# STICKY COOK

## Created by

Thanaphat Intarapuk 6631324221

Tatthon Limmonthon 6631325921

**2110215 Programming Methodology**

**Semester 2 Year 2023**

**Chulalongkorn University**

# STICKY COOK

---

## Introduction

Sticky cook is a cooking simulator game inspired by Overcook, in which The player will take on the role of a chef who must cook dishes to earn money. The objective of the game is for the player to earn as much money as possible within a set time limit by cooking dishes.

## Main Menu Page

# How to play


Player need to prepare meals (now we have only one meal which is hamburger) by combining various ingredients to cook meals and sell it to earn money.

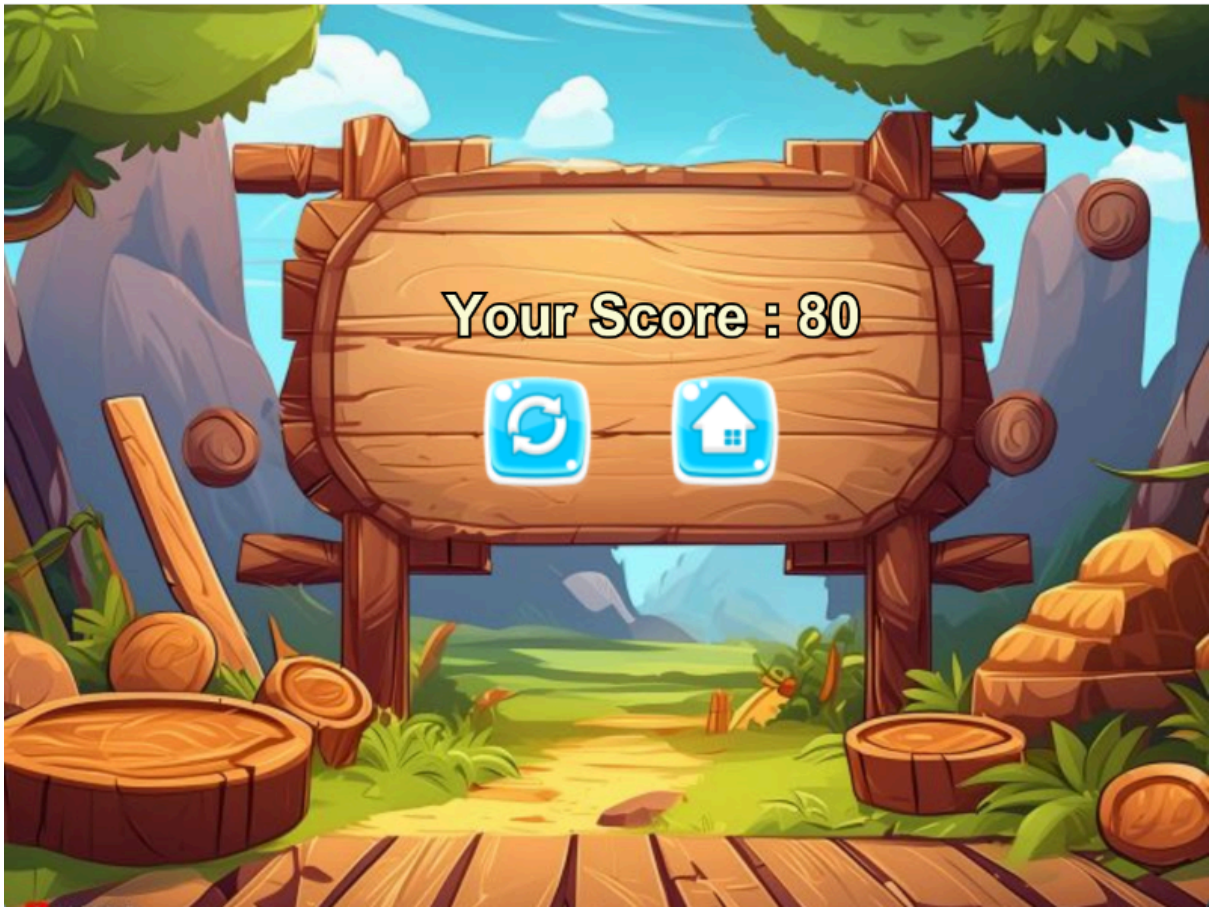


Each type of ingredient has a different preparation process before being combined into dishes."


