Spidraw Documentation

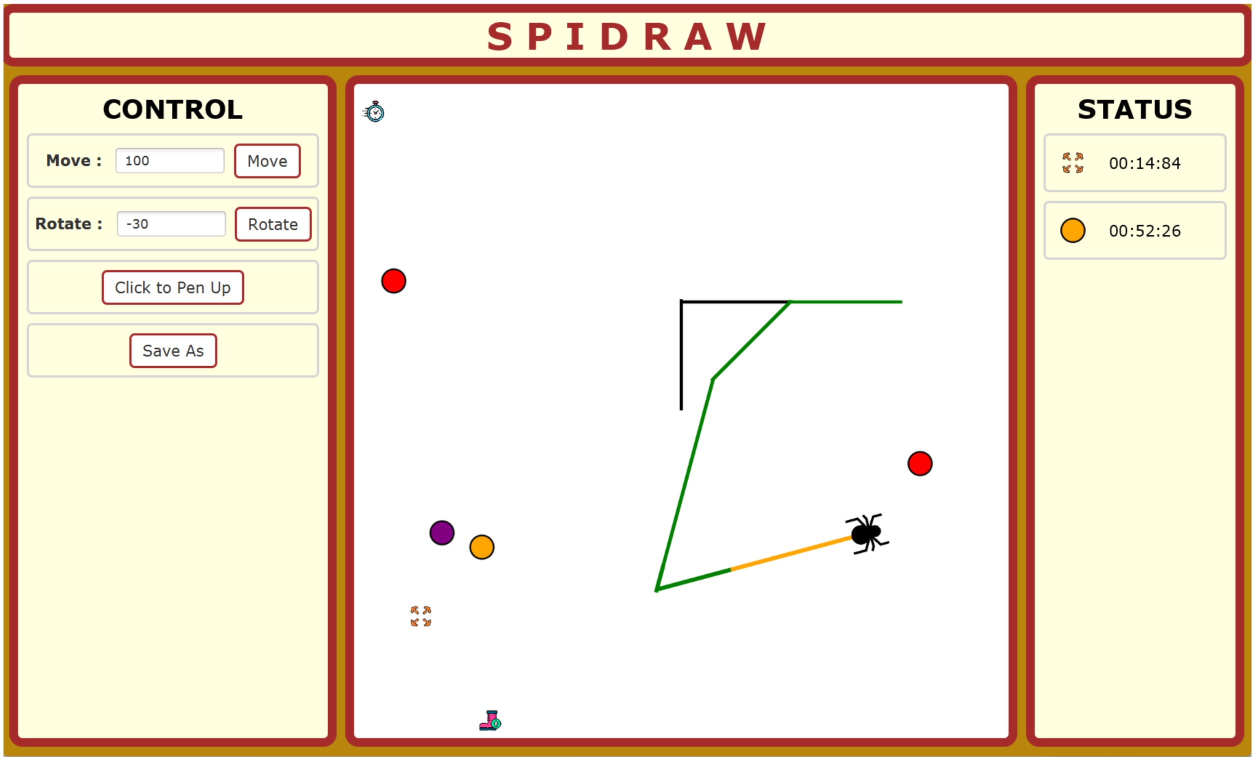
Created By

Thammarit Phiwton 6432071421

2110215 Programming Methodology
Semester 2 Year 2021
Chulalongkorn University

Introduction

Spidraw is inspired from the program name “Turtle Graphics”, that program is about programming the turtle to draw something, the popular way to introduce programming to kids. I took that idea and made it into a game (and use the spider instead of the turtle).



