Group aaa Report

on

Ludiquarium

A part of 2110215 Programming Methodology studying

Presented by

Phanukorn Sunthornwetchapong 5931050721

Patipon Kongsomboonsak 5930299521

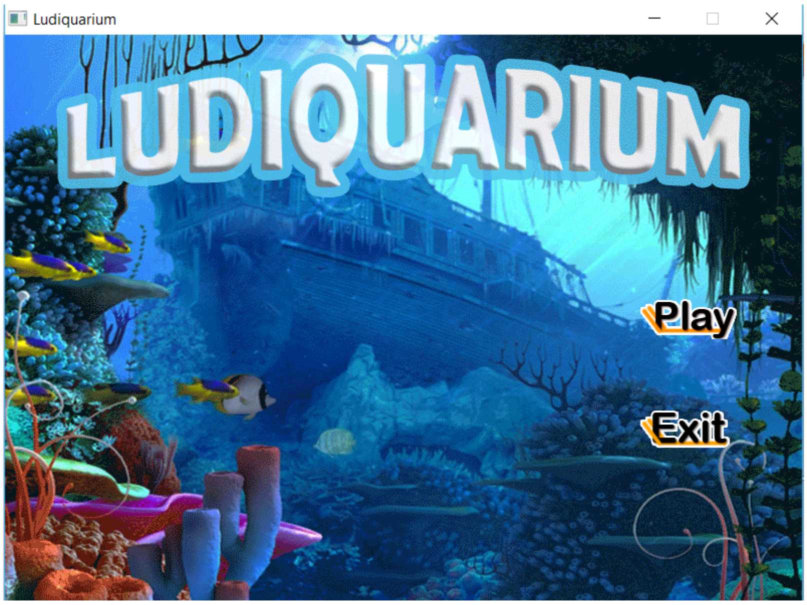
Introduction

Ludiquarium (ludicrous + aquarium), ACTION game derived from Insaniquarium. The game has been changed in aspect of fish roles and some appearance of fish. (To be sincere, all of fish are changed to someone we already known)

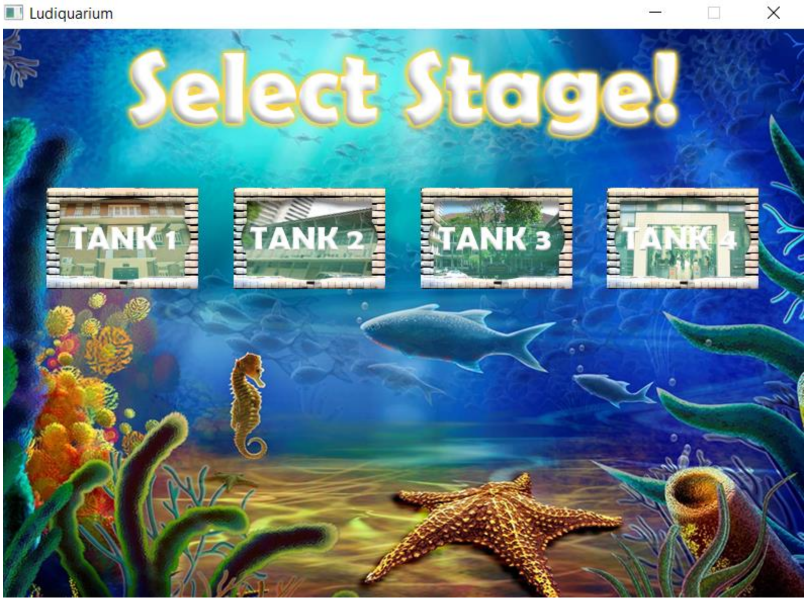
This game consists of 4 tanks like Insaniquarium, contains Fish and Aliens like original, but the goal is to take care of Special Fish until it fully grown up and not eaten by Aliens

We hope you could enjoy mouse spamming like the old days. (And some frame drops like the old fashion game)

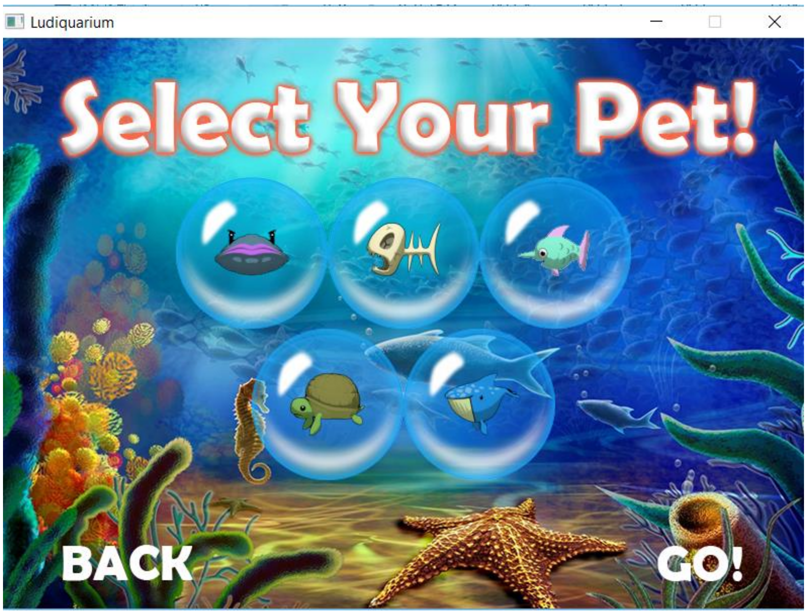
How to Play



This is our first page in the game, click on Play (anywhere which is not transparent) will take you to tank select screen. Or if you didn't want to play anymore, click on Exit below Play button (This will prompt you for confirmation) or X at the top of screen (This will close program instantly)



This is tank selection screen, click one time for highlight the tank and click again to go to that scene. Or if you are not sure about your decision, you can click the other tank to change your mind. (After main menu, closing game can be done by click X on the Title Bar)



This is pet selection screen. All pets will be unlocked at the first time you play.

Pet Description (From Left to Right, Top to Bottom)

Niko: Craft Pearl, which value for 350 coins. But it crafts slowly.