After you combine all ingredient according to the recipes, you will get a meals that you can sell it to the boat on the right side of the map.

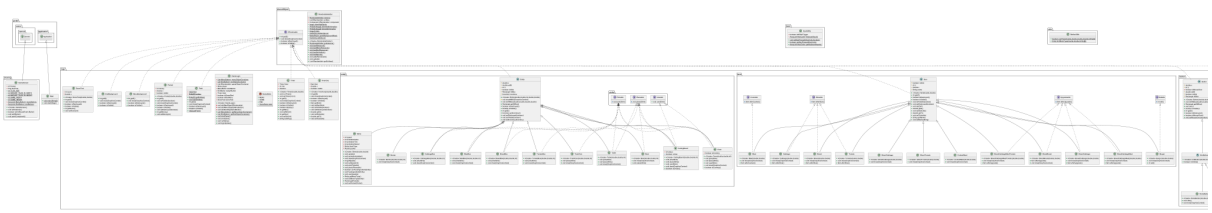

After selling a meal, you will earn money based on the type of meal. Your goal is to earn as much money as possible within the given time frame.

# End Page




After you finish the game, it will show the score for each round. you can restart the game by clicking the restart button and go to the main menu by clicking the home button.

# Class Diagram



# 1.Package logic

1.1 public class GameLogic

Field

| Name | Description |
|---|---|
| - List<MenuButton> menuObjectContainer | MenuButton holder |
| - List<EndButton> endObjectContainer | EndButton holder |
| - List<IRenderable> gameObjectContainer | IRenderable object holder |
| - Slime player | Game player |
| - MenuButton startButton | start button |
| - EndButton restartButton | restart button |
| - EndButton homeButton | home button |
| - Timer timer | timer |
| - boolean isGameStart | return whether game is start |
| - boolean isRunMenu | return whether game is in run menu |
| - ScoreText scoreText | player score |

Constructor

| Name | Description |
|---|---|
| + GameLogic() | Initiate all button,background,timer,scoretext and field |

Method

| Name | Description |
| --- | --- |
| - void addNewObject(IRenderable renderObject) | add IRenderable object to game |
| - void addMenuObject(MenuButton menuButton) | add menu object to game |
| - void addEndObject(EndButton endButton) | add end object to game |
| + List<MenuButton> getMenuObjectContainer() | return menuObjectContainer |
| + List<EndButton> getEndObjectContainer() | return endObjectContainer |
| + void startGame() | run game |
| + void endGame() | end game |
| + void runMenu() | run menu page |
| + void logicUpdate() | update game logic |

1.2 package entity

1.2.1 public abstract class Entity implement IRenderable

Field

| Name | Description |
| --- | --- |
| # double x,y | position in x-axis and y-axis of |

| | |
|---|---|
| | Entity |
| # double width,height | width and height of Entity |
| # boolean visible,destroyed | show that Entity is visible,destroy |
| # Rectangle hitBox | hitbox of Entity |
| # Inventory inventory | inventory of Entity |

Constructor

| Name | Description |
|---|---|
| + Entity(double x, double y, int z, double width, double height) | -set x,y,z,width,height<br>-set visible = true<br>-set destroyed = false<br>-setHitBox by x,y,widht,height<br>-initialize inventory for entity |

Method

| Name | Description |
|---|---|
| # void drawHitBox(GraphicsContext gc) | -set line width = 2.0<br>-set Fill color pink<br>-set StrokeRect by hitBox.getX(),hitBox.getY(),hitBox.getWidth(),hitBox.getHeight() |
| + void setHitbox(double x, double y, double width, double height) | initialize Rectangle hitbox by x,y,width,height |

| | |
|---|---|
| + boolean isDestroyed() | return destroyed |
| + boolean isVisible | if Gamestate is a GameState.GAME<br>return false<br>otherwise return visible |
| + int getZ() | return z |
| + setter/getter of all field | setter and getter of all field |

