buy” button to see what are selling and see description when entering the mouse.



- After clicking on what to buy, there will be a pop up window asking for confirmation.

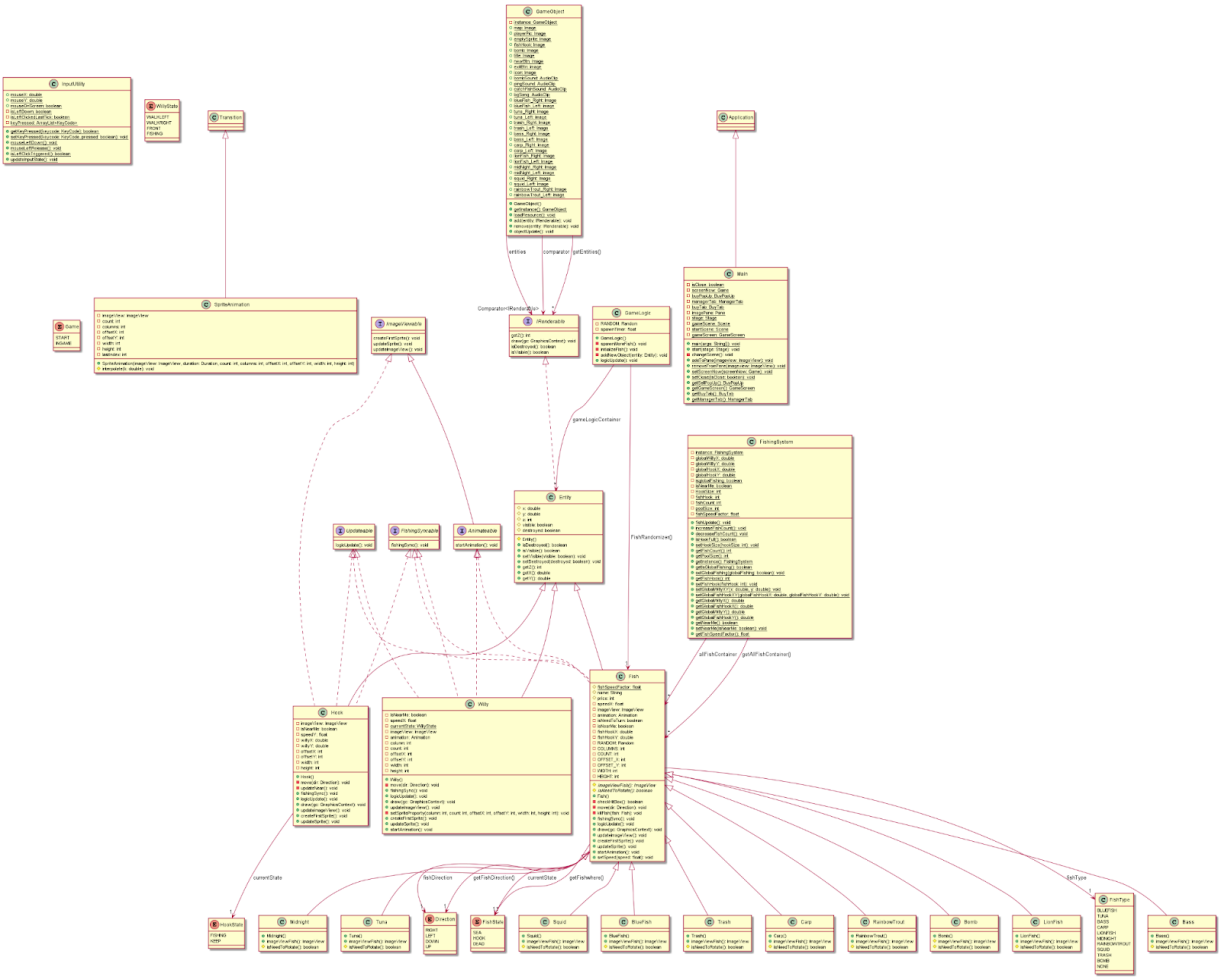


Main Menu Scene

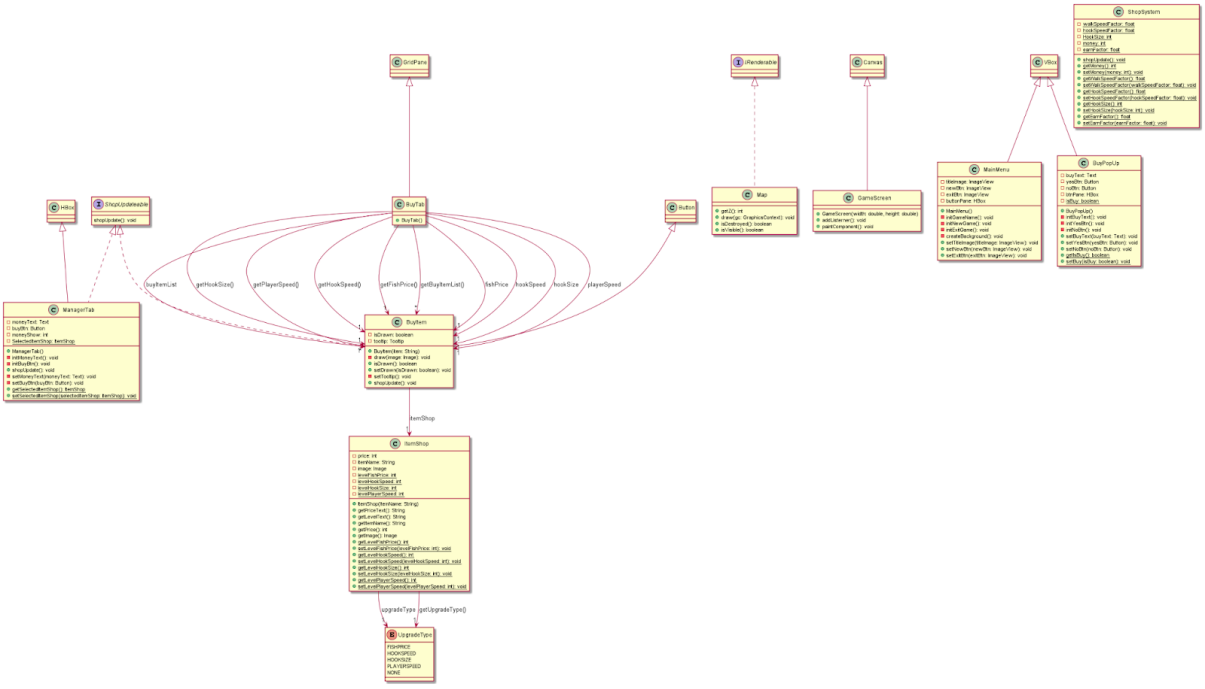
Click “New Game” to proceed to the game stage and click “Exit” to close the game.

Class Diagram

1. Package animation, fish, fishing, input, logic, main and player



2. Package shop and ui



1. Package animation

1.1 interface Animationable

Method

+ void startAnimation()	initialize animation

1.2 interface ImageViewable

Method

+ void createFirstSprite()	Initialize sprites
+ void updateSprite()	Update sprites for new animation
+ void updateImageView()	Update imageView of sprites for adding to pane in GameScreen

1.3 class SpriteAnimation extends Transition

Field

- final ImageView imageView	ImageView of sprites
- final int count	A number of frame that you want to be animated
- final int columns	A number of frame in a column (line of sprites)
- final int offsetX	Offset in X-axis from Origin
- final int offsetY	Offset in Y-axis from Origin
- final int width	Width of frame in animation

- final int height	Height of frame in animation
- int lastIndex	Stopping index used for loop in Method Interpolate

Constructor

+ SpriteAnimation(ImageView imageView, Duration duration, int count, int columns, int offsetX, int offsetY, int width, int height)	Initialize Fields

Method

# void interpolate(double k)	Looping show a frame in animation

2. Package fish

2.1 enum FishType

BLUEFISH, TUNA, BASS, CARP, LIONFISH, MIDNIGHT, RAINBOWTROUT, SQUID, TRASH, BOMB, NONE
Type of fish (NONE is Type of a Bomb that was exploded)

2.2 enum FishState

SEA, HOOK, DEAD
State of fish

2.3 class Fish Extends Entity implements Updateable, Animateable, FishingSyncable

