

PuakOuan Documntation

Created by

Navanon Nekkhum 6531320621

Atsawin sungsuwan 6532201821

2110215 Programming Methodology

Semester 1 Year 2019

Chulalongkorn University

Ghost Buster

Introduction

Ghost Buster is inspired by the general 8-bit picture game and the Pac-Man game. The objective of the game is to collect all of the ghost to win the game.

Rules

The rule is simple, just like a Pac-Man game that requires collecting all the dots. However, in this game, we will switch to collecting the movable ghost characters instead. The control user uses in this game is W/S when scrolling the option and SPACE to confirm that option. To go back or toggle option screen while playing user can press ESC. Lastly, to move the character, the user uses WASD. After hitting the ghost, it will disappear. When there are no more ghosts left, you win!

Title screen



Gameplay

- Start scene



- Playing scene



- Ending scene



- Pausing/Option scene



- Info option scene



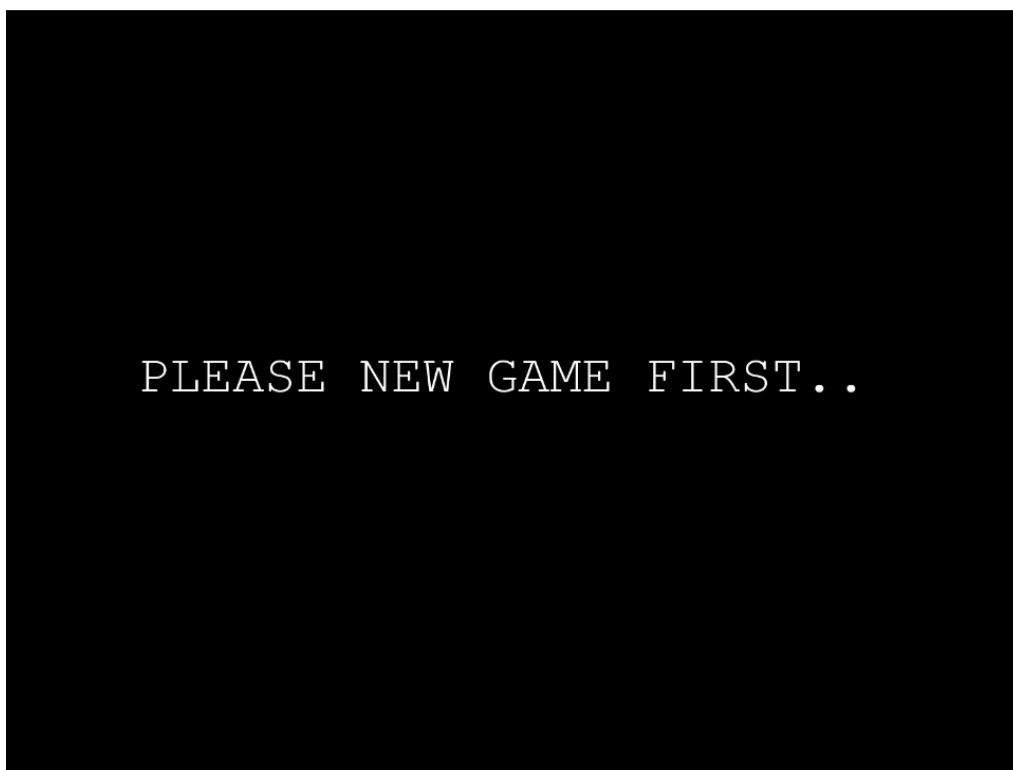
- Control option scene



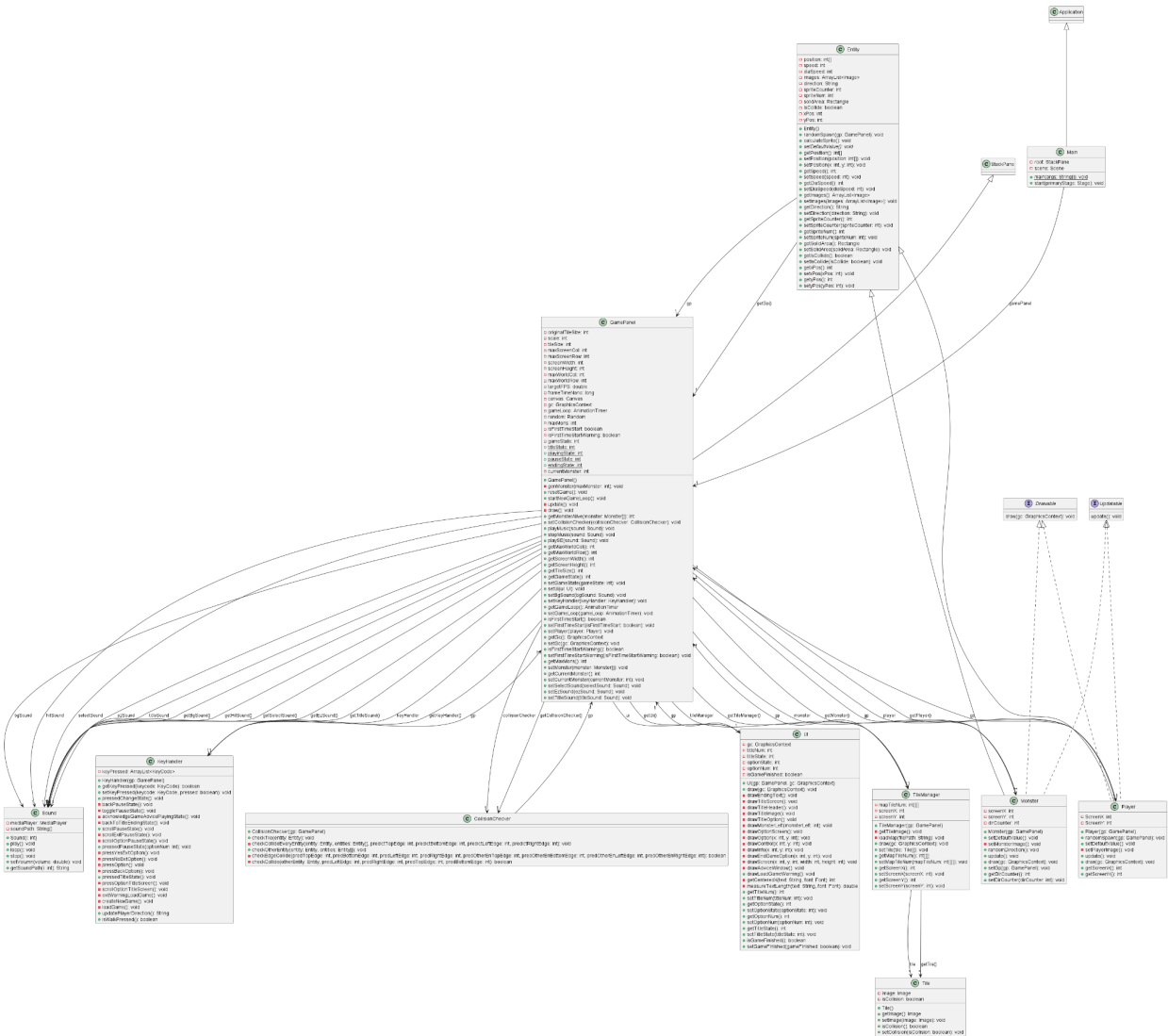
- Exit game option scene



- LoadGame before New Game scene



Class diagram



1.Package entity

1.1 interface Drawable

1.1.1 Methods

+void draw(GraphicsContext gc)	Set image and draw on the screen
--------------------------------	----------------------------------

1.2 interface Updatable

1.1.1 Methods

+void update()	Update variable to show on screen
----------------	-----------------------------------