1.2.2 public class Slime extends Entity

Field

| Name | Description |
|---|---|
| - int speed | slime’s speed |
| - int animationIndex | animationIndex = 0 |
| - int animationTick | animationTick = 0 |
| - int animationSpeed | animationSpeed = 15 |
| - Entity blockType | type of block in front of slime |
| - Pocket pocket | slime’s pocket |

Constructor

| Name | Description |
|---|---|
| + Slime(double x, double y) | -set x,y<br>-set z = 100<br>-set width,height = 32,30<br>-set inventory x=9*32,y=13*32<br>-set inventory visible=true<br>-initialize pocket and set<br>-visible=true |

Method

| Name | Description |
| --- | --- |
| + void update() | -update position of slime<br>-update whether slime pick or place anything<br>-update whether slime use anything |
| + void positionUpdate() | update position of slime while clicking WASD according to speed of the slime by always checking whether slime can pass through a block type(entity) in front of it |
| + void pickAndPlaceUpdate() | update when slime picks or places items in any entity by always checking whether block type in front of the slime can pick or place. |
| + void useUpdate() | update when slime uses an entity by always checking whether block type in front of the slime can be used. |
| + void animationUpdate() | update animation of slime<br>(idle animation & walk animation) |
| + void draw(GraphicsContext gc) | -draw Slime at it current position(by animation of slime)<br>-draw inventory of slime |
| + void pick(Pickable entity) | Slime pick item form Pickable entity |
| + void place(Placeable entity) | Slime place item to Placeable entity |

| | |
|---|---|
| + boolean useFuseIngredient(Entity entity) | return whether slime can combined ingredient (follow the recipes) |
| + void fuseIngredient(Entity entity) | fuse item in inventory of entity in front of slime and item in inventory of slime |
| + void use(Useab entity) | Slime use Useable entity |
| + setter/getter of all field | setter and getter of all field |

1.2.3 public class TomatoBox extends Entity implements Pickable

Constructor

| Name | Description |
|---|---|
| + TomatoBox(double x, double y, int z) | set x,y,z<br>set width,height = 32,32<br>initialize item tomato<br>add tomato into this inventory |

Method

| Name | Description |
|---|---|
| + void pick(Slime user) | transfer Item inside of this inventory into Slime's inventory and add new tomato into this inventory |
| + void draw(GraphicsContext gc) | draw tomato box at x,y |

1.2.4 public class MeatBox extends Entity implements Pickable

Constructor

| Name | Description |
| --- | --- |
| + MeatBox(double x, double y, int z) | set x,y,z<br>set width,height = 32,32<br>initialize item meat<br>add meat into this inventory |

Method

| Name | Description |
| --- | --- |
| + void pick(Slime user) | transfer Item inside of this inventory into Slime’s inventory and add new meat into this inventory |
| + void draw(GraphicsContext gc) | draw meat box at x,y |

1.2.5 public class CabbageBox extends Entity implements Pickable

Constructor

| Name | Description |
| --- | --- |
| + CabbageBox(double x, double y, int z) | set x,y,z<br>set width,height = 32,32<br>initialize item cabbage<br>add cabbage into this inventory |

Method

| Name | Description |
| --- | --- |
| + void pick(Slime user) | transfer Item inside of this inventory into Slime’s inventory |

| | |
|---|---|
| | and add new cabbage into this inventory |
| + void draw(GraphicsContext gc) | draw cabbage box at x,y |

1.2.6 public class BreadBox extends Entity implements Pickable

Constructor

| Name | Description |
|---|---|
| +BreadBox(double x,double y,int z) | set x,y,z<br>set width,height = 32,32<br>initialize item bread<br>add bread into this inventory |

Method

| Name | Description |
|---|---|
| + void pick(Slime user) | transfer Item inside of this inventory into Slime's inventory and add new bread into this inventory |
| + void draw(GraphicsContext gc) | draw bread box at x,y |

1.2.7 public class TrashCan extends Entity implements Placeable

Constructor

| Name | Description |
|---|---|
| + TrashCan(double x, double y, int z) | set x,y,z<br>set width,height = 32,32 |

Method

| Name | Description |
|---|---|
| + void place(Slime user) | make item inside slime's inventory null |
| + void draw(GraphicsContext gc) | draw trashcan at x,y |