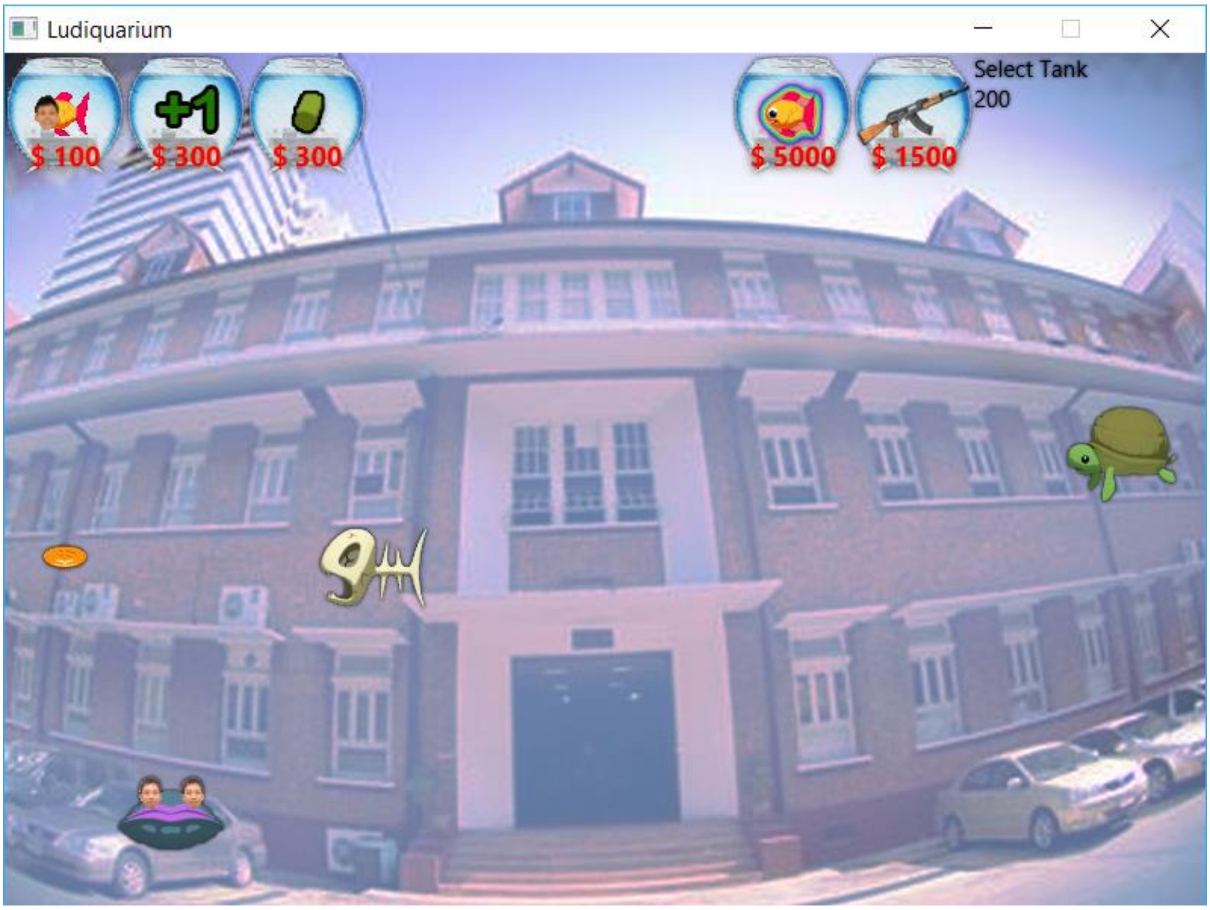
Vert: Act Like Lv.2 Guppy, craft golden coin.

Itchy: Help you in fighting with aliens.

Seymour: Make coin drop slower

Wadsworth: Keep Guppy in safe place when aliens came (But can make you lose the game for don't have any fish in the tank)

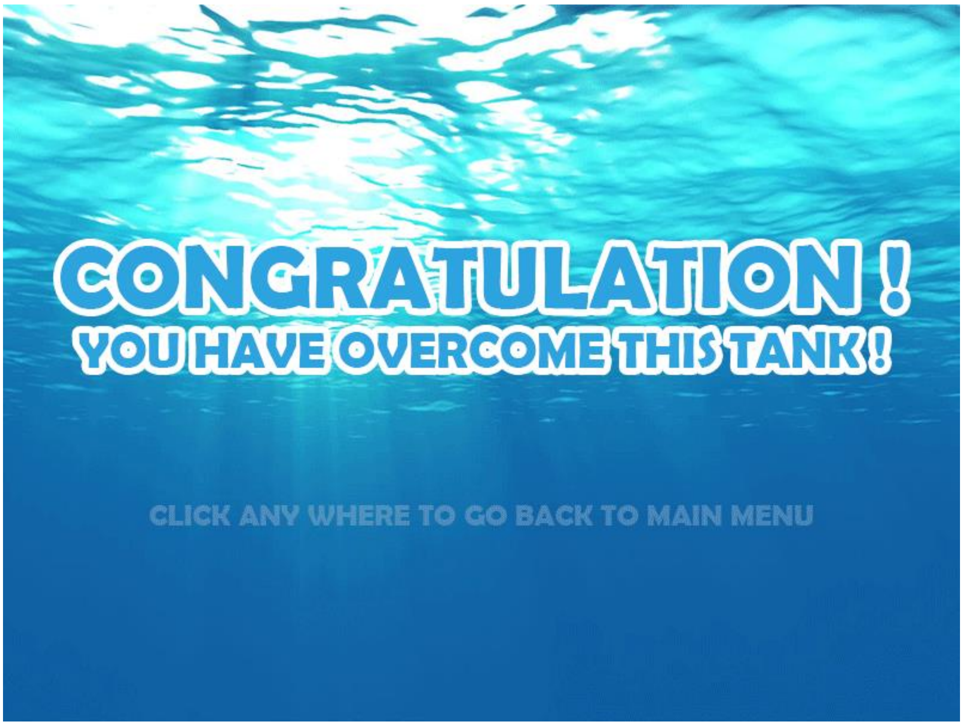
You can select at most 3 pets (which means you can go by yourself without any companion) by clicking on desired pet and click again for deselecting it. After finished selecting, click GO! To play the game or click BACK if you think the tank (level) you selected before is not what you want.