Move : 100

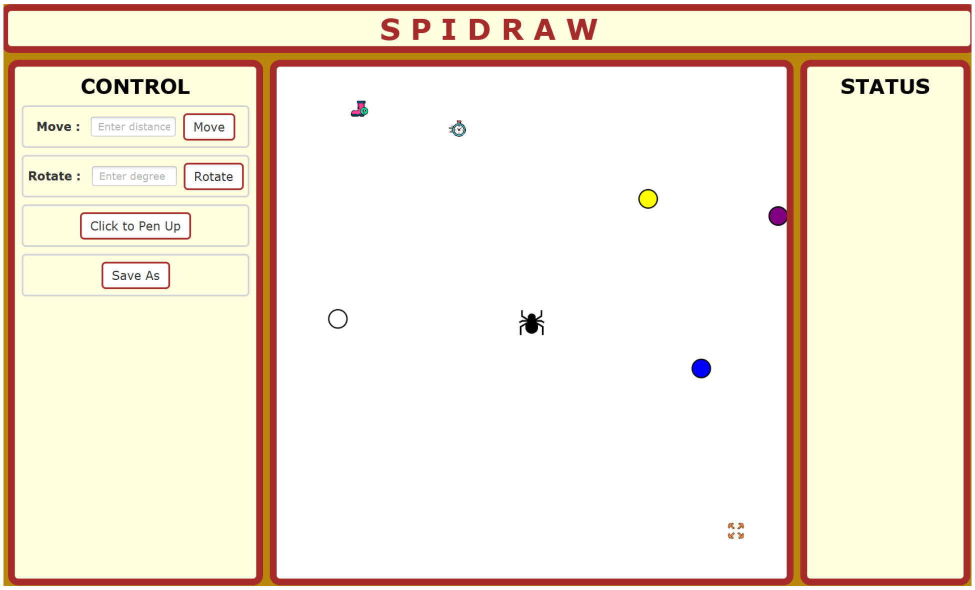
Rotate : -30

00:14:84

00:52:26

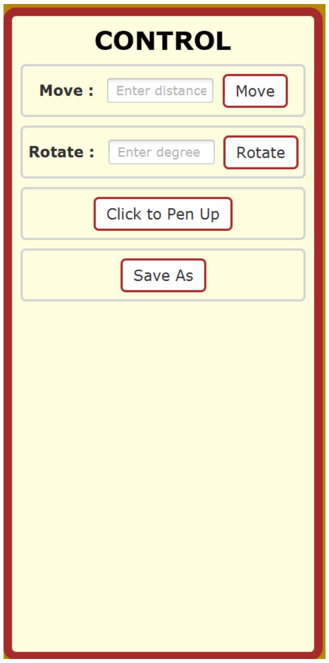
The example screenshot of Spidraw.

Gameplay



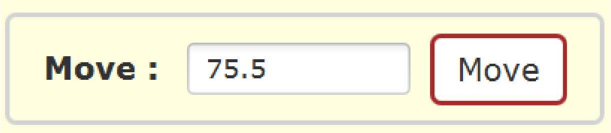
When you open the game, the player (spider) will be summoned in the center of the canvas 600 x 600 pixels.

Control Bar



You can control the game by using the control bar in the left sidebar.

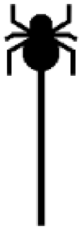
Moving



Move : 75.5

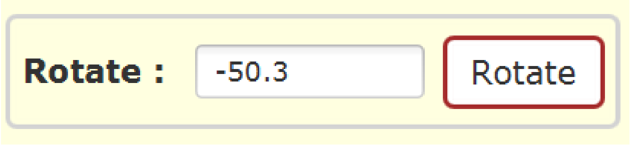
you can enter the distance you want, to move in pixel units.

- If the distance > 0, move forward.
- If the distance < 0, move backward.



The example, when the player (spider) moves 75.5 pixels.

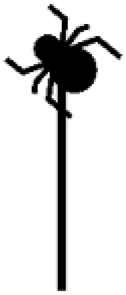
Rotation



Rotate : -50.3

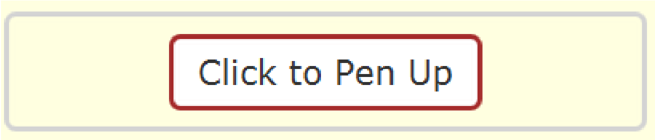
You can enter the degree you want, to rotate in degree units.

- If the degree > 0, rotate clockwise.
- If the degree < 0, rotate counterclockwise.



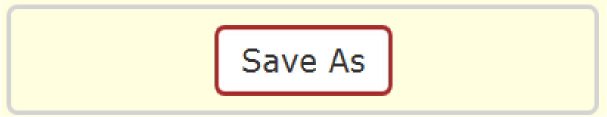
The example, when the player (spider) moves 75.5 pixels and rotates -50.3 degrees.

Pen up / Pen down



You can click the button to pen up (ยกปากกา) to skip drawing.
You can click it again to pen down (วางปากกา) to continue drawing.

Save



You can save the canvas as a png file by clicking this button.

Items

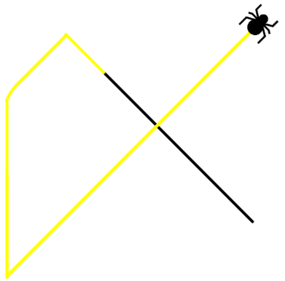
ColorItem, changes the player pencolor to the color of the item by 1 minute.

SpeedUpItem, increases the player speed by 30 seconds.

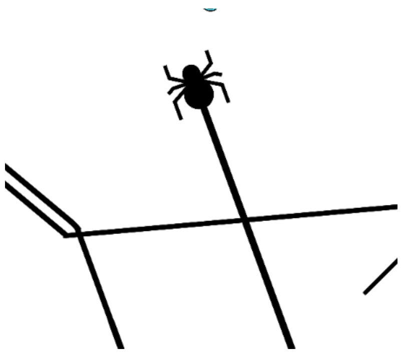
SizeUpItem, increases the player size (drawing line width) by 30 seconds.

IncreasingTimeItem, increases the other items' status time by 30 seconds.

SpeedUpItem, SizeUpItem and IncreasingTimeItem **can stack buffs**.