1.2.8 public class Table extends Entity implements Pickable,Placeable

Constructor

| Name | Description |
|---|---|
| + Table(double x, double y, int z) | set x,y,z<br>set width,height = 32,32 |

Method

| Name | Description |
|---|---|
| + void pick(Slime user) | transfer Item inside of this inventory into Slime's inventory |
| + void place(Slime user) | transfer Item inside of slime's inventory into this inventory |
| + void draw(GraphicsContext gc) | draw table at x,y |

1.2.9 public class Oven extends Entity implements Pickable,Placeable,useable

Field

| Name | Description |
|---|---|

| | |
|---|---|
| - boolean isCooking | whether oven is cooking |

Constructer

| Name | Description |
|---|---|
| + Oven(double x, double y, int z) | set x,y,z<br>set width,height = 32,32 |

Method

| Name | Description |
|---|---|
| + void pick(Slime user) | transfer Item inside of this inventory into Slime's inventory (cannot pick item while oven is cooking) |
| + void place(Slime user) | transfer Item inside of slime's inventory into this inventory (cannot put item into oven while oven is cooking and item that put in oven need to have interface Cookable) |
| + void use(Slime user) | after use oven, item that can cook will turn to cooked item<br>(cannot use while oven is cooking) |
| + void draw(GraphicsContext gc) | draw oven at x,y<br>if oven is cooking draw cooking oven |
| + boolean isCooking() | return isCooking |

1.2.10 public class CuttingBoard extends Entity implements Pickable,Placeable,useable

Field

| Name | Description |
|---|---|
| - boolean onUse | whether cutting board is using |
| - int status | status of cutting board that show the progress of cutting |

Constructor

| Name | Description |
|---|---|
| + CuttingBoard(double x, double y, int z) | set x,y,z<br>set width,height = 32,32 |

Method

| Name | Description |
|---|---|
| + void pick(Slime user) | transfer Item inside of this inventory into Slime’s inventory |
| + void place(Slime user) | transfer Item inside of slime’s inventory into this inventory |
| + void use(Slime user) | can use cutting board for item that implement sliceable only<br>after use cutting board, item that can slice will turn to sliced item |
| + void draw(GraphicsContext gc) | draw cutting board at x,y<br>draw state when use cutting board |
| + boolean isOnUse() | return onUse |

1.2.11 public class Boat extends Entity implements Placeable

Constructor

| Name | Description |
|---|---|
| + Boat(double x, double y, int z) | set x,y,z<br>set width,height = 64,64 |

Method

| Name | Description |
|---|---|
| + void place(Slime user) | if item implement sellable, sell item inside slime's inventory |
| + void draw(GraphicsContext gc) | draw boat at x,y |

1.2.12 public class Border extends Entity

Constructor

| Name | Description |
|---|---|
| + Border(double x, double y, int z) | set x,y,z<br>set width,height = 32,32<br>set visible = false |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw border at x,y |

1.2.13 package ability

1.2.13.1 public interface Pickable

Method

| Name | Description |
| --- | --- |
| # void pick(Slime user) | action of Entity when slime pick |

1.2.13.2 public interface Placeable

Method

| Name | Description |
| --- | --- |
| + void place(Slime user) | action of Entity when slime place |

1.2.13.3 public interface Useable

Method

| Name | Description |
| --- | --- |
| + void use(Slime user) | action of Entity when slime use |

1.3 package item

1.3.1 public abstract class Item implement IRenderable

Field

| Name | Description |
| --- | --- |
| # boolean visible,destroyed | show that Item is visible/destroyed |
| # int z | position on z-axis of item |
| # int x,y | position on x-axis and y-axis of item |
| # string name | name of item |

Constructor

| Name | Description |
| --- | --- |
| + Item(double x, double y) | set x,y<br>set z = 0 |

Method

| Name | Description |
| --- | --- |
| + boolean isDestroyed() | return destroyed |
| + boolean isVisible() | return visible |
| + int getZ() | return z |
| setter/getter of all field | setter and getter of all field |

1.3.2 public class Bread extends Item implement Sliceable

Constructor

| Name | Description |
| --- | --- |
| + Bread(double x, double y) | set x,y |

| | |
|---|---|
| | set name = “bread” |

Method

| Name | Description |
|---|---|
| + void draw(GraphicsContext gc) | draw bread at x,y |
| + Item afterSlice() | return SlicedBread position at x,y |