Field

# FishType fishType	fishType of fish
# String name	Name of fish
# int price	Price of fish
# float fishSpeedFactor	speedFactor of all fish
- float speedX	Speed of fish in X-Axis
- boolean isNeedToTurn	Return True if Fish need to turn in opposite site
- boolean isNearMe	Return True if fishHook is near to Willy
- Direction fishDirection	Current direction of fish
- FishState currentState	Current state of fish
- double fishHookX	X-position of fish
- double fishHookY	Y-position of fish
- ImageView imageView	imageView of fish
- Animation animation	Animation of fish
- final Random RANDOM	Use for random stuffs inside this class
- final int COLUMNS	ImageView's number of frame that you want to be animated
- final int COUNT	ImageView's number of frame in a column (line of sprites)
- final int OFFSET_X	ImageView's offset in X-axis from Origin

- final int OFFSET_Y	ImageView's offset in Y-axis from Origin
- final int WIDTH	ImageView's width of frame in animation
- final int HEIGHT	ImageView's height of frame in animation

Constructor

+ Fish()	Initialize Fields and also start create First Sprites

Method

- boolean checkHitBox()	Return True if fish position is in fish hook HitBox
- void move(Direction dir)	Handle movement of fish to not touching bound
- void killFish(Fish fish)	Kill fish out of game
+ void fishingSync()	Synchronize Logic with all fish stuff
+ void logicUpdate()	Update all logic about fish - If Fish gets into FishHook's HitBox, count it as Hooked and stick with Hook until Willy Keeps his FishHook. - If Bomb gets into FishHook's HitBox, All fish in Hook is Bombed and count as Dead. - If Hook is near Willy, He can keep his Hook and sell all fishes in the hook automatically. - If nothing above happen, Fish continue swim in its way

+ void draw(GraphicsContext gc)	Draw an image of fish (But we use ImageView instead)
+ void updateImageView()	Remove old imageView and add new update imageView of fish to imagePane
+ void createFirstSprite()	Initialize imageView of fish
+ void updateSprite()	Update new imageView for Animation (rotate it if it's needed)
+ void startAnimation()	Update new Animation
Generate Getter for some fields	

2.4 class Bass extend Fish

Constructor

+ Bass()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return True If fish is needed to rotate to show correctly

2.5 class BlueFish extend Fish

Constructor

+ Bass()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return True If fish is needed to rotate to show correctly

2.6 class Bomb extend Fish

Constructor

+ Bomb()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return True If fish is needed to rotate to show correctly

2.7 class Carp extend Fish

Constructor

+ Carp()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return True If fish is needed to rotate to show correctly

2.8 class LionFish extend Fish

Constructor

+ LionFish()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return true If fish is needed to rotate to show correctly

2.9 class Midnight extend Fish

Constructor

+ Midnight()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return true If fish is needed to rotate to show correctly

2.10 class RainbowTrout extend Fish

Constructor

+ RainbowTrout()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return true If fish is needed to rotate to show correctly

2.11 class Squid extend Fish

Constructor

+ Squid()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return true If fish is needed to rotate to show correctly

2.12 class Trash extend Fish

Constructor

+ Trash()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return true If fish is needed to rotate to show correctly

2.13 class Tuna extend Fish

Constructor

+ Tuna()	Initialize remaining Fields

Method

+ ImageView imageViewFish()	Return imageView of fish depend of fish direction of fish
# boolean isNeedToRotate()	Return true If fish is needed to rotate to show correctly

3. Package input

3.1 InputUtility

Field

+ double mouseX	X-Position of mouse
+ double mouseY	Y-Position of mouse
+ boolean mouseOnScreen	Return true if mouse is on screen
- boolean isLeftDown	Return true If mouse left down
- boolean isLeftClickedLastTick	Return true If mouse left down in last tick
- ArrayList<KeyCode> keyPressed	Store a keyPress input

Method

+ boolean getKeyPressed(KeyCode keycode)	Return true If keypress contains keycode input
+ void setKeyPressed(KeyCode keycode,boolean pressed)	Set KeyCode to keypress
+ void mouseLeftDown()	Set IsLeftDown and isisLeftClickedLastTick to false
+ void mouseLeftRelease()	Set IsLeftDown to false
+ void updateInputState()	Update isLeftClickedLastTick to False
+ boolean isLeftClickTriggered()	Return isLeftClickedLastTick