This is our tank screen. Top sections which contain bowls are item buying section, where you can buy the fish to your tank or upgrades to make your game much easier. Items are derived in many tanks. However, there are 4 items which sell in every tank.

- Max Food Upgrade (Item with +1 sign, $300): upgrade number of maximum food which can be in the same tank (Maximum: 9)
- Food Quality Upgrade (Right-hand side of Max Food Upgrade, $300): Upgrade food quality, which make fish grow faster
- Special Fish (Guppy with rainbow glowing, varied by tank level): Ultimate fish that you must raise it until fully grown. However, it doesn't give you anything like other fish.
- Gun Upgrade (Item with gun icon, $1500): Upgrade attack damage when you clicked on some alien (However, some alien doesn't affect by this upgrade)

About the other fishes, you should discover some of the tips by yourself. But I could `tell you that in this game, fish can strike back too! However, fish are ludicrous enough to make a fuss.

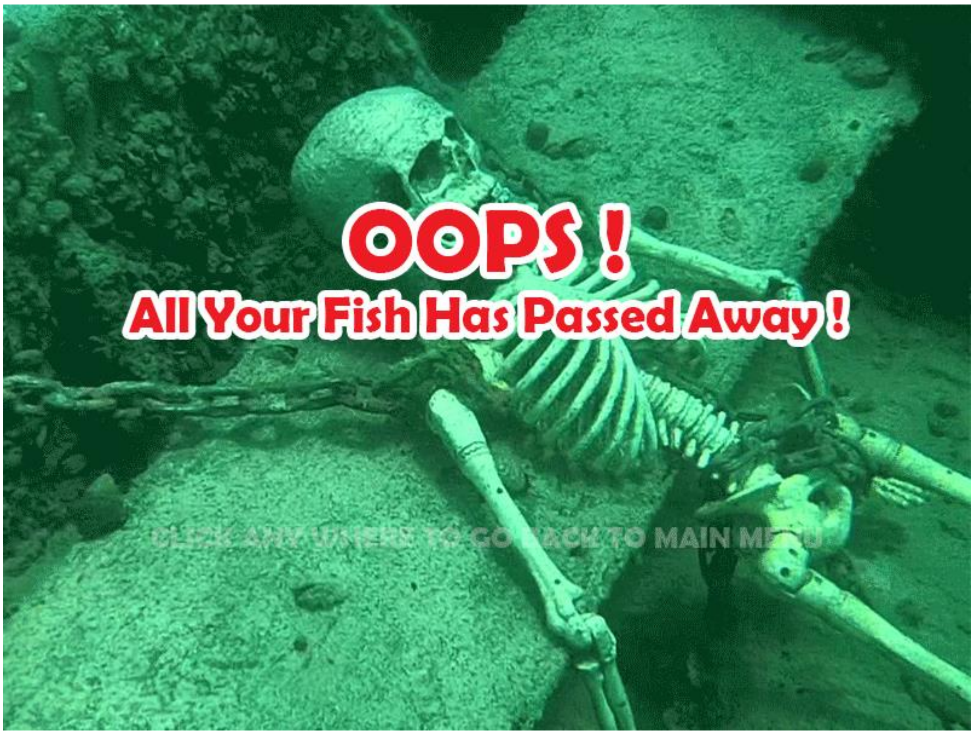


CONGRATULATION !

YOU HAVE OVERCOME THIS TANK !

CLICK ANY WHERE TO GO BACK TO MAIN MENU

After raised special fish, you should see this screen. Or if you lose the game (don't have any fish in the tank at some point of time), you will see something like below screen capture.



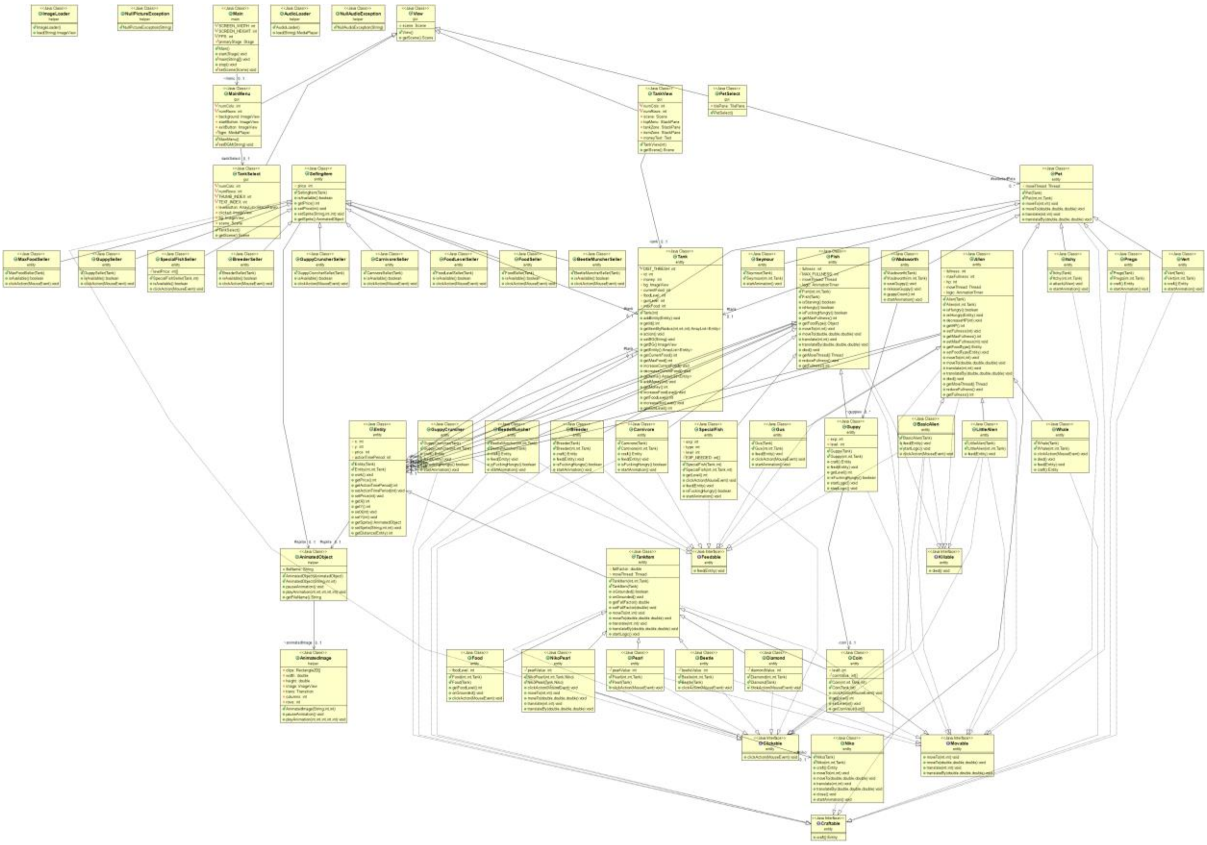
OOPS !