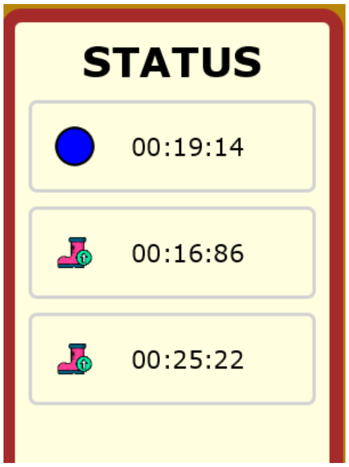


The example of collecting the yellow ColorItem.



The example of collecting the SizeUpItem x4 (notice that the line slightly bigger)

Status Bar



STATUS

● 00:19:14

00:16:86

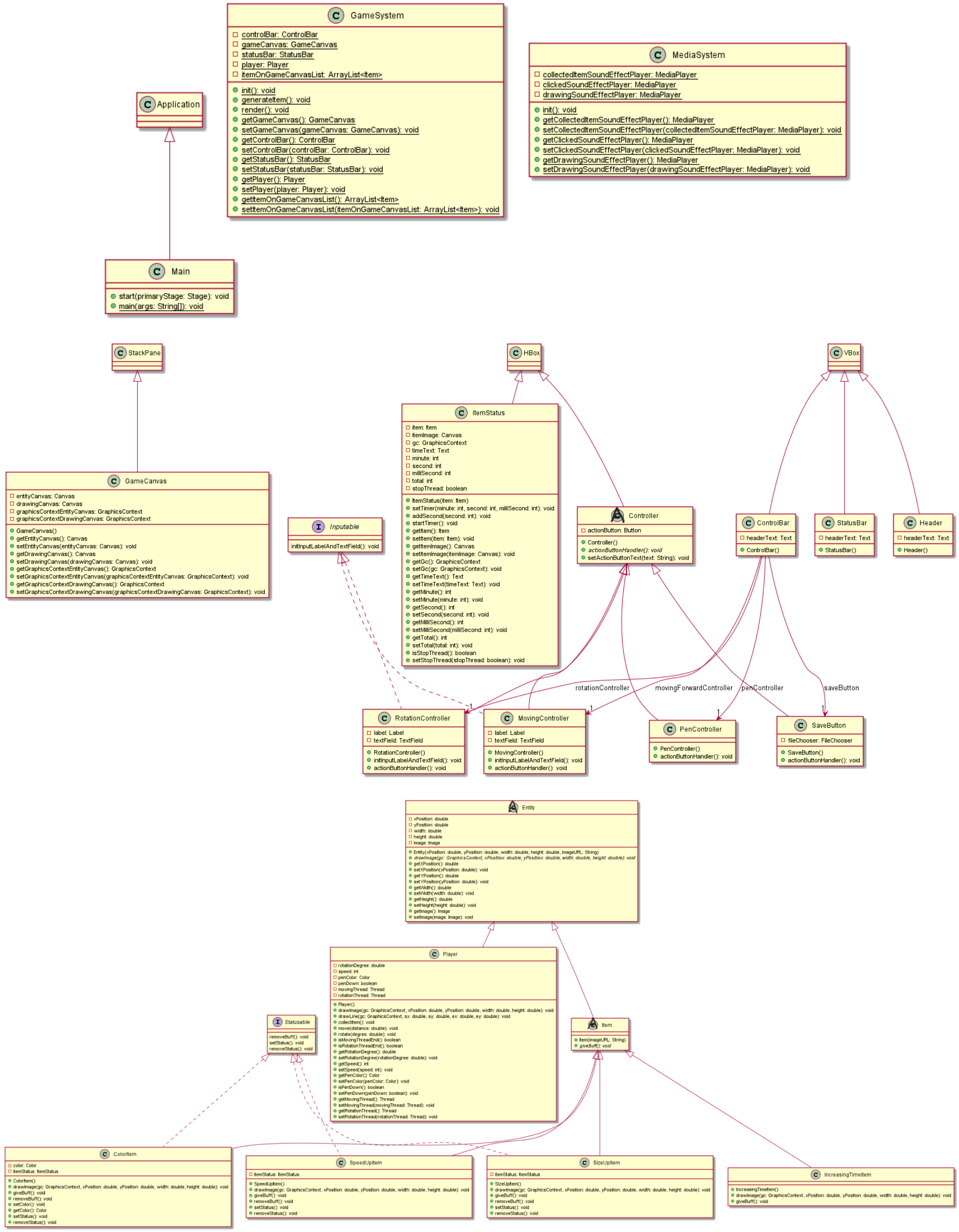
00:25:22

When you collect an item, you will receive a status that shows the remaining time of the item.