4. Package fishing

4.1 interface FishingSyncable

Method

+ void fishingSync()	Sync fishing stuff with global

4.2 class FishingSystem

Field

- final FishingSystem instance	Instance of FishingSystem
- double globalWillyX	Global field of Willy's X position
- double globalWillyY	Global field of Willy's Y position
- double globalHookX	Global field of Hook's X position
- double globalHookY	Global field of Hook's Y position
- boolean isglobalFishing	Global field of isFishing
- boolean isNearMe	Global field of isNearMe
- final ArrayList<Fish> allFishContainer	Store all fish
- int HookSize	Count of fish that hook can hold
- int fishHook	Count of fish in Hook
- int fishCount	Count of fish in Game
- int poolSize	Max count of fish in Game
- float fishSpeedFactor	Speed factor of fish

Method

+ void fishUpdate()	Update fishing Stuff - Update HookSize - Update fish in Hook - Eliminate fish that is dead
+ increaseFishCount()	Increase fish in game

+ decreaseFishCount()	Decrease fish in game
+ boolean isHookFull()	Return true if Hook is Full
Generate Getter/Setter for some fields	

4.3 enum HookState

FISHING , KEEP
State of Hook

4.4 class Hook extend Entity implements Updateable, ImageView ,FishingSyncable

Field

- double willyX	X Position of Willy
- double willyY	Y Position of Willy
- boolean isNearMe;	Return True If Hook is near Willy
- float speedY;	Speed of Hook in Y-Axis
- ImageView imageView;	imageView of Hook
- HookState currentState;	Current state of Hook
- final int OFFSET_X	ImageView's offset in X-axis from Origin
- final int OFFSET_Y	ImageView'soffset in Y-axis from Origin
- final int WIDTH	ImageView's width of frame in animation
- final int HEIGHT	ImageView's height of frame in

	animation

Constructor

+ Hook()	Initialize Fields and also start create First Sprites

Method

- void move(Direction dir)	Handle Hook movement
- void updateNear()	Update isNearMe
+ void fishingSync()	Synchronize field with Global
+ void logicUpdate()	Update all logic about Hook - Show Hook when Go fishing - Hide Hook when Keep Hook - Press S to lower Hook - Press W to higher Hook
+ void draw(GraphicsContext gc)	Draw an image of Hook (But We use ImageView instead)
+ void updateImageView()	Remove old imageView and add new update imageView of hook to imagePane
+ void createFirstSprite()	Initialize imageView of hook
+ void updateSprite()	Update new imageView

5. Package player

5.1 enum WillyState

WALKLEFT ,WALKRIGHT, FRONT, FISHING

State of Willy

5.2 class Willy extends Entity implements Updateable, Animateable, FishingSyncable

Field

- boolean isNearMe	Return true if hook is near to Wily
- float speedX	Speed of Willy in X-Axis
- WillyState currentState	Current Willy State
- ImageView imageView	imageView of Willy
- Animation animation	Animation of Willy
- int COLUMNS	ImageView's number of frame that you want to be animated
- int COUNT	ImageView's number of frame in a column (line of sprites)
- int OFFSET_X	ImageView's offset in X-axis from Origin
- int OFFSET_Y	ImageView's offset in Y-axis from Origin
- int WIDTH	ImageView's width of frame in animation
- int HEIGHT	ImageView's height of frame in animation

Constructor

+ Willy()	Initialize Fields and also start

	creating first sprites

Method

- void move(Direction dir)	Handle movement of Willy
- void fishingSync()	Synchronize Logic with all fish stuff
+ void logicUpdate()	Update all logic about Willy - Press Space to Go fishing - Press E when hook is near Willy to keep hook - Press D to move right - Press A to move left - When Not fishing and doesn't press D or A Willy will be head to sea
- void setSpriteProporty(int column, int count, int offsetX, int offsetY, int width, int height)	Setter of ImageView' Field (Column,Count,offsetX,offsetY,width, height)
+ void draw(GraphicsContext gc)	Draw an image of Willy (But We use ImageView instead)
+ void updateImageView()	Remove old imageView and add new update imageView of Willy to imagePane
+ void createFirstSprite()	Initialize imageView of Willy
+ void updateSprite()	Update new imageView for Animation
+ void startAnimation()	Update new Animation

6. Package logic

6.1 enum Direction

RIGHT, LEFT, DOWN, UP
Direction of all entities (Except Map)