All Your Fish Has Passed Away !

CLICK ANY WHERE TO GO BACK TO MAIN MENU

Good luck, have fun on your journey!

UML Diagram



Implementation Detail

Package helper

Class AnimatedImage extends ImageView

Field

-Rectangle2D[] clips	Storing clips position from large sprite sheet
-double width	Sprite sheet width
-double height	Sprite sheed height
-ImageView image	ImageView storing sprite shee
-Transition trans	Transition for sprite
-int columns	Number of columns in sprite sheet
-int rows	Number of rows in sprite sheet

Constructor

+AnimatedImage(String fileName, int columns, int rows)	Set viewport width&height and construct clips

Method

+void pauseAnimation()	Stop animation (transition thread)
+void playAnimation(int cycleCount, int fromRow, int formCol, int toRow, int toCol)	Construct transition thread with 20FPS refesh rate and play from (fromRow, fromCol) to (toRow, toCol)

Class AnimatedObject extends Pane

Field

-AnimatedImage animatedImage	AnimatedImage to be shown on the pane
-String fileName	Sprite sheet fileName

Constructor

+AnimatedImage(String fileName, int columns, int rows)	Initialize animatedImage with sprite named fileName and add to the pane (itself)

Method

+void pauseAnimation()	Stop animation (transition thread)
+void playAnimation(int cycleCount, int fromRow, int formCol, int toRow, int toCol)	Call animatedImage with function playAnimation with same parameter
+String getFileName()	Getter for fileName

Class AudioLoader

Method

+MediaPlayer load(String fileName)	Load MediaPlayer from SpriteLoader and throw NullAudioException if file is not found

Class ImageLoader

Method

+ImageView load(String fileName)	Load ImageView from SpriteLoader and throw NullPictureException if file is not found

Class NullAudioException extends Exception

Constructor

+NullAudioException (String string)	Set message to "<string> is not found in audio."

Class NullPictureException extends Exception

Constructor

+NullImageException (String string)	Set message to "<string> is not found in images."

Class SpriteLoader

Field

-static Map<String, ImageView> iv	HashMap for ImageView, map with fileName
-static Map<String, MediaPlayer> mp	HashMap for MediaPlayer, map with fileName

Method

+ImageView getImage(String fileName)	Try to find fileName in iv (find Image), add file to HashMap if found. Otherwise, print stack trace and return null to throw NullPictureException in ImageLoader
+MediaPlayer getMedia(String fileName)	Try to find fileName in mp (find Media), add file to HashMap if found. Otherwise, print stack trace and return null to throw NullAudioException in AudioLoader

Package entity

Interface Feedable

Method

+void feed(Entity food)	Check whether Entity has same type as foodType

Interface Craftable

Method

+Entity craft()	Return what is crafted as Entity
+void resetCoin()	Reset Crafted Entity

Interface Clickable

Method

+void clickAction(MouseEvent event)	Set click action

Interface Movable

Method

+void moveTo(int x, int y)	Change (x, y) to given position
+void moveTo(double x, double y, int time)	Move with constant velocity within time given and assgin movement to moveThread
+void translate(int x, int y)	Change (x, y) by delta(x, y) given
+void translateBy (double x, double y, double time)	Change (x, y) by delta(x, y) given withint time given and assgin movement to moveThread

Class Tank

Field

+static final int DIST_THRESH	Fish's Threshold distance to determined that should it action with something
#int id	Tank id (level)
#ArrayList<Entity> entities	Entities in the tank (Food, Fish, Pet and Alien)
#ArrayList<Entity> items	List of item sellers (Collectable Items)
#int money	Money for buying items.
#ImageView bg	Background for current tank
#int currentFood	Current food in the tank

#int foodLevel	Current food quality (max. 2)
#int gunLevel	Current gun level to attack alien (Max. 9)
#int maxFood	Current maximum food allow in tank (Max. 9)

Constructor

+Tank (ArrayList<Item> items)	- Start a tank with two Guppies - Start with 200 gold - Set background/Food level, Current food, Max food

Method

+Entity addEntity(Entity entity)	Add Entity to the tank, return added item
+void addMoney(int amount)	Add money by amount (negative value for buying items)
+int getId()	Getter of id
+ArrayList<Entity> getEntityByRadius(int x, int y, int r)	Get Item within radius r from point (x, y) from entity and return
+int getGunLevel()	Getter of gunLevel
+void setBG(String fileName)	Load background by fileName from ImageLoader
+ImageView getBG()	Getter of bg
+ArrayList<Entity> getEntity()	Getter of entities
+int getCurrentFood()	Getter of currentFood
+int getMaxFood()	Getter of maxFood
+void increaseCurrentFood()	Increase currentFood by 1
+void decreaseCurrentFood()	Decrease currentFood by 1 and set to zero if less than zero
+ArrayList<Entity> getItem()	Getter of items
+void addMoney(int amount)	Add money (decrease for negative value) and set to zero if money is negative
+void increaseFoodLevel()	Increase foodLevel by 1 and regulate the value to not exceed 2
+int getFoodLevel()	Getter of foodLevel
+void increaseGunLevel()	Increase gunLevel by 1 and regulate the value to not exceed 9
+void increaseMaxFood()	Increase maxFood by 1 and regulate the value to not exceed 9