1.3.3 public class Cabbage extends Item implement Sliceable

Constructor

| Name | Description |
|---|---|
| + Cabbage(double x, double y) | set x,y<br>set name = “cabbage” |

Method

| Name | Description |
|---|---|
| + void draw(GraphicsContext gc) | draw cabbage at x,y |
| + Item afterSlice() | return SlicedCabbage position at x,y |

1.3.4 public class Tomato extends Item implement Sliceable

Constructor

| Name | Description |
| --- | --- |
| + Tomato(double x, double y) | set x,y<br>set name = “tomato” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw tomato at x,y |
| + Item afterSlice() | return SlicedTomato position at x,y |

1.3.5 public class Meat extends Item implement Cookable

Constructor

| Name | Description |
| --- | --- |
| + Meat(double x, double y) | set x,y<br>set name = “meat” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw cabbage at x,y |
| + Item afterCooked() | return CookedMeat position at x,y |

1.3.6 public class SlicedBread extends Item implement Upgradeable

Constructor

| Name | Description |
| --- | --- |
| + SlicedBread(double x, double y) | set x,y<br>set name = “sliced bread” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw sliced bread at x,y |
| + Item afterUpgrade() | return BreadCabbage position at x,y |

1.3.7 public class SlicedCabbage extends Item

Constructor

| Name | Description |
| --- | --- |
| + SlicedCabbage(double x, double y) | set x,y<br>set name = “sliced cabbage” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw sliced cabbage at x,y |

1.3.8 public class SlicedTomato extends Item

Constructor

| Name | Description |
| --- | --- |
| + SlicedTomato(double x, double y) | set x,y<br>set name = “sliced tomato” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw sliced tomato at x,y |

1.3.9 public class CookedMeat extends Item

Constructor

| Name | Description |
| --- | --- |
| + CookedMeat(double x, double y) | set x,y<br>set name = “cooked meat” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw cooked meat at x,y |

1.3.10 public class BreadCabbage extends Item implement Upgradeable

Constructor

| Name | Description |
| --- | --- |

| | |
|---|---|
| + BreadCabbage(double x, double y) | set x,y<br>set name = “bread with cabbage” |

Method

| Name | Description |
|---|---|
| + void draw(GraphicsContext gc) | draw bread with cabbage at x,y |
| + Item afterUpgrade() | return BreadCabbageMeat at position x,y |

1.3.11 public class BreadCabbageMeat extends Item implement Upgradeable

Constructor

| Name | Description |
|---|---|
| + BreadCabbageMeat(double x, double y) | set x,y<br>set name = “bread with cabbage with meat” |

Method

| Name | Description |
|---|---|
| + void draw(GraphicsContext gc) | draw bread with cabbage with meat at x,y |
| + Item afterUpgrade() | return BreadCabbageMeatTomato at position x,y |

1.3.12 public class BreadCabbageMeatTomato extends Item implement Upgradeable

Constructor

| Name | Description |
| --- | --- |
| + BreadCabbageMeatTomato(double x, double y) | set x,y<br>set name = “bread with cabbage with meat with tomato” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw bread with cabbage with meat with tomato at x,y |
| + Item afterUpgrade() | return Burger at position x,y |

1.3.13 public class Burger extends Item implement Sellable

Constructor

| Name | Description |
| --- | --- |
| + Burger(double x, double y) | set x,y<br>set name = “burger” |

Method

| Name | Description |
| --- | --- |
| + void draw(GraphicsContext gc) | draw burger at x,y |
| + int sell() | return value of burger (20) |

1.3.14 public Interface Cookable

Method

| Name | Description |
| --- | --- |
| + Item afterCooked() | return an item after cooked |

1.3.14 public Interface Sliceable

Method

| Name | Description |
| --- | --- |
| + Item afterSlice() | return an item after slice |

1.3.14 public Interface Sellable

Method

| Name | Description |
| --- | --- |
| + int sell() | return money according to the item you sell |