When you run the JAR file, you **must** use this command.

```
java -jar --module-path “<path to javafx>" --add-modules
javafx.controls,javafx.fxml,javafx.media,javafx.swing <File name>
```

UML Diagram


GameSystem
controlBar: ControlBar
gameCanvas: GameCanvas
statusBar: StatusBar
player: Player
itemOnGameCanvasList: ArrayList<Item>
init(): void
generateItem(): void
render(): void
getGameCanvas(): GameCanvas
setGameCanvas(gameCanvas: GameCanvas): void
getControlBar(): ControlBar
setControlBar(controlBar: ControlBar): void
getStatusBar(): StatusBar
setStatusBar(statusBar: StatusBar): void
getPlayer(): Player
setPlayer(player: Player): void
getItemOnGameCanvasList(): ArrayList<Item>
setItemOnGameCanvasList(itemOnGameCanvasList: ArrayList<Item>): void
MediaSystem
collectedItemSoundEffectPlayer: MediaPlayer
clickedSoundEffectPlayer: MediaPlayer
drawingSoundEffectPlayer: MediaPlayer
init(): void
getCollectedItemSoundEffectPlayer(): MediaPlayer
setCollectedItemSoundEffectPlayer(collectedItemSoundEffectPlayer: MediaPlayer): void
getClickedSoundEffectPlayer(): MediaPlayer
setClickedSoundEffectPlayer(clickedSoundEffectPlayer: MediaPlayer): void
getDrawingSoundEffectPlayer(): MediaPlayer
setDrawingSoundEffectPlayer(drawingSoundEffectPlayer: MediaPlayer): void
Application
Main
start(primaryStage: Stage): void
main(args: String[]): void
StackPane
GameCanvas
entityCanvas: Canvas
drawingCanvas: Canvas
graphicsContextEntityCanvas: GraphicsContext
graphicsContextDrawingCanvas: GraphicsContext
GameCanvas()
getEntityCanvas(): Canvas
setEntityCanvas(entityCanvas: Canvas): void
getDrawingCanvas(): Canvas
setDrawingCanvas(drawingCanvas: Canvas): void
getGraphicsContextEntityCanvas(): GraphicsContext
setGraphicsContextEntityCanvas(graphicsContextEntityCanvas: GraphicsContext): void
getGraphicsContextDrawingCanvas(): GraphicsContext
setGraphicsContextDrawingCanvas(graphicsContextDrawingCanvas: GraphicsContext): void
HBox
ItemStatus
item: Item
itemImage: Canvas
gc: GraphicsContext
timeText: Text
minute: int
second: int
milliSecond: int
total: int
stopThread: boolean
ItemStatus(item: Item)
setTimer(minute: int, second: int, milliSecond: int): void
addSecond(second: int): void
startTimer(): void
getItem(): Item
setItem(item: Item): void
getItemImage(): Canvas
setItemImage(itemImage: Canvas): void
getGc(): GraphicsContext
setGc(gc: GraphicsContext): void
getTimeText(): Text
setTimeText(timeText: Text): void
getMinute(): int
setMinute(minute: int): void
getSecond(): int
setSecond(second: int): void
getMilliSecond(): int
setMilliSecond(milliSecond: int): void
getTotal(): int
setTotal(total: int): void
isStopThread(): boolean
setStopThread(stopThread: boolean): void
VBox
Inputable
initInputLabelAndTextField(): void
Controller
actionButton: Button
Controller()
actionButtonHandler(): void
setActionButtonText(text: String): void
ControlBar
headerText: Text
ControlBar()
StatusBar
headerText: Text
StatusBar()
Header
headerText: Text
Header()
rotationController
movingForwardController
penController
saveButton
1
1
1
1
RotationController
label: Label
textField: TextField
RotationController()
initInputLabelAndTextField(): void
actionButtonHandler(): void
MovingController
label: Label
textField: TextField
MovingController()
initInputLabelAndTextField(): void
actionButtonHandler(): void
PenController
PenController()
actionButtonHandler(): void
SaveButton
fileChooser: FileChooser
SaveButton()
actionButtonHandler(): void
Entity
xPosition: double
yPosition: double
width: double
height: double
image: Image
Entity(xPosition: double, yPosition: double, width: double, height: double, imageURL: String)
drawImage(gc: GraphicsContext, xPosition: double, yPosition: double, width: double, height: double): void
getXPosition(): double
setXPosition(xPosition: double): void
getYPosition(): double
setYPosition(yPosition: double): void
getWidth(): double
setWidth(width: double): void
getHeight(): double
setHeight(height: double): void
getImage(): Image
setImage(image: Image): void
Player
rotationDegree: double
speed: int
penColor: Color
penDown: boolean
movingThread: Thread
rotationThread: Thread
Player()
drawImage(gc: GraphicsContext, xPosition: double, yPosition: double, width: double, height: double): void
drawLine(gc: GraphicsContext, sx: double, sy: double, ex: double, ey: double): void
collectItem(): void
move(distance: double): void
rotate(degree: double): void
isMovingThreadEnd(): boolean
isRotationThreadEnd(): boolean
getRotationDegree(): double
setRotationDegree(rotationDegree: double): void
getSpeed(): int
setSpeed(speed: int): void
getPenColor(): Color
setPenColor(penColor: Color): void
isPenDown(): boolean
setPenDown(penDown: boolean): void
getMovingThread(): Thread
setMovingThread(movingThread: Thread): void
getRotationThread(): Thread
setRotationThread(rotationThread: Thread): void
Statusable
removeBuff(): void
setStatus(): void
removeStatus(): void
Item
Item(imageURL: String)
giveBuff(): void
ColorItem
color: Color
itemStatus: ItemStatus
ColorItem()
drawImage(gc: GraphicsContext, xPosition: double, yPosition: double, width: double, height: double): void
giveBuff(): void
removeBuff(): void
setColor(): void
getColor(): Color
setStatus(): void
removeStatus(): void
SpeedUpItem
itemStatus: ItemStatus
SpeedUpItem()
drawImage(gc: GraphicsContext, xPosition: double, yPosition: double, width: double, height: double): void
giveBuff(): void
removeBuff(): void
setStatus(): void
removeStatus(): void
SizeUpItem
itemStatus: ItemStatus
SizeUpItem()
drawImage(gc: GraphicsContext, xPosition: double, yPosition: double, width: double, height: double): void
giveBuff(): void
removeBuff(): void
setStatus(): void
removeStatus(): void
IncreasingTimeItem
IncreasingTimeItem()
drawImage(gc: GraphicsContext, xPosition: double, yPosition: double, width: double, height: double): void
giveBuff(): void