6.2 interface Updateable

+ void logicUpdate()	Update logic

6.3 interface IRenderable

+ int getZ()	Return Z of entities
+ draw(GraphicsContext gc)	Draw by graphic context (Some class use imageView)
+ boolean isDestroyed()	Return true if it's destroyed
+ boolean isVisible()	Return true if it's able to see

6.4 class Entity implements IRenderable

Field

# protected double x	X Position of entity
# protected double y	Y Postion of entity
# protected int z	Z Position of entity (Z is Layer Position : more Z means more the layer go upper)
# protected boolean visible	Return true if it's able to see
# protected boolean destroyed	Return true if it's destroyed

Constructor

# Entity()	Initialize fields

Method

Generate Getter/Setter for some fields	

6.5 class GameObject

- final GameObject instance	Instance of GameObject
- ArrayList<IRenderable> entities	ArrayList stored entity
- Comparator<IRenderable> comparator	Comparator among entity
- Image blueFish_Right, blueFish_Left, tuna_Right, tuna_Left, trash_Right, trash_Left, bass_Right, bass_Left, carp_Right, carp_Left, lionFish_Right, lionFish_Left, midNight_Right, midNight_Left, squid_Right, squid_Left, rainbowTrout_Right, rainbowTrout_Left	All fish image
- AudioClip bombSound, pingSound, catchFishSound,bgSong	All sound
- Image map, playerPic, emptySprite, fishHook,bomb,title,newBtn,exitBtn,icon	Remaining images that are not fish

Constructor

+ GameObject()	Initialize entities and comparator

Method

+ GameObject getInstance()	Get instance of Gameobject
+ void loadResource()	Load all image and sound in game
+ void add(IRenderable entity)	Add new entity into entities

	and sort
+ void remove(IRenderable entity)	Remove given entity out of entities
+ void objectUpdate()	Update and remove destroyed entity
+ List<IRenderable> getEntities()	Getter of entities

6.6 class GameLogic

Field

- final List<Entity> gameLogicContainer	ArrayList stored entity
- final Random RANDOM	Use to random stuff inside this class
- float spawnTimer	Timer count up

Constructor

+ GameLogic()	Setup Map Player Hook and spawn some of fishes

Method

- void spawnMoreFish()	Random spawn 1 fish and add to gameLogicContainer
- void initializeFish()	Random spawn 10-19 fishes
- Fish FishRandomizer()	Return Fish Randomly
- void addNewObject(Entity entity)	Add new entity into 3 Container - gameLogicContainer - entities inside Class GameObject - allFishContainer inside Class FishingSystem (If that entity is fish)

+ void logicUpdate()	- Update logic of all entity - Eliminate if it's destroyed - If the count of fish in game is less than PoolSize, it'll spawn more.

7. Package UI

7.1 class MainMenu extends VBox

Field

- ImageView titleImage	Image of game title
- ImageView newBtn	Image of new game button
- ImageView exitBtn	Image of exit game button
- HBox buttonPane	Box for the setting newBtn and exitBtn alignment

Constructor

+ MainMenu()	Set up main menu page

Method

- void initGameName()	Initialize game title
- void initNewGame()	- Initialize new game button - Set MouseEvent on new game button
- void initExitGame()	- Initialize exit game button - Set MouseEvent on exit game button
- void createBackground()	Create background for main menu page

+ void setTitleImage(ImageView titleImage)	Set image for game title
+ void setNewBtn(ImageView newBtn)	Set image for new game button
+ void setExitBtn(ImageView exitBtn)	Set image for exit game button
Generate Setter for each ImageView fields	

7.2 Class GameScreen extends Canvas

Constructor

+ GameScreen(double width, double height)	Set up Game Screen and add its event listener

Method

+ void addListerner()	Receive Input through Game screen and set their input to class InputUtility
+ void paintComponent()	Draw all entity that use graphic context (some stuff use imageview but it need to call pass this function)

7.3 Class Map implements IRendable

Method

+ int getZ()	Return -99
+ void draw(GraphicsContext gc)	Draw map on game screen

+ boolean isDestroyed()	Return false
+ boolean isVisible()	Return true

7.4 class class ManagerTab extends HBox implements ShopUpdateable

Field

- Text moneyText	Text displaying money
- Button buyBtn	Button for open BuyTab
- int moneyShow	Money for show
- ItemShop SelectedItemShop	Currently selected upgrade