Abstract Class Entity

Field

#int x	x-axis Position in tank
#int y	y-axis Position in tank
#Tank tank	Tank reference
#AnimatedObject sprite	Sprite of each entities
#AnimationTimer logic	Logic controller (movement and actions)
#Entity self	Self reference (for referencing in AnimationTimer)

Constructor

+Entity (Tank tank)	Construct an Entity, random value of this.x and this.y in range of tank.getbg().getWidth(), tank.getbg().getHeight() set other fields to 0 and set self reference
+Entity (int x, int y, Tank tank)	Construct an Entity like Entity(Tank tank)with specified this.x and this.y

Method

+int getX()	Getter of x position
+int getY()	Getter of y position
+int setX(int x)	Setter of x position, which check position according to GUI
+int setY(int y)	Setter of y position, which check position according to GUI
+AnimatedObject getSprite()	Getter of sprite
+void setSprite(String fileName, int col, int row)	Setter of sprite, which set AnimatedObject to sprite
+int getDistance(Entity other)	Measure Euclidean distance between self and other entity, return in square of distance
+void startLogic()	Instantiate AnimationTimer and start it
+void stopLogic()	Stop AnimationTimer if it’s running
+void moveTo(int x, int y)	@Override Change (x, y) to given position
+void moveTo(double x, double y, int time)	@Override Move with constant velocity within time given and assgin movement to moveThread
+void translate(int x, int y)	@Override Change (x, y) by delta(x, y) given
+void translateBy (double x, double y, double time)	@Override Change (x, y) by delta(x, y) given withint time given and assgin movement to moveThread

Package entity.fish

Abstract Class Fish extends Entity implements

Field

#int fullness	Fish's fullness
#final int MAX_FULLNESS	Fish's maximum fullness value
#Thread moveThread	Fish's moving thread for reference
#Entity coin	Current crafted item reference

Constructor

+Fish (Tank tank)	Construct a Fish and random value of this.x and this.y, also set default fullness to 70
+Fish (int x, int y, Tank tank)	Construct a Fish with specified this.x and this.y, also set default fullness to 70

Method

+boolean isStarving()	Return true if fullness is equal or less than 0 (Starving to dead)
+boolean isHungry()	Return true if fullness is equal or less than 50% (Hungry enough to have a food)
+boolean isFuckingHungry()	Return true if fullness Is equal or less than 25% (So hungry that I'm yellow right now)
+int getMaxFullness	Getter of MAX_FULLNESS
+void reduceFullness()	Reduce fullness by 3
+int getfullNess()	Getter of fullness
+void setY(int Y)	@Override avoid fish being to low (and make item disappear as soon as it crafted)
+Methods from Movable (as described above)	
+Thread getMoveThread()	Getter of moveThread
+void died()	@Override stopLogic() and remove itself from tank
+void resetCoin()	@Override Reset crafted item to null

Class Guppy extends Fish implements Feedable, Killable, Craftable, Movable

Field

-int exp	Guppy's Exp.
-int level	Guppy's level

Constructor

+Guppy (Tank tank)	Construct a Guppy with super() and set exp = 90; (For presenting purpose) level to 0 setup animation and start logic
+Guppy (int x, int y, Tank tank)	Like Guppy(Tank tank) but also setX(x) and setY(y)

Method

+Entity craft()	If Guppy has no coin craft by itself in the tank, drop silver coin if level is 2, and drop gold coin if its level is 3.
+void feed(Entity food)	Feed guppy with Food, not Fish, increase fullness and exp according to foodLevel, decrease currentFood and remove it from tank and restore sprite if already hungry
+int getLevel()	Return Guppy's level calculated from exp
+boolean isFuckingHungry()	@Override Like super.isFuckingHungry() but also set sprite according hungry status
+void startLogic()	@Override Implement Guppy's Logic (Moving toward food, Idle movement, crafting coin and alien attack, which decrease HP only 1 point)

Class Carnivore extends Fish implements Feedable, Killable, Craftable, Movable

Constructor

+Carnivore (Tank tank)	Construct a Carnivore with super() and setup animation and start logic
+Carnivore (int x, int y, Tank tank)	Like Carnivore(Tank tank) but also setX(x) and setY(y)

Method

+Entity craft()	Like Guppy's craft method, but craft Diamond instead of coin and craft every 5 seconds (from last crafted time, not last collected time)
+void feed(Entity food)	Feed Carnivore with level 0 Guppy, increase fullness to the max, kill that Guppy and restore sprite if already hungry
+boolean isFuckingHungry()	@Override Like super.isFuckingHungry() but also set sprite according hungry status
+void startLogic()	@Override Implement Carnivore's Logic (Moving toward food, Idle movement, crafting coin and alien attack, which decrease alien's HP by 75)

Class GuppyCruncher extends Fish implements Feedable, Killable, Craftable, Movable

Field

-boolean isAlreadyHungry	Check whether GuppyCruncher changed sprite according to it hungriness

Constructor

+GuppyCruncher (Tank tank)	Construct a GuppyCruncher with super() and setup animation and start logic
+GuppyCruncher (int x, int y, Tank tank)	Like GuppyCruncher(Tank tank) but also setX(x) and setY(y)

Method

+Entity craft()	Like Guppy's craft method, but craft Beetle for BeetleMuncher instead of coin (which also collectable) and craft every 8 seconds
+void feed(Entity food)	Feed GuppyCruncher with any level Guppy, increase fullness to the max, kill that Guppy and restore sprite if already hungry
+boolean isFuckingHungry()	@Override Like super.isFuckingHungry() but also set sprite according hungry status
+void startLogic()	@Override Implement GuppyCruncher's Logic (Moving toward food, Idle movement, crafting coin and alien attack, which decrease alien's HP by 75)

Class BeetleMuncher extends Fish implements Feedable, Killable, Craftable, Movable

Field

-boolean isAlreadyHungry	Check whether BeetleMuncher changed sprite according to it hungriness

Constructor

+BeetleMuncher (Tank tank)	Construct a BeetleMuncher with super() and setup animation and start logic
+BeetleMuncher (int x, int y, Tank tank)	Like BeetleMuncher(Tank tank) but also setX(x) and setY(y)

Method

+Entity craft()	Like Guppy's craft method, but craft Pearl instead of coin and craft every 15 seconds