1.4 public class ScoreText implements IRenderable

Field

| Name | Description |
| --- | --- |
| - int score | Player Score |
| - double x | Position in x-axis |
| - double y | Position in y-axis |

Constructor

| Name | Description |
| --- | --- |
| + ScoreText(double x,double y) | Set x,y of ScoreText |

Method

| Name | Description |
| --- | --- |
| + int getZ() | return z |
| + void draw(GraphicsContext gc) | draw ScoreText |
| + boolean isDestroyed() | return destroyed status |
| + boolean isVisible() | return visible status |
| + int getScore() | return score |
| + int setScore(int score) | set score |

1.5 public class MenuBackGround implements IRenderable

Method

| Name | Description |
| --- | --- |
| + int getZ() | return z |
| + void draw(GraphicsContext gc) | draw MenuBackground |
| + boolean isDestroyed() | return destroyed status |
| + boolean isVisible() | return visible status |

1.6 public class EndBackground implements IRenderable

Method

| Name | Description |
|---|---|
| + int getZ() | return z |
| + void draw(GraphicsContext gc) | draw EndBackGround |
| + boolean isDestroyed() | return destroyed status |
| + boolean isVisible() | return visible status |

1.7 public class Inventory implements IRenderable

Field

| Name | Description |
|---|---|
| - Item item | item in inventory |
| - double x | Position in x-axis |
| - double y | Position in y-axis |
| - boolean visible | visible status |
| - boolean destroyed | destroyed status |

Constructor

| Name | Description |
|---|---|
| + Inventory(double x,double y) | Set x,y of Inventory |

Method

| Name | Description |
|---|---|

| | |
|---|---|
| + int getZ() | return z |
| + void draw(GraphicsContext gc) | draw Inventory with Item inside it |
| + boolean isDestroyed() | return destroyed status |
| + boolean isVisible() | return visible status |
| + Item getItem() | return item in inventory |
| + void setItem(Item item) | return item in inventory |
| + void setVisible(boolean visible) | set visible status |
| + void setDestroyed(boolean destroyed) | set destroyed status |
| + boolean isDestroyed() | return destroyed status |
| + boolean isVisible() | return visible status |
| + double getX() | return x-axis position x-axis |
| + double getY() | return y-axis position |
| + void setX(double x) | set x-axis position |
| + void setY(double y) | set y-axis position |

1.8 public enum GameState

This enum show game status.There are MENU,GAME,END

Field

| Name | Description |
|---|---|
| + GameState state | state of game |

1.9 public class Pocket implements IRenderable

Field

| Name | Description |
| --- | --- |
| - int money | money in pocket |
| - double x | Position in x-axis |
| - double y | Position in y-axis |
| - boolean visible | visible status |
| - boolean destroyed | destroyed status |

Constructor

| Name | Description |
| --- | --- |
| + Pocket(double x,double y) | Set x,y of Pocket |

Method

| Name | Description |
| --- | --- |
| + int getZ() | return z |
| + void draw(GraphicsContext gc) | draw pocket with money |
| + boolean isDestroyed() | return destroyed status |
| + boolean isVisible() | return visible status |
| + void setDestroyed(boolean destroyed) | set destroyed status |
| + int getMoney() | return money in pocket |
| + void setMoney(int money) | set money in pocket |

1.10 package button

1.10.1 public abstract class Button implements IRenderable

Field

| Name | Description |
|---|---|
| # double x | Position in x-axis |
| # double y | Position in y-axis |
| - int z | Position in z-axis |
| - boolean isMouseOver | return true if mouse over button else return false |
| # double height | height of button |
| # double width | width of button |
| # boolean visible | visible status |
| # boolean destroyed | destroyed status |
| # Rectangle hitbox | hitbox of button |

Constructor

| Name | Description |
|---|---|
| + Button(double x, double y,int z, double width, int height) | set x,y,z,width and height of button |

Method

| Name | Description |
|---|---|
| + void setHitBox(double x, double y, double width, double height) | initial hitbox |
| + Rectangle getHitbox() | return hitbox of button |
| + void click() | action of button when clicked |
| + boolean isVisible() | return visible status |
| + getZ() | return z |
| + boolean isDestroyed() | return destroyed status |
| + boolean isMouseOver() | return isMouseOver status |
| + void setMouseOver() | set isMouseOver status |

