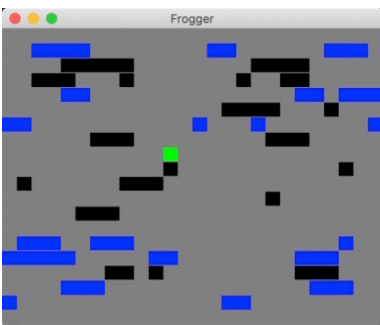
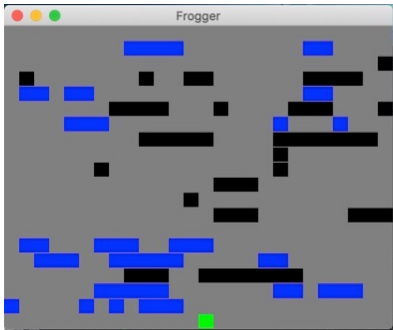
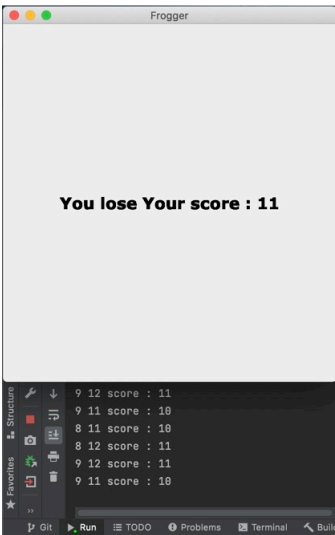
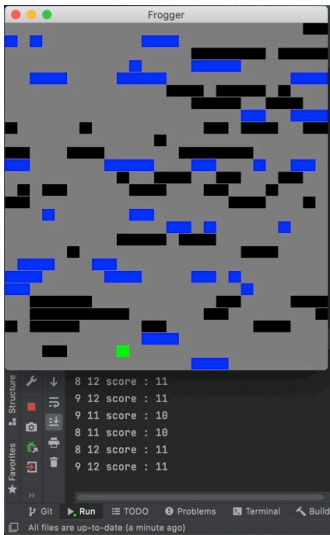


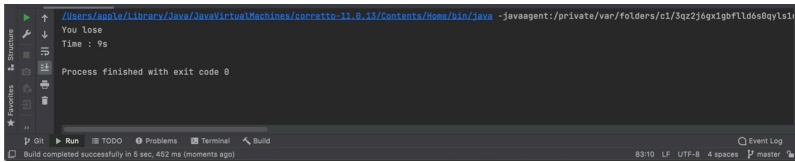
Nom: Yeqian He et Jincheng Ye



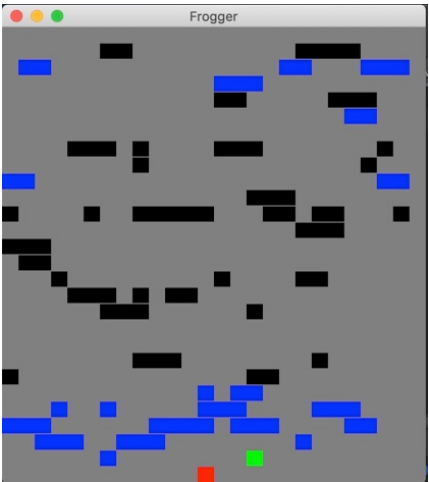
Environnement



Game infini



Temps affiche



Deux joueurs

Liste de
fonctions
finis:

Partie 1:

Frog.java : void move(final Direction key)

utilise le clavier de déplacer le grenouille

Game.java : boolean testLose()

Si le grenouille déplacer sur les voitures, il échoue

boolean testWin()

Si le grenouille gagne à déplacer à la dernier ligne, il gagne

Partie 2: (package partie2)

Car2.java : Constructeur(Game2 game, Case leftPosition, boolean leftToRight)

Créer une nouvelle voiture

void moveCar(boolean a)