+void feed(Entity food)	Feed BeetleMuncher with Beetle from GuppyCruncher, increase fullness to the max, remove that Beetle (doesn't add money to the tank) and restore sprite if already hungry
+boolean isFuckingHungry()	@Override Like super.isFuckingHungry() but also set sprite according hungry status
+void startLogic()	@Override Implement BeetleMuncher's Logic (Moving toward food, Idle movement, crafting coin and alien attack, which decrease alien's HP by 1000)

Class SpecialFish extends Fish implements Feedable, Killable, Movable

Field

-int exp	SpecialFish's Exp.
-int level	Level of SpecialFish

Constructor

+SpecialFish (Tank tank)	Construct a SpecialFish with super() and setup animation and start logic
+SpecialFish (int x, int y, Tank tank)	Like SpecialFish(Tank tank) but also setX(x) and setY(y)

Method

+void feed(Entity food)	Feed SpecialFish with Food, increase fullness to the max, increase exp base on foodLevel, recalculate level and restore sprite if already hungry
+boolean isFuckingHungry()	@Override Like super.isFuckingHungry() but also set sprite according hungry status
+void startLogic()	@Override Implement BeetleMuncher's Logic (Moving toward food, Idle movement, crafting coin and tank winning, changing scene)

Package entity.pet

Abstract Class Pet extends Entity implements Movable

Constructor

+Pet(Tank tank)	Construct a Pet with super(tank)

+Pet(int x, int y, Tank tank)	Construct a Pet with super(x, y, tank)

Class Niko extends Pet implement Craftable

Field

-Coin coin	Crafted coin reference

Constructor

+Niko (Tank tank)	Construct a Niko with super(tank) and set proper sprite

Method

+Entity craft()	Create a NikoPearl at current position, like Guppy.
+void startLogic()	@Override Setting up logic for Niko (crafting and animation)
+void resetCoin()	@Override Reset coin reference to null

Class Vert extends Pet implement Craftable, Movable

Field

-Coin coin	Crafted coin reference

Constructor

+Vert (Tank tank)	Construct a Vert with super(tank) and set proper sprite
+Vert (int x, int y, Tank tank)	Construct a Vert with super(x, y, tank) and set proper sprite

Method

+Entity craft()	Create a gold coin at current position, like Guppy.
+void startLogic()	@Override

	Setting up logic for Vert (crafting and movement)
+void resetCoin()	@Override Reset coin reference to null

Class Wadsworth extends Pet implement Movable

Field

+ArrayList<Guppy> guppies	List of kept Guppy Lv.0

Constructor

+Wadsworth (Tank tank)	Construct a Wadsworth with super(tank) and set proper sprite

Method

+void saveGuppy(ArrayList<Entity> list)	Keep only Guppy Lv.0 in list, stopLogic(), remove it from the list, and add to this.list (Which can make game ended if there is only lv. 0 Guppy)
+void releaseGuppy()	Add all guppy in the list to tank
+void startLogic()	@Override Setting up logic for Wadsworth (movement, detecting alien and relaseing Guppy)

Class Itchy extends Pet implement Movable

Constructor

+Itchy (Tank tank)	Construct an Itchy with super(tank) and set proper sprite
+Itchy (int x, int y, Tank tank)	Construct an Itchy with super(x, y, tank) and set proper sprite

Method

+void startLogic()	@Override Setting up logic for Itchy (movement and attacking routine)

Class Seymour extends Pet implement Movable

Constructor

+Seymour (Tank tank)	Construct a Seymour with super(tank) and set proper sprite
+Seymour (int x, int y, Tank tank)	Construct a Seymour with super(x, y, tank) and set proper sprite

Method

+void startLogic()	@Override Setting up logic for Seymour, only movement

Package entity.alien

Abstract Class Alien extends Entity implements Movable, Killable, Clickable, Feedable

Field

# int fullness	Alien’s fullness
# int maxFullness	Alien’s maximum value of fullness
# int hp	Alien’s HP

Constructor

+Alien(Tank tank)	Construct a Alien with super(tank) and set default maxFullness to 30
+Alien(int x, int y, Tank tank)	Like Alien(Tank tank), but setX(x) and setY(y)

Method

+boolean isHungry()	Return whether fullness is equal or less than zero
+void onHungry(Entity Food)	Do nothing (for overriding)
+void decreaseHP(int amount)	Decrease alien’s hp by amount (negative amount means healing alien)
+int getHP()	Get alien’s HP (return zero if less than zero)
+void setHP(int hp)	Setter of hp
+void reduceFullness()	Reduce fullness by 3
+int getMaxFullness()	Getter of maxFullness
+int getFullness()	Getter of fullness

+void died()	Remove itself from entities in tank and stop current logic
+void clickAction(MouseEvent event)	Do nothing (default action)

Class BasicAlien extends Alien implements Feedable, Clickable, Killable, Movable

Constructor

+BasicAlien (Tank tank)	Construct an Alien with super(tank) and set default value:
	maxFullness = 30
	hp = 100
	proper sprite and eventlistener for clicking and startLogic()

Method

+void feed(Entity food)	Eat provided Fish and remove from tank and reset Fullness
+void startLogic()	@Override Setup logic for BasicAlien (Idle movement and fish finding)
+void clickAction(MouseEvent event)	@Override Decrease BasicAlien HP by (tank.getGunLevel()^2) * 3 and make alien move away from mouses

Class Gus extends Alien implements Feedable, Killable, Movable

Constructor

+Gus (Tank tank)	Construct Gus with super(tank) and set default value:
	maxFullness = 15
	hp = 2000
	proper sprite and eventlistener for clicking and startLogic()

Method

+void feed(Entity food)	Eat provided Entity (Fish, Food , Coin, Diamond and Pearl), remove from tank and reset Fullness
+void startLogic()	@Override Setup logic for Gus (Idle movement and foods finding)

Class Whale extends BasicAlien implements Clickable, Killable, Movable, Craftable

Field

+ArrayList<LittleAlien> alienList	Contain spawned LittleAlien

Constructor

+Whale (Tank tank)	Construct Whale with super(tank) and set default value:
	maxFullness = 70
	hp = 150000
	proper sprite and eventlistener for clicking and startLogic()