1.10.2 public abstract class Endbutton extends Button

Constructor

| Name | Description |
|---|---|
| + EndButton(double x, double y, int z, double width, int height) | set x,y,z,width and height of EndButton object |

Method

| Name | Description |
|---|---|

| | |
|---|---|
| + boolean isVisible() | return visible status for EndButton object |

1.10.3 public abstract class Menubutton extends Button

Constructor

| Name | Description |
|---|---|
| + MenuButton(double x, double y, int z, double width, int height) | set x,y,z,width and height of MenuButton object |

Method

| Name | Description |
|---|---|
| + boolean isVisible() | return visible status for MenuButton object |

1.10.4 public class HomeButton extends EndButton

Constructor

| Name | Description |
|---|---|
| + HomeButton(double x, double y, int z, double width, int height) | set x,y,z,width and height of HomeButton |

Method

| Name | Description |
|---|---|

| | |
|---|---|
| + void click() | go to menu page when button clicked |
| + void draw(GraphicsContext gc) | draw home button |

1.10.5 public class RestartButton extends EndButton

Constructor

| Name | Description |
|---|---|
| + RestartButton(double x, double y, int z, double width, int height) | set x,y,z,width and height of RestartButton |

Method

| Name | Description |
|---|---|
| + void click() | restart game when button clicked |
| + void draw(GraphicsContext gc) | draw restart button |

1.10.6 public class StartButton extends MenuButton

Constructor

| Name | Description |
|---|---|
| + StartButton(double x, double y, int z, double width, int height) | set x,y,z,width and height of StartButton |

Method

| Name | Description |
|---|---|

| | |
|---|---|
| + void click() | start game when button clicked |
| + void draw(GraphicsContext gc) | draw start button |

1.11 public class Timer implement IRenderable

Field

| Name | Description |
|---|---|
| - Timer timer | timer |
| - int min,sec | countdown timer in minute and second |
| - double x,y | position of timer at x-axis and y-axis |

Constructor

| Name | Description |
|---|---|
| + Timer(double x, double y) | set position of timer at x-axis and y-axis |

Method

| Name | Description |
|---|---|
| + void runTimer() | -Run the timer thread according to the specified time.<br>-when time runs out change game state to GameState.End |
| + void setTimer(int min, int sec) | set countdown timer |
| + void draw(GraphicsContext gc) | -set stroke black<br>-set font size 20 |

| | |
|---|---|
| | -draw stroke text with timer at position x,y |
| + int getZ() | return 10 |
| + boolean isDestroyed() | return false |
| + boolean isVisible() | visible only when game state is GameState.GAME |
| + boolean isEmpty() | return whether timer is does not have countdown time |
| + String toString() | return string of timer |
| setter/getter of all field | setter and getter of all field |

1.12 public class Field implement IRenderable

Field

| Name | Description |
|---|---|
| - int[][] field | -field of each tile in game screen<br>-you can put the specified number to create a specified entity in each tile |
| - Entity[][] entities | -field that Convert the number taken from the intreger field into an entity.<br>-size equal to int[][] field |

Method

| Name | Description |
|---|---|
| + Entity[][] getEntities | return entities field |

| | |
|---|---|
| + void loadEntities() | - Create an entities field by converting numbers from an integer field into entities.<br>-each entity have a specified number |
| + int getZ() | return -999 |
| + void draw(GraphicsContext gc) | draw background image of the field |
| + boolean isDestroyed() | return false |
| + boolean isVisible() | visible only when game state is GameState.GAME |
| setter/getter of all field | setter and getter of all field |

# 2.Package sharedObject

2.1 public interface IRenderable

Method

| Name | Description |
|---|---|
| + int getZ() | return z |
| + void draw(GraphicsContext gc) | draw various things on screen |
| + boolean isDestroyed() | return that whether things is destroyed |
| + boolean isVisible() | return that whether things is visible |