Si "a = true" cette voiture déplace une fois(Case est décide par Lane), si "a = false" ne déplace pas

boolean correct()

Détermine la position de voiture est bien dans le Game, si "leftPosition.absc < game.width" ou "leftPosition.absc + this.length > 0" return true

boolean isSafe(Case c)

Détermine s'il y a une voiture dans ce Case, si c est dans le cadre de voiture
return false

Lane2.java : Constructeur(Game2 game, int ord, double density)

Créer une nouvelle ligne pour déplacer les voitures

void update()

D'après le speed et puis décide si les voitures bougent dans ce "tic d'horloge"

void move(boolean b)

Si "a = true" les voitures dans "cars" déplace une fois(la destination de Case est décide par Lane), si "a = false" ne déplace pas

void remove()

Supprimer les voitures qui ne sont pas correctes dans la list

boolean isSafe(Case c)

Détermine s'il y a une voiture dans ce Case, si c est dans le cadre d'une voitures return false

Environment2.java : Constructeur(Game2 game)

Créer une nouveau environnement pour déplacer les lignes, le density de la première ligne et la dernière ligne est égal à 0

boolean isSafe(Case c)

Idem que Lane2.isSafe(c)

boolean isWinningPosition(Case c)

Définit que les Cases de la dernière ligne sont les positions gagnées

void update()

Mise à jour tous les lignes dans chacun "tic d'horloge"

Partie 3 : (package infinie et gameCommons)

IfCar.java et IfLane.java sont identique qu'avant

IfFrog.java : void move(final Direction key)

Ajouter les nouvelles instruction : quand le grenouille déplace "up", on doit ajouter une ligne suivie et le score "+1" si le score maintenant est plus que le Maxscore, Maxscore change;

et plus, si déplace "down" le score "-1", Maxscore ne change pas

IfEnvironment.java : Constructeur(Game2 game)

le density de les deux lignes avant est 0 maintenant

boolean isWinningPosition(Case c)

Il est jeu infini maintenant, donc il ne gagne pas du tout, toujours return false

void addLane()

Ajoute une nouvelle ligne dans l'environnement

DEnvironment.java : Ajoute un methode nouveau "addLane()"

Game.java : Contructeur(IFroggerGraphics graphic, int width, int height, int minSpeedInTimerLoop, double defaultDensity)

Ajoute le score et le scoreMax dans le game

boolean testLose()

Affiche le scoreMax dans le EndScreen

void update()

Mettre le grenouille initial à deuxième ligne, pour le joueur peut reculer et regarder bien les voitures dans la ligne avant

void addL()

Ajoute la nouvelle ligne dans le jeu

Partie 4.1 : (package temps)

Game.java : Constructeur(...)

Ajoute l'attribut "time" calcul le temps de construire le jeu

void boolean testLose()

Teste si le jeu échoue, calcule la différence entre le temps final et le temps construit, affiche dans le terminal

void boolean testWin()

Teste si le jeu gagné, la suite même que "testLose()"

Partie 5.1 : (package twoplayer)

FroggerGraphic2.java : Contructeur(...)

Ajoute le graphique du deuxième grenouille

void keyPressed(KeyEvent e)

Ajoute les instructions du deuxième grenouill par "AWS" sur le clavier d'anglais

(Modifie l'interface IFroggerGraphics2 en même temps)

Game3.java : void setFrog(IFrog frog1, IFrog frog2)

Set les deux grenouille dans le jeu

boolean testLose()

Si un grenouille échoue, le jeu finit

boolean testWin()

Si les deux grenouille gagnent, le jeu gagne

void update()

Mettre un grenouille vert, un grenouille rouge

(Change les type des attributs dans Car3.java, Lane3.java et Environment3.java)

aussi)

Problemes:

1.Quand on a exécute "main" de Partie2, il n'affiche pas les voitures

Solution:N'oublie pas d'ajoute le methode "addToGraphics()" dans le methode de bouger les voitures

2.Le graphique de Partie3 ne