Method

+void feed(Entity food)	Eat provided Fish and spawn LittleAlien in the Fish position
+Entity craft()	Spawn LittleAlien to the tank
+void resetCoin()	@Override Do nothing
+void startLogic()	@Override Setup logic for Whale (Idle movement, Little Alien spawning and foods finding)
+void removeAlien(LittleAlien alien)	Remove alien form alienList

Class LittleAlien extends BasicAlien implements Clickable, Killable, Movable

Field

-Whale whale	Parent reference

Constructor

+LittleAlien (Tank tank, Whale whale)	Construct an Small Alien with super(tank) and set:
	this.whale = whale
	maxFullness = 30
	hp = 250
	set proper sprite (and start logic from super(tank))

+LittleAlien (int x, int y, Tank tank, Whale whale)	Like LittleAlien(Tank tank, Whale whale) but setting X and Y position

Method

+void died()	@Override

	Remove from the tank and whale (parent) and stop logic

Package entity.item

Abstract Class TankItem extends Entity implements Movable, Clickable

Field

-static double fallFactor	fallFactor is a translate time for each unit of time - 1 for normal case - 2 when Seymour present in the tank

Constructor

+TankItem(Tank tank)	Construct a TankItem with super(x, y) and set fallFactor
+TankItem(int x, int y, Tank tank)	Construct a TankItem with super(x, y, tank) and set fallFactor

Method

+boolean isGrounded()	Return true if item is already on the ground
+void onGrounded()	Remove itself from the list
+void clickAction(MouseEvent event)	Do nothing (default action)
+void startLogic()	@Override Item continuously fall to the ground
+Getter & setter for fallFactor	

Class Food extends TankItem implements Movable

Field

-int foodLevel	Level of the food

Constructor

+Food (int x, int y, Tank tank)	Construct a Food with super(x, y, tank) and set foodLevel = tank.getFoodLevel() increase currentFood by 1

Method

+int getFoodLevel()	Getter for foodLevel
+void onGrounded()	@Override Remove from tank and decrease currentFood by 1

Class Coin extends TankItem implements Clickable, Movable

Field

-int level	Level of the coin
-Entity fish	Reference to crafter
-final int[] coinValue	Value of coin in each level

Constructor

+Coin (Tank tank, Entity fish, int level)	Construct a Food with super(tank), set this.level = level, proper sprite and startLogic()
+Coin (int x, int y, Tank tank, Entity fish, int level)	Construct a Food with super(x, y, tank), set this.level = level, proper sprite and startLogic()

Method

+void clickAction(MouseEvent event)	@Override Add money to tank.money respectively to coinValue[level] and remove itself from entity list
+void onGrounded()	@Override Remove itself from tank and reset reference from fish refer above
+int getLevel()	Getter for level
+Entity getParent()	Getter for fish

Class Beetle extends TankItem implements Clickable, Movable

Field

-final int beetleValue	Value of Beetle when collected
-Entity fish	Reference to crafter

Constructor

+Beetle (Tank tank, Entity fish)	Construct a Beetle with super(tank), set proper sprite and startLogic()
+Beetle (int x, int y, Tank tank, Entity fish)	Construct a Beetle with super(x, y, tank), set proper sprite and startLogic()

Method

+void clickAction(MouseEvent event)	@Override Add money to tank.money respectively to beetleValue and remove itself from entity list

+void onGrounded()	@Override Remove itself from tank and reset reference from fish refer above
+boolean isGrounded()	@Override Return true if Beetle reach top of the tank
+void startLogic()	@Override Beetle move up instead of falling down
+Entity getParent()	Getter for fish

Class Diamond extends TankItem implements Clickable, Movable

Field

-final int diamondValue	Value of Diamond when collected
-Entity fish	Reference to crafter

Constructor

+Diamond (Tank tank, Entity fish)	Construct a Diamond with super(tank), set proper sprite and startLogic()
+Diamond (int x, int y, Tank tank, Entity fish)	Construct a Diamond with super(x, y, tank), set proper sprite and startLogic()

Method

+void clickAction(MouseEvent event)	@Override Add money to tank.money respectively to diamondValue and remove itself from entity list
+void onGrounded()	@Override Remove itself from tank and reset reference from fish refer above
+Entity getParent()	Getter for fish

Class Pearl extends TankItem implements Clickable, Movable

Field

-final int pearlValue	Value of Pearl when collected
-Entity fish	Reference to crafter

Constructor

+Pearl (Tank tank, Entity fish)	Construct a Pearl with super(tank), set proper sprite and startLogic()
+Pearl (int x, int y, Tank tank, Entity fish)	Construct a Pearl with super(x, y, tank), set proper sprite and startLogic()

Method

+void clickAction(MouseEvent event)	@Override Add money to tank.money respectively to pearlValue and remove itself from entity list
+void onGrounded()	@Override Remove itself from tank and reset reference from fish refer above
+Entity getParent()	Getter for fish

Class NikoPearl extends TankItem implements Clickable, Movable

Field

-final int pearlValue	Value of Pearl when collected
-Entity fish	Reference to crafter

Constructor

+NikoPearl (Tank tank, Entity fish)	Construct a NikoPearl with super(tank) and set proper sprite
+NikoPearl (int x, int y, Tank tank, Entity fish)	Construct a NikoPearlPearl with super(x, y, tank) and set proper sprite

Method

+void clickAction(MouseEvent event)	@Override Add money to tank.money respectively to pearlValue and remove itself from entity list
+Entity getParent()	Getter for fish

Abstract Class SellingItem implement Clickable

Field

#int price	Selling price of each items
#Tank tank	Reference for current tank
#AnimatedObject sprite	Item thumbnail

Constructor

+SellingItem(Tank tank)	Construct a SellingItem, setting this.tank = tank

Method

+int getPrice	Getter for price

+void setPrice(int price)	Setter for price
+AnimatedObject getSprite()	Getter for sprite

Class GuppySeller extends SellingItem

Constructor

+GuppySeller (Tank tank)	Construct a GuppySeller with super(tank), set price = 100 and setup sprite