1.3 Entity Class

1.3.1 Fields

- GamePanel gp	GamePanel
-int[] position	Array to contain the entity position in the game
-int speed	Number to represent speed of each entity
-int diaSpeed	Diagonal speed of each entity
-ArrayList<Image> images	Arraylist to contain picture of each entity
-String direction	String to contain the current direction of each entity
-int spriteCounter	Number to delay the change of spriteNum

-int spriteNum	Number to check what picture of the entity will be shown
-Rectangle solidArea	The shape that contains the hit box of the entity
-boolean isCollide = false	Boolean to check whether the entity is colliding with something or not
-int xPos	The number represent in-game tile number in X-axis
-int yPos	The number represent in-game tile number in Y-axis

1.3.2 Constructor

+Entity()	-Initialize all field -Set images to empty ArrayList<Image> -Set solidArea to new Rectangle() -Set position to empty int[2]
-----------	--

1.3.3 Methods

+randomSpawn(GamePanel gp)	Set entity to random spawn in map
+void calculateSprite()	Switch the value of spriteNum to swap the character picture
+abstract void setDefaultValue()	-
+setPosition(int[] position)	To assign the array to the field position

+setPosition(int x, int y)	To set position exact position in X-axis and Y-axis after creating the the position array
Generate getter/setter of fields	

1.4 Player class

This class extends from the entities and implements Drawable and Updatable.