Implementation Detail

1. Package application

1.1 public class **Main** extends Application

This class is the starting point of the program.

Method

Name	Description
+ void start(Stage primaryStage) throws Exception	- Initialize all GUI classes. - Initialize Player. - Initialize GameSystem. - Initialize MediaSystem. - Set all GUI and player variables to GameSystem. - Set the background color as a dark goldenrod color. - Set alignment as center. - Set spacing as 8 - Set padding as 8 - Set all GUI to scene. - Set scene to primaryStage - Set primaryStage title as “Spidraw” - Show primaryStage
+ static void main(String[] args)	The main method for the application.

2. Package gui

This package contains subpackages that are about GUI.

2.1 Package gui.header

2.1.1 public class **Header** extends VBox

This class is the header in the program window.

Field

Name	Description
- Text headerText	The text that shows in the header as “S P I D R A W”.

Constructor

Name	Description
+ Header()	- Set alignment as center. - Set border as brown color, solid, corner radius is 8 and border width is 8. - Set background color as lightyellow color. - Initialize headText. - Set text of headerText as “S P I D R A W” - Set font of headerText as Verdana family, bold, size 34 and brown color. - Add headerText to Header.

2.2 Package gui.canvas

2.2.1 public class **GameCanvas** extends StackPane

This class is extended from StackPane which contains 2 layers, EntityCanvas layer and DrawingCanvas layer.

- EntityCanvas is used for drawing the entities.
- DrawingCanvas is used for actual drawing from the player(spider).

Field

Name	Description
- Canvas entityCanvas	The canvas is used for drawing the entities.
- Canvas drawingCanvas	The canvas is used for actual drawing from the player(spider).
- GraphicsContext graphicsContextEntityCanvas	The graphics context of entityCanvas.
- GraphicsContext graphicsContextDrawingCanvas	The graphics context of drawingCanvas.

Constructor

Name	Description
+ GameCanvas()	- Set preferred width as 600. - Set preferred height as 600. - Set border as brown color, solid, corner radius is 8, border width is 8. - Set background color as white color. - Initialize entityCanvas and drawingCanvas as 600 x 600. - Initialize

	graphicsContextEntityCanvas and graphicsContextDrawingCanvas. - Add all canvases to GameCanvas.

Method

Name	Description
+ Getters / Setters	

2.3 Package gui.control

2.3.1 public abstract class **Controller** extends HBox

This class is the base of all controllers UI which are used for controlling the program. This class is extended from HBox to display as a horizontal row.

Field

Name	Description
- Button actionButton	The button that when it's clicked it will handle some actions to control the program.

Constructor

Name	Description
+ Controller()	- Set preferred height as 50. - Set border as lightgrey color, solid, corner radius is 4 and border width is 2. - Set alignment as center.

	- Set spacing as 8. - Initialize actionButton. - Set border of actionButton as brown color, solid, corner radius is 4 and border width is 2. - Set background color of actionButton as white color. - Set onMouseClicked for actionButton. - Set actionButton to Controller.

Method

Name	Description
+ abstract void actionButtonHandler()	Handle some actions when the actionButton is clicked.
+ void setActionButtonText(String text)	- Set text for actionButton. - Set font of actionButton as Verdana family, normal weight and size 14.