Method

+void clickAction(MouseEvent event)	Construct a Guppy and add it to Tank

Class BeetleSeller extends SellingItem

Constructor

+BeetleSeller (Tank tank)	Construct a BeetleSeller with super(tank), set price = 2000 and setup sprite

Method

+void clickAction(MouseEvent event)	Construct a BeetleMuncher and add it to Tank

Class CarnivoreSeller extends SellingItem

Constructor

+CarnivoreSeller (Tank tank)	Construct a CarnivoreSeller with super(tank), set price = 2000 and setup sprite

Method

+void clickAction(MouseEvent event)	Construct a Carnivore and add it to Tank

Class GuppyCruncherSeller extends SellingItem

Constructor

+GuppyCruncherSeller (Tank tank)	Construct a GuppyCruncherSeller with super(tank), set price = 750 and setup sprite

Method

+void clickAction(MouseEvent event)	Construct a GuppyCruncher and add it to Tank

Class SpecialFishSeller extends SellingItem

Field

#final int[] levelPrice	Selling price of SpecialFish in different tank

Constructor

+SpecialFishSeller (Tank tank, int level)	Construct a SpecialFishSeller with super(tank), set price = levelPrice[level - 1] and setup sprite

Method

+void clickAction(MouseEvent event)	Construct a SpecialFish and add it to Tank

Class FoodLevelSeller extends SellingItem

Constructor

+FoodLevelSeller (Tank tank)	Construct a FoodLevelSeller with super(tank), set price = 300 and setup sprite

Method

+void clickAction(MouseEvent event)	Add tank.foodLevel by calling tank.increaseFoodLevel(), change sprite according to its level and play sound for buying item

Class GunSeller extends SellingItem

Constructor

+GunSeller (Tank tank)	Construct a GunSeller with super(tank), set price = 300 and setup sprite

Method

+void clickAction(MouseEvent event)	Call tank.increaseGunLevel() and play sound for buying item

Class MaxFoodSeller extends SellingItem

Constructor

+MaxFoodSeller (Tank tank)	Construct a MaxFoodSeller with super(tank), set price = 300 and setup sprite

Method

+void clickAction(MouseEvent event)	Call tank.increaseMaxFood() and play sound for buying item

Package main

Class Main extends Application

Field

+static final int SCREEN_WIDTH	Window's width
+static final int SCREEN_HEIGHT	Window's height
+static final int FPS	Refreshing rate (Frame per second)
+static SpriteLoader sprite	Construct SpriteLoader to load ImageView and MusicPlayer
-static Stage primaryStage	primaryStage for game screen
-MainMenu menu	MainMenu reference

Method

+start(Stage primaryStage)	Setup stage to not resizeable and not full screen, then set screen size according to SCREEN_WIDTH and SCREEN_HEIGHT, set screen title and show it.
+static void main(String[] args)	launch(args)

Package gui

Abstract class View

Field

#Scene scene	Scene to be show

Method

+Scene getScene()	Getter for scene

Class MainMenu extends View

Field

-static final int numCols	Number of columns in GridPane
-static final int numRows	Number of rows in GridPane
-static ImageView background	Background for main menu
-static ImageView startButton	Start button image
-static ImageView exitButton	Exit button image
-static TankSelect tankSelect	Reference for TankSelect screen

Constructor

+MainMenu()	- Load all ImageViews - Setup BGM - Setup eventlisteners for buttons - Put all button to root and at to scene

Method

+static void setBGM(String fileName)	Set background music from fileName (stop previous sound and then load new MusicPlayer which come from AudioLoader)
+static TankSelect getTankSelect()	Getter for tankSelect

Class PetSelect extends View

Field

-String bgFile	Background image file name
-TilePane tilePane	TilePane for pet choosing buttons
-BorderPane root	Root pane
-Canvas background	Canvas for drawing background
-Canvas go	Canvas for Go button
-StackPane goPane	StackPane for Go button to be insert to the root
-Tank tank	Tank to be loaded reference
-ArrayList<Pet> selectedPets	Selected pet arrayLists to be insert to tank
-FlowPane bottomPane	Bottom section for Go and Back button

Constructor

+PetSelect() throws NullPictureException	- Setup each section size - Construct all pets - Add button to select pet (if pet is already selected, remove it from selectedPets)

	- Setup evenlistener for hovering and clicking buttons - Put all button and pane to root and at to scene

Class TankSelect extends View

Field

-static final int numCols	Number of columns in GridPane
-static final int numRows	Number of rows in GridPane
-static final int THUMB_INDEX	Tank thumbnail’s index in levelButton
-static final int TEXT_INDEX	Tank overlay stuffs index in levelButton
-static ArrayList<StackPane> levelButton	Tank selection buttons
-static ImageView clicked	Store clicked button
-static ImageView bg	Background image

Constructor

+PetSelect() throws NullPictureException	- Setup each section size - Construct all button and add to levelButton & root - Setup evenlistener for hovering and clicking buttons 1 times and 2 times - Put all button and pane to root and at to scene

Class TankView extends View

Field

-static final int numCols	Number of columns in GridPane
-static final int numRows	Number of rows in GridPane
-Tank tank	Current stage reference
-StackPane topMenu	Item buying section
-StackPane tankZone	Tank section (which contain all entities)
-Text mainMenuText	Text for returning to tank select screen
-VBox menu	Menu which contain current money and mainMenuText
-Text moneyText	Current money text
-AnimationTimer logic	Logic controller for tank
-StackPane foodSeller	foodSeller stackPane which can be removed
-StackPane maxFoodSeller	maxFoodSeller stackPane which can be removed
-StackPane gunSeller	gunSeller stackPane which can be removed

Constructor

+TankView(Tank newTank)	- Setup each section size - Construct and add all pane to root which also setting up tank specificaton - Setup logic, which contain clicking collectable items, feeding fish, attacking alien refresh current money text and refresh all sprite - Setup event listeners - Start pets logic
+StackPane getGrid(SellingItem s)	Construct Item selling button from SellingItem

Class EndingScene

Field

-static Scene scene	Ending scene to be returned
-boolean isLose	Determine that what type of ending should be loaded

Methods

+static Scene getScene(int n)	Construct Ending Scene according to n (1 for lose and 0 for win) and add button for returning to tank select screen