1.4.1 Fields

-final GamePanel gp	GamePanel
-final int ScreenX	Position on the screen on the X-axis
-final int ScreenY	Position on the screen on the Y-axis

1.4.2 Constructor

-Player(GamePanel gp)	<ul style="list-style-type: none"> -Set all fields -Set ScreenX to the center of the X-axis of window -Set ScreenY to the center of the Y-axis of window -Set solidArea position to (10,18) -Set solidArea width and height to 28 -Set the setPlayerImage() -Set the setDefaultValue()
-----------------------	---

1.4.3 Methods

+void randomSpawn(GamePanel gp)	Random spawn player and check to not spawn on the tile that player will collide
+void setDefaultValue()	<ul style="list-style-type: none">-Randomspawn player-Set player position by xPos and yPos variable-Set player speed to 6-Set player diagonal speed to $6/\sqrt{2}$-set player direction to "down"
+void setPlayerImage()	Add the player images in the up, down, left ,right order
+void update()	<ul style="list-style-type: none">-Player will not be update until player acknowledge game advice-Update the player direction depend on the input keys-Set isCollide to 'false'-Check if it collides with another entity.-Check if it collides with tiles-Set new in-game position to player-Calculate the sprite of the player
+void draw(GraphicsContext gc)	-Choose image depend on the direction and spriteNum and draw it on the screen
+int getScreenX()	Return the ScreenX number
+int getScreenY()	Return the ScreenY number

1.5 Monster Class

This class extends from the entities and implements Drawable and Updatable.

1.5.1 Fields

-GamePanel gp	GamePanel
-int screenX	Position on the screen on the X-axis
-int screenY	Position on the screen on the Y-axis
-int dirCounter	Number to delay the change of direction of the monster

1.5.2 Constructor

+Monster(GamePanel gp)	<ul style="list-style-type: none">-Set all fields-Set default value-Set solidArea position to (10,18)-Set solidArea width and height to 28
------------------------	---

1.5.3 Methods

+void setDefaultValue()	<ul style="list-style-type: none">-Random spawn monster-Set monster position by xPos and yPos variable-Set monster speed to 4-Set monster diagonal speed to $4/\sqrt{2}$
-------------------------	--

	<ul style="list-style-type: none"> -set monster direction to "down" -set dirCounter to 59
-void setMonsterImage()	Add the monster images in the up, down, left ,right order
-void randomDirection()	<ul style="list-style-type: none"> -Random the direction of monster to walk by random 1-800 every time the dirCounter is divisible by 60 -if result in 1-100 monster will move up -if result in 101-200 monster will move down -if result in 201-300 monster will move left -if result in 301-400 monster will move right -if result in 401-500 monster will move right and up -if result in 501-600 monster will move right and down -if result in 601-700 monster will move left and up -if result in 701-800 monster will move left and down -If the dirCounter is not divisible by 60 increase the dirCounter value by 1
+void update()	<ul style="list-style-type: none"> -Random the direction -Set isCollide to true -Check if it is collide with player -Check if it is collide with tiles -Calculate the position on the screen of the monster -Calculate the sprite of the monster
+void draw(GraphicsContext gc)	<ul style="list-style-type: none"> -Choose image depend on the direction and spriteNum and draw it on the screen

+void setGp(GamePanel gp)	set gp
+int getDirCounter()	get dirCounter
+void setDirCounter(int dirCounter)	set dirCounter

2.Package main

2.1 class GamePanel

2.1.1 Fields

- final int originalTileSize	tile size from image = 16pixel
- final int scale	use for multiple originalTileSize to get bigger size = 3
- final int tileSize	real tileSize in game from $scale * originalTileSize$
- final int maxScreenRow	set MaxScreenRow = 16 ; number of tileSize in y axis (screen height in tileSize)
- final in maxScreenCol	set MaxScreenRow = 12 ; number of tileSize in x axis (screen width in tileSize)
- final int screenWidth	screen Width in pixel = $MaxScreenRow * tileSize$
- final int screenHeight	screen Height in pixel = $MaxScreenCol * tileSize$
- final int maxWorldCol	max column of the map