2.3.2 public Interface **Inputable**

This interface defines a method for Controllers that need input.

Method

Name	Description
+ void initInputLabelAndTextField()	Create a label and textField.

2.3.3 public class **MovingController** extends Controller implements Inputable

This class is a UI for receiving the input as a distance and sending it to the player(spider) to process moving.

Field

Name	Description
- Label label	The label for describing what the UI does.
- TextField textField	The field for receiving the input.

Constructor

Name	Description
+ MovingController()	- setActionButtonText as “Move” - Call initInputLabelAndTextField() method.

Method

Name	Description
+ void initInputLabelAndTextField()	- Initialize label. - Set text of label as “Move : ”. - Set font of label as Verdana family, bold and size 14. - Initialize textField. - Set prompt text of textField as “Enter distance” - Set preferred width of textField as 100. - Set font of textField as Verdana family, normal

	weight and size 12. - Add label and textField to MovingController.
+ void actionButtonHandler()	- Play clickedSoundEffect. - If the input is invalid, alert the message “Please Enter Decimal Number”. - If the player(spider) is currently moving, alert the message “You Need To Wait Until Moving End!!!". - Otherwise, send the distance to the player(spider) to move.

2.3.4 public class **RotationController** extends Controller implements Inputable

This class is a UI for receiving the input as degrees and sending it to the player(spider) to process rotating.

Field

Name	Description
- Label label	The label for describing what the UI does.
- TextField textField	The field for receiving the input.

Constructor

Name	Description
+ RotationController()	- setActionButtonText as “Rotate” - Call initInputLabelAndTextField() method.

Method

Name	Description
+ void initInputLabelAndTextField()	- Initialize label. - Set text of label as “Rotate : ”. - Set font of label as Verdana family, bold and size 14. - Initialize textField. - Set prompt text of textField as “Enter degree” - Set preferred width of textField as 100. - Set font of textField as Verdana family, normal weight and size 12. - Add label and textField to MovingController.
+ void actionButtonHandler()	- Play clickedSoundEffect. - If the input is invalid, alert the message “Please Enter Decimal Number”. - If the player(spider) is currently rotating, alert the message “You Need To Wait Until Rotation End!!!". - Otherwise, send the degrees to the player(spider) to rotate.

2.3.5 public class **PenController** extends Controller

This class is a UI for setting the pen of the player up (doesn’t draw) or down (does draw).

Constructor

Name	Description
+ PenController()	- setActionButtonText as “Click to Pen Up”. - Set the pen of the player down (does draw).

Method

Name	Description
+ void actionButtonHandler()	- Play clickedSoundEffect. - If the pen of the player is down, set the pen of the player up and setActionButtonText as “Click to Pen Down”. - If the pen of the player is up, set the pen of the player down and setActionButtonText as “Click to Pen Up”.

2.3.6 public class **SaveButton** extends Controller

This class is used for saving the drawingCanvas to a png file.

Field

Name	Description
- FileChooser fileChooser	- Open file chooser.

Constructor

Name	Description
+ SaveButton()	- setActionButtonText as “Save As”. - Initialize fileChooser. - Add the extension filter to fileChooser as png. - Set InitialFileName as “canvas”.

Method

Name	Description
+ void actionButtonHandler()	- Play clickedSoundEffect. - Show save dialog from fileChooser. - Snapshot the drawingCanvas to the WritableImage and save that image as a png file.

2.3.7 public class **ControlBar** extends VBox

This class is the left side bar, containing all Controllers. This class is extended from VBox to display as a vertical column.

Field

Name	Description
- MovingController movingController	A controller that controls moving.
- RotationController rotationController	A controller that controls rotation.
- PenController	A controller that controls the pen

penController	of the player.
- SaveButton saveButton	A controller that controls saving drawingCanvas to the image as a png file.
- Text headerText	The text that shows “CONTROL” in the header.

Constructor

Name	Description
+ ControlBar()	- Set preferred width as 300. - Set preferred height as 600. - Set border as brown color, solid, corner radius is 8 and border width is 8. - Set background color as lightyellow color. - Set padding as 8. - Set spacing as 8. - Set alignment as center. - Initialize headerText. - Set text of headerText as “CONTROL”. - Set font of headerText as Verdana family, bold and size 24. - Initialize all controllers. - Add all controllers and headerText to ControlBar.

2.4 Package gui.status

2.4.1 public class **ItemStatus** extends HBox

This class is a UI for showing the status (countdown time) of an item collected by the player(spider).

Field

Name	Description
- Item item	The item is collected by the player.
- Canvas itemImage	The canvas for showing an item image.
- GraphicsContext gc	The graphics context of itemImage.
- Text timeText.	The text for showing countdown time.
- int minute	The minute of the timer.
- int second	The second of the timer.
- int milliSecond	The millisecond of the timer.
- int total	The sum of minute, second and milliSecond as a millisecond unit.
- boolean stopThread	The state of a thread whether stopping or not.