Constructor

+ moneyShow	Setup ManagerTab

Method

- void initMoneyText()	Set up MoneyText
- void initBuyBtn()	Set up BuyButton
+ void shopUpdate()	Update Money and calculate money and level after buying. After Buying, reset selected upgrade to be None
Generate Getter/Setter for remaining fields	

7.5 class BuyTab extends GridPane

Field

- ObservableList<BuyItem>	Collect all upgrade to buy in the pane

buyItemList	
- BuyItem fishPrice	Button for selecting fish price upgrade
- BuyItem hookSpeed	Button for selecting hookSpeed upgrade
- BuyItem hookSize	Button for selecting hookSize upgrade
- BuyItem playerSpeed	Button for selecting playerSpeed upgrade

Constructor

+ BuyTab()	Set up the BuyTab pane

Method

Generate Getter for all fields	

7.6 class BuyItem extends Button implements ShopUpdateable

Field

- ItemShop itemShop	itemShop for making buttons
- ToolTip tooltip	Tooltip when mouse enters
- boolean isDrawn	Return true if the image is drawn

Constructor

+ BuyItem()	Set up the itemShop into buttons and set MouseEvent

Method

- void draw(Image image)	Draw image for BuyItem
- void setTooltip()	Setup tooltip and its response
- void shopUpdate()	update Level on Tooltip
Generate Getter/Setter for isDrawn	

7.7 class BuyPopUp extends VBox

Field

- Text buyText;	Text displaying "Buy this upgrade?"
- Button yesBtn	Yes Button
- Button noBtn	No Button
- HBox btnPane	Pane for putting Button
- boolean isBuy	Return true if buying upgrade successful

Constructor

+ BuyPopUp()	Set up Buy PopUp

Method

- void initBuyText()	Set up BuyText
- void initYesBtn()	Set up YesButton
- void initNoBtn()	Set up NoButton
- boolean getIsBuy()	Getter of isBuy
Generate Setter for yesBtn, noBtn and buyText	

7.8 class ItemShop

Field

- int price	Price of upgrade
- String itemName	Name of upgrade
- Image image	Image of upgrade
- UpgradeType upgradeType	Type of upgrade
- int levelFishPrice	Current level of FishPrice
- int levelHookSpeed	Current level of HookSpeed
- int levelHookSize	Current level of HookSize
- int levelPlayerSpeed	Current level of PlayerSpeed

Constructor

+ itemShop(String itemName)	Initialize type of upgrade following itemName

Method

+ String getPriceText()	Get Price for displaying in ToolTip
+ String getLevelText()	Get Level for displaying in ToolTip
Generate Getter/Setter for all remaining fields	

8. Package shop

8.1 enum UpgradeType

FISHPRICE, HOOKSPEED, HOOKSIZE, PLAYERSPEED, NONE
Type of Upgrade (None represents unselected state)

8.2 interface ShopUpdateable

Method

+ void shopUpdate()	Update component's details about shop

8.3 interface ShopSystem

Field

- float walkSpeedFactor	Willy walk speed factor
- float hookSpeedFactor	Hook speed factor
- int HookSize	Hook Size
- int money	Money in pocket use for buying upgrades
- float earnFactor	Fish price factor

Method

+ void shopUpdate()	Update all implemented Shopupdateable
Generate Getter/Setter for all fields	

9. Package main

9.1 enum game

START, INGAME
State of game displaying

9.2 class Main extends Application

Field

- boolean isClose	Return true If close game

- Game screenNow	State of screen displaying
- BuyPopUp buyPopUp	Confirm BuyPopUp
- ManagerTab managerTab	ManagerTab
- BuyTab buyTab	BuyTab
- Pane imagePane	Pane for placing imageView
- stage stage	Stage
- Scene gameScene	In game scene
- Scene startScene	Main Menu scene
- GameScreen gameScreen	Game screen

Method

+ void main(String[] args)	Launch javaFX application
+ void start(Stage stage)	- Set up all game component and place on right scene - Looping Update all stuff
- void changeScene()	Change scene following state of game state
+ void addToPane(ImageView imageview)	Add imageView to ImagePane
+ void removeFromPane(ImageView imageview)	Remove imageView out of ImagePane
Generate Getter/Setter for some fields	