- final int maxWorldRow	max row of the map
- final double targetFPS	target FPS
- final long frameTimeNano	time to draw 1 frame
- Canvas canvas	canvas
- GraphicsContext gc	GraphicsContent
- AnimationTimer gameLoop	game loop
- Sound bgSound	background song in game
- Sound hitSound	hit ghost sound effect
- Sound selectSound	scrolling option sound effect
- Sound ezSound	end game sound effect
- Sound titleSound	background song in title screen
- KeyHandler keyHandler	handling key press
- CollisionChecker collisionChecker	for check collide of entity
- Random random	for create random number in gamePanel use to random number of monster in game
- int maxMons	max number of monster each game
- UI ui	user interface
- boolean FirstTimeStart	variable to check if user just NEW GAME = true
- boolean FirstTimeStartWarning	variable to check if user LOAD GAME before NEW GAME = false this variable depend on FirstTimeStart
- int gameState	state of the game represent what user seeing = 0

- static final int titleState	gameState = 0
- static final int playingState	gameState = 1
- static final int pauseState	gameState = 2
- static final int endingState	gameState = 3
- int currentMonster	number of monsters use with method getMonsterAlive to know if number of monsters change
- Monster monster[]	monsters
- TileManager tileManager	Manage Tile, load map
- Player player	player that user control

2.1.2 Constructor

+GamePanel()	<ul style="list-style-type: none"> -Initialize canvas with size equals to window screen -Set background to black -Set hitSound volumn to 0.2 -Set bgSound volumn to 0.2 -Set titleSound volumn to 0.2 -Set ezSound volumn to 0.7 -Set frameTimeNano to 1000000000 / targetFPS -Set the keypress that is used in game -Set option screen to toggle -Set title Screen to titleState -Set option screen to pauseState -Start animation timer
--------------	---

2.1.3 Methods

-void genMonster(int maxMonster)	Generate the monster and the maximum monster is 10
+void resetGame()	<ul style="list-style-type: none"> -Initialize new player -Random maxMonster between 5-9 -Generate monster -Set option state to 0 -Set option num to 1 -Set title num to 0 -Set title state to 0 -Set game state to titleState -Play titleSound
+void startNewGameLoop()	<ul style="list-style-type: none"> -ResetGame no matter it gameloop is running or not -Create new animation timer and update and draw -Start game loop
+void update()	<ul style="list-style-type: none"> -If the game state is pause state do nothing -If the game is playing state <ul style="list-style-type: none"> -update player -if the monster is not null update the monster -if monster alive is not equal to current monster play hitsound and set current monster to monster alive -if monster alive is 0 set the setGameFinished to true, stop background sound, play ezSound and set game state to ending state
+void draw()	<ul style="list-style-type: none"> - Clear the window screen -If the game is titleState and user load game before playing the game it will show warning otherwise it will show the title screen

	-If the gameState is not titleState it will generate map, player and if the monster is not null it will be generated too -Draw how many monster is left -Draw option screen if it is pause state -Draw ending screen if it is ending state
+int getMonsterAlive()	Count how many monster is alive
+void playMusic(Sound sound)	play and loop the music sound
+void stopMusic	stop the music sound
+void playSE(Sound sound)	play the music sound only once
Generate getter/setter of fields	

2.2 class KeyHandler

2.2.1 Fields

- ArrayList<KeyCode> keyPressed	ArrayList to contain key that user press
- GamePanel gp	GamePanel

2.2.2 Constructor

+ KeyHandler(GamePanel gp)	initialize fields
----------------------------	-------------------

2.2.3 Methods

+ void setKeyPressed(KeyCode keycode,boolean pressed)	Add key Press to array list for use later when user press and remove key from array list when user release that key.
boolean getKeyPressed(KeyCode keycode)	return if arraylist contain keycode from parameter
+ void pressedChangeState()	contain method that use to change state of the game -use to close advice window after New game -close ending scene and back to title screen -toggle option screen and pause game -back to the option screen when user inside other option method contains : -acknowledgeGameAdvicePlayingState -backToTitleEndingState -togglePauseState -backPauseState
- void backPauseState()	press ESC to back to option screen when user is inside other option
- void togglePauseState	press ESC to toggle option screen and pause the game
- void acknowledgeGameAdvicePlayingState()	press SPACE to close game advice window and play game
- void backToTitleEndingState()	press ESC to go back to title screen after ending game