Constructor

Name	Description
+ ItemStatus(Item item)	- Set preferred height as 50.

	- Set border as lightgrey color, solid, corner radius is 4 and border width is 2. - Set spacing as 8. - Set alignment as center left. - Initialize item. - Initialize itemImage as 50 x 50. - initialize gc. - draw itemImage. - initialize timeText. - Set font of timeText as Verdana, normal weight and size 14. - initialize stopThread as false. - Add itemImage and timeText to ItemStatus.

Method

Name	Description
+ void setTimer(int minute, int second, int milliSecond)	initialize the variables and convert them to total millisecond.
+ void addSecond(int second)	Add total time by second.
+ void startTimer()	- Start a thread to countdown. - Set text of timeText by current time. - If the countdown is over, remove buff and remove status.
+ Getters / Setters	

2.4.2 public class **StatusBar** extends VBox

This class is the right side bar, containing all ItemStatus. This class is extended from VBox to display as a vertical column.

Field

Name	Description
- Text headerText	The text that shows “STATUS” in the header.

Constructor

Name	Description
+ StatusBar(Item item)	- Set preferred width as 200. - Set preferred height as 600. - Set border as brown color, solid, corner radius is 8 and border width is 8 - Set background color as lightyellow color. - Set padding as 8. - Set spacing as 8. - Set alignment as center. - Initialize headerText. - Set text of headerText as “STATUS” - Set font of headerText as Verdana family, bold and size 24. - Add headerText to StatusBar.

3. Package logic

This package contains subpackages that are about logic.

3.1 Package logic.entity

This package provides the entities that will appear on the entityCanvas and also contain their logic.

3.1.1 public abstract class **Entity**

This class is the base class for all entities in the game.

Field

Name	Description
- double xPosition	The position of the entity in x coordinate.
- double yPosition	The position of the entity in y coordinate.
- double width	The width of the entity.
- double height	The height of the entity.

Constructor

Name	Description
+ Entity(double xPosition, double yPosition, double width, double height, String imageURL)	- Initialize and set xPosition. - Initialize and set yPosition. - Initialize and set width. - Initialize and set height. - Initialize and set image by using imageURL as the reference.

Method

Name	Description

+ abstract void drawImage(GraphicsContext gc, double xPosition, double yPosition, double width, double height)	Draw an entity image into entityCanvas.
+ Getters / Setters	

3.1.2 public class **Player** extends Entity

This class contains essential logic for the player or the spider.

Field

Name	Description
- double rotationDegree	The rotation degree of the player.
- int speed	The speed of the player.
- Color penColor	The pen's color of the player.
- boolean penDown	The state of the player whether pen down (does draw) or not.
- Thread movingThread	The thread that runs the move method.
- Thread rotationThread	The thread that runs the rotate method.

Constructor

Name	Description
+ Player()	(Use super(...)) - Set xPosition as entityCanvas width / 2. - Set yPosition as entityCanvas width / 2. - Set width as 30. - Set height as 30.

	- Set image by using spider.png. - initialize rotationDegree and set as 90. - initialize speed and set as 1. - initialize movingThead and rotationThread.

Method

Name	Description
+ void drawImage(GraphicsContext gc, double xPosition, double yPosition, double width, double height)	Draw the image at xPosition and yPosition and consider the rotationDegree of the player in drawing.
+ void drawLine(GraphicsContext gc, double sx, double sy, double ex, double ey)	Draw the line from (sx, xy) to (ex, ey) by using color as penColor and using linewidth as the player's width * 0.1.
+ void collectItem()	If there is an item under a distance of 30 pixels near the player, do the following. - play collectedItemSoundEffect - The Item giveBuff. - If the item is an instance of Statusable, cast it to Statusable and setStatus. - Remove the item from the gameCanvas. - Generate new items.
+ void move(double distance)	- Play drawingSoundEffect. - Set the player position by current speed. - If the pen is down,

	drawLine. - Call collectItem. - Render. (Repeat the above until the player reaches the distance.) (Use movingThread to repeat)
+ void rotate(double degree)	- Set the player rotationDegree by current speed. - Render. (Repeat the above until the player reaches the degree.) (Use rotationThread to repeat)
+ Getters / Setters	

3.1.3 public abstract class **Item** extends Entity

This class is the base for all Items.

Constructor

Name	Description
+ Item(String imageURL)	(Use super(...)) - Set xPosition by random. - Set yPosition by random. - Set width as 20 - Set height as 20. - Set image by using imageURL as the reference.

Method

Name	Description
+ abstract void giveBuff()	Give the specific buff to the player.

3.1.4 public interface **Statusable**

This interface defines the item that needs status. (All items except IncreasingTimeItem).

Method

Name	Description
+ void removeBuff()	Remove a buff from the player.
+ void setStatus()	Set ItemStatus.
+ void removeStatus()	Remove ItemStatus from StatusBar

3.1.5 public class **ColorItem** extends Item implements Statusable

This class is the item that can change the player penColor.