2.2 public class RenderableHolder

Field

| Name | Description |
|---|---|
| - RenderableHolder instance | instance of RendarableHolder |
| - List<IRendarable> entities | list of IRendareable entities |
| - Comparator<IRendarable> comparator | comparator |
| + Image<br><br>slimeWalkSprite,slimeIdleSprite,tomatoBox,table,tomato,fieldSprite,fieldSprite2,fieldSprite3,<br><br>UIsprite,BackgroundImage,inventoryBar,pocket,coin,trashcan,border,oven,usingOven,kitchenSprite,cuttingBoard,<br><br>slicedTomato,meatBox,meat,cookedMeat,cabbageBox,cabbage,slicedCabbage,breadBox,bread,slicedBread,burgerSprite,<br><br>breadWithCabbage,breadWithCabbageWithMeat,breadWithCabbageWithMeatWithTomato,burger,boat,menu,startButton,<br><br>startButtonHover,menuBackground,croppedMenuBackground,endBackground,restartButton,homeButton,croppedEndBackground; | all resource image |
| + WritableImage[] slimeWalkAnimation,slimeIdleAnimation | image list of slime animation |
| + Image[] state | image list of cutting state |
| + AudioClip pickItemSound | pick item sound |
| + MediaPlayer | background music of menu page |

| | |
|---|---|
| gameBackgroundMusic,menuBackgroundMusic | and game page |
| + AudioClip walkSound,chopSound,grillSound, doneCookingSound,sold, clockSound,endSound | various sound of the game |

Constructor

| Name | Description |
|---|---|
| + RenderableHolder() | initialize entities<br>make comparator compare according to z-axis |

Method

| Name | Description |
|---|---|
| + RenderableHolder getInstance() | return instance of this class |
| static block method | loadResource<br>loadAnimation<br>loadMenuResource<br>loadSound<br>loadEndResource |
| + void loadResource | load all image resource of the game |
| + void loadMenuResource | load all image resource of the menu |

| | |
|---|---|
| + void loadEndResource | load all image resource of the end scene |
| + void loadAnimation | load slime's animation(list of image) |
| + void loadSound | load all sound of the project |
| + void add(IRenderable entity) | add IRenderable object to entities and sort it according to comparator |
| + void update() | if object is destroyed delete it form entities |
| + List<IRenderable> getEntities | return entities |

# 3.Package drawing

3.1 public class GameScreen extends Canvas

Field

| Name | Description |
|---|---|
| + int TILES_SIZE | each tile of game (size = 32) |
| + int AMOUNT_TILES_IN_WIDTH | amount of tile in width (amount = 20) |
| + int AMOUNT_TILES_IN_HEIGHT | amount of tile in height (amount = 20) |
| + int GAME_WIDTH | TILES_SIZE * AMOUNT_OF_TILES_IN_WIDTH |

| | |
|---|---|
| + int GAME_HEIGHT | TILES_SIZE * AMOUNT_OF_TILES_IN_HEIGHT |
| - ArrayList<MenuButton> menuButtons | array list of menu button |
| - ArrayList<EndButton> endButtons | array list of end button |

Constructor

| Name | Description |
|---|---|
| + GameScreen() | set canvas with GAME_WIDTH and GAME_HEIGHT<br>add event listener<br>add button |

Method

| Name | Description |
|---|---|
| + void addListener() | add listener to GameScreen |
| + boolean isOn(MouseEvent e,Button button) | check if mouse is on button in GameScreen |
| + void addButton() | add button to GameScreen |
| + void paintComponent() | paint all component to GameScreen |

# 4.Package input

4.4 public class InputUtility

Field

| Name | Description |
|---|---|
| - boolean leftClickTrigger | status mouse’s left click trigger |
| - boolean leftClickPressed | status mouse’s left click pressed |
| *- Arraylist<KeyCode> keyboardInputs* | keyboard inputs holder |

Method

| Name | Description |
|---|---|
| + void setKeyPressed(KeyCode keycode, boolean pressed) | add keyboard input to keyboardInputs |
| + boolean getKeyPressed(KeyCode keyCode) | check if there is specific keyboard input in keyboardInputs |
| + ArrayList<KeyCode> getKeyboardInputs() | return keyboardInputs |

# 5.Package utils

5.1 public class MethodUtils

Method

| Name | Description |
| --- | --- |
| + boolean canPass(double x,double y) | check if coordinates x,y can pass or not |
| + Entity findBlockType(double x,double y) | return entity type at coordinates x,y |