+ void scrollPauseState()	contain methods that use to choose option in PausingState
- void scrollExitPauseState()	press W/S to choose option in EXIT GAME screen , move cursor
- void scrollOptionPauseState()	press W/S to choose option in option screen , move cursor
+ void pressedPauseState()	contain method to select option that cursor is at
- void pressYesExitOption()	select option YES in EXIT GAME option
- void pressNoExitOption()	select option NO in EXIT GAME option
- void pressBackOption()	select BACK option to close option screen
- void pressOption()	select other option
+ void pressedTitleState()	contain methods about pressing in TitleState
- void pressOptionTitleScreen()	contain methods depend on what option user confirm <ul style="list-style-type: none"> - NEW GAME : call createNewGame method - LOAD GAME : call loadGame method - EXIT : exit application
- void scrollOptionTitleScreen()	press W/S to choose option in title screen , move cursor
- void exitWarningLoadGame()	press ESC to back to title screen when you in warning screen after press LOAD GAME when never NEW GAME
- void createNewGame()	create new game

- void loadGame()	if user already new game :load game if not bring user to Warning Screen
+ String updatePlayerDirection()	decide which direction player is move from the key user press return direction in String
+ boolean isWalkPress()	return if user press a key that relate to moving control (WASD)

2.3 class UI

2.3.1 Fields

- GamePanel gp	GamePanel
- GraphicsContext gc	GraphicsContext
- int titleNum	number represent where cursor is in title Screen 0:NEW GAME (default) 1:LOAD GAME 2:EXIT
- int titleState	number represent state in Title Screen 0:normal Title Screen (default) 1:Warning Screen when press LOAD GAME before NEW GAME
- int state	number represent state in Option screen 0:option screen (default state) 1:in INFO option 2:in CONTROL option 3:in EXIT GAME option
- int optionNum	number represent where cursor is in Option Screen

	1:INFO option (default state) 2:CONTROL option 3:EXIT GAME option 4:BACK option
- boolean isGameFinished	boolean to check if game finished

2.3.2 Constructor

+ UI(GamePanel gp,GraphicsContext gc)	initialize fields to default and from parameter
---------------------------------------	---

2.3.3 Methods

+ void draw(GraphicsContext gc)	draw UI depend on GameState endingState : call drawEndingText method titleState : call drawTitleScreen method pauseState : call drawOptionScreen method playingState : call drawMonsterLeft method if user just NEW GAME call drawAdviceWindow method
- void drawEndingText()	draw text “congratulation” and text “press esc to go back to title screen..”
- void drawTitleScreen()	draw title Screen -game name : call drawTitleHeader method -game logo : call drawTitleImage method

	-title option : call drawTitleOption method
- void drawTitleHeader()	draw game name : the ghost buster
- void drawTitleImage()	draw image of player with monster lean next to him
- void drawTitleOption()	draw Title Screen options -NEW GAME -LOAD GAME -EXIT
- void drawMonsterLeft(int monsterLeft)	draw monster left in the map
- void drawOptionScreen()	draw option screen by call DrawScreen method and this Option Screen content show depend on its state 0:call drawOption method 1:call drawInfo method 2:call drawControl method 3:call drawEndGameOption method
- void drawScreen(int x,int y,int width,int height)	draw round rectangle screen with 80% transparent black color and white round border parameter x,y is position x,y to draw this screen. width,height is size of this screen
- void void drawOption (int x,int y)	draw all options and header also draw cursor in front of option user choosing Header -OPTION Options -INFO -CONTROL -END GAME

	-BACK parameter x,y are position x,y to draw relate from Option Screen that draw before
- void drawControl(int x,int y)	draw text to display control use in this game which are -WASD for moving -ESC for back/toggle option -W/S for scrolling choosing option -SPACE for select/confirm that option parameter x,y are position x,y to draw relate from Option Screen
- void drawInfo(int x,int y)	draw info : the project's creators ,bless user for enjoying game
- void drawEndGameOption(int x,int y)	draw texts to ask user again for confirm to exit and save game,a warning that game won't save if user didn't accept game advice and exit draw option YES/NO to confirm and draw cursor in front of this option if player choosing it
- void drawAdviceWindow()	call drawScreen method and draw text of game's story and advice how to play this game and text to tell user press SPACE to continue to game
+ void drawLoadGameWarning()	draw Text "PLEASE NEW GAME FIRST.." if user press LOAD GAME before NEW GAME
- int getCenteredX(String text,Font font)	return centerX which is position x that if use to draw this text it's going to be center of the screen parameter text is text that going to

	draw and Font is font of this text
- double measureTextLength(String text,Font font)	return text length use in getCenteredX method parameter text is text that going to draw and Font is font of this text
Generate getter/setter of fields	no need getter/setter for GamePanel and GraphicsContent