Field

Name	Description
- Color color	The color of the item.
- ItemStatus itemStatus	The UI for showing the status of the item.

Constructor

Name	Description

+ ColorItem()	(Use super(...)) - Set image by using blank.png. - Initialize and set random color (using setColor method). - Initialize itemStatus.

Method

Name	Description
+ setColor()	Random a color within {white, red, blue, yellow, green, greenyellow, orange, purple, pink}.
+ void drawImage(GraphicsContext gc, double xPosition, double yPosition, double width, double height)	- Draw Image by using the color. - Draw Image as circle. - Draw at xPosition and yPosition.
+ void giveBuff()	Set the player penColor by the color.
+ void removeBuff()	Set the player penColor to black color (default).
+ void setStatus()	- If there is already ColorItem status, remove that status. - setTimer by 1 minute, 0 second, 0 milliSecond. - Add itemStatus to StatusBar. - startTimer.
+ removeStatus()	- Remove itemStatus from StatusBar.

3.1.6 public class **SpeedUpItem** extends Item implements Statusable

This class is the item that can increase the player speed.

Field

Name	Description
- ItemStatus itemStatus	The UI for showing the status of the item.

Constructor

Name	Description
+ SpeedUpItem()	(Use super(...)) - Set image by using speed_up.png. - Initialize itemStatus.

Method

Name	Description
+ void drawImage(GraphicsContext gc, double xPosition, double yPosition, double width, double height)	- Draw the image. - Draw at xPosition and yPosition.
+ void giveBuff()	Increase the player speed by 1.
+ void removeBuff()	Decrease the player speed by 1.
+ void setStatus()	- setTimer by 0 minute, 30 second, 0 milliSecond. - Add itemStatus to StatusBar. - startTimer.

+ removeStatus()	- Remove itemStatus from StatusBar.

3.1.7 public class **SizeUpItem** extends Item implements Statusable

This class is the item that can increase the player size (that makes the linewidth bigger).

Field

Name	Description
- ItemStatus itemStatus	The UI for showing the status of the item.

Constructor

Name	Description
+ SizeUpItem()	(Use super(...)) - Set image by using size_up.png. - Initialize itemStatus.

Method

Name	Description
+ void drawImage(GraphicsContext gc, double xPosition, double yPosition, double width, double height)	- Draw the image. - Draw at xPosition and yPosition.
+ void giveBuff()	Increase the player width and height by 5.

+ void removeBuff()	Decrease the player width and height by 5.
+ void setStatus()	- setTimer by 0 minute, 30 second, 0 milliSecond. - Add itemStatus to StatusBar. - startTimer.
+ removeStatus()	- Remove itemStatus from StatusBar.

3.1.8 public class **IncreasingTimeItem** extends Item

This class is the item that increase the other items’ status time.

Constructor

Name	Description
+ IncreasingTimeItem()	(Use super(...)) - Set image by using clock.png.

Method

Name	Description
+ void drawImage(GraphicsContext gc, double xPosition, double yPosition, double width, double height)	- Draw the image. - Draw at xPosition and yPosition.
+ void giveBuff()	Increase the other items' status time by 30 second.

3.2 Package logic.game

3.2.1 public class **GameSystem**

This class gathers several classes into it, making it easier to manipulate multiple classes. This class also contains global logic or methods.

Field

Name	Description
- static ControlBar controlBar	The left side bar, containing all Controllers.
- static GameCanvas gameCanvas	The canvas.
- static StatusBar statusBar	The right side bar, containing itemStatus.
- static Player player	The player or the spider.
- static ArrayList<Item> itemOnGameCanvasList	The list that contains items on the gameCanvas.

Method

Name	Description
+ static void init()	- Initialize itemOnGameCanvasList - Generate new items. - render.
+ static void generateItem()	- Generate random items. - Possibility of ColorItem is 30%, SpeedUpItem is 30%, SizeUpItem is 20% and IncreasingTimeItem is 20%. - Limit of items in gameCanvas is 7.

+ static void render()	- clear entityCanvas. - Draw Images of the player and all items.
+ Getters / Setters	

4. Package media

4.1 public class **MediaSystem**

This class manages sound effects.

Field

Name	Description
- static MediaPlayer collectedItemSoundEffectPlayer	The sound effect when an item is collected.
- static MediaPlayer clickedSoundEffectPlayer	The sound effect when UI is clicked.
- static static MediaPlayer drawingSoundEffectPlayer;	The sound effect when the player is drawing.

Method

Name	Description
+ static void init()	- Initialize collectedItemSoundEffectPlayer and set as collect.mp3. - Initialize clickedSoundEffectPlayer and set as click.wav. - Initialize drawingSoundEffectPlayer and set as scribble.mp3.

	- setOnEndOfMedia, let the sound stop playing.
+ Getters / Setters	