2.4 class Sound

2.4.1 Fields

- MediaPlayer mediaPlayer	mediaPlayer to play sound
- String soundPath[]	array to contain path of songs Dict for using sound **NOTE SE means Sound Effect 0 : BG Song 1 : HitMonster SE 2 : wining SE 3 : select option using W,S SE 4 : wining SE2 5 : title BG Song

2.4.2 Constructor

+ Sound(int i)	parameter i is a number of index of the sound to create mediaPlayer which play that song
----------------	--

2.4.3 Methods

+ void play()	play the song/sound
+ void loop()	loop the song/sound
+ void stop()	stop the song/sound
+ setVolumn(double volume)	set song/sound volume by parameter parameter volume range is 0-1 represent % such as 0.6 mean 60%
+ String getSoundPath(int i)	getSoundPath from parameter i represent index of SoundPath array

2.5 class CollisionChecker

2.5.1 Field

-GamePanel gp	game panel
---------------	------------

2.5.2 Constructor

-CollisionChecker(GamePanel gp)	initialize fields to default and from parameter
---------------------------------	---

2.5.3 Methods

+void checkOtherEntity(Entity entity, Entity entities[])	Check whether the entity is collide with every other entity
-boolean checkEdgeCollide(int predTopEdge, int predBottomEdge, int predLeftEdge, int predRightEdge, int predOtherEnTopEdge, int predOtherEnBottomEdge, int predOtherEnLeftEdge, int predOtherEnRightEdge)	Check if the edges of both entities are colliding.
-boolean checkCollide(Entity otherEntity, int predLeftEdge, int predRightEdge, int predTopEdge, int predBottomEdge)	Check if both entities are colliding.
-void checkCollideEveryEntity(Entity entity, Entity entities[], int predictTopEdge, int predictBottomEdge, int predictLeftEdge, int predictRightEdge)	Set the value of every other entity that collides to null.
+void checkTile(Entity entity)	Check whether entity can move through the tile.
Generate getter/setter of fields	no need getter/setter for GamePanel

	and GraphicsContent
--	---------------------

2.6 class Main

2.6.2 Fields

- GamePanel gamePanel	gamePanel
- StackPane root	root pane
- Scence scene	scene

2.6.2 Methods

+ void start(Stage primaryStage)	initialize field
+ static void main(String[] args)	main application

3. Package tile

3.1 class Tile

3.1.1 Fields

- Image image	image of tile
- boolean isCollision	variable to tell if this tile can walk through or not

3.1.2 Constructor

+ Tile()	set Collision to false set image to null
----------	---

3.1.3 Methods

generate getter/setter for all fields	
---------------------------------------	--

3.2 class TileManager

3.2.1 Fields

- GamePanel gp	GamePanel
- Tile[] tile	array of tiles
- int[][] mapTileNum	2D array represents number map from text file
- screenX	position x in user Screen
- screenY	position y in user Screen

3.2.2 Constructor

+ TileManager(GamePanel gp)	-call getTileImage to initialize Tile array Images -initialize mapTileNum array dimension with MaxWorldCol,MaxWorldRow in GamePanel -call loadMap to initialize mapTileNum value initialize the left field (gp)
-----------------------------	--

3.2.3 Methods

- void getTileImage()	initialize Tile images from files in res package with each index represent each kind of tiles List index of tile 0 : normal ground 1 : tree1 2 : tree2 3 : mossy ground 4 : foot path 5 : foot path2 6 : corrupt area (didn't use) 7 : half foot path (didn't use) 8 : mystery crystal in the middle of crater 9 : crater ground
- void loadMap(String filePath)	load map from parameter file path loop get line of text in file and split

	them with “ “ get the value for initialize mapTileNum array
+ void draw(GraphicsContext gc)	loop in mapTileNum draw Tile in Screen (only draw in player POV to decrease runtime)
generate getter/setter for fields	no need for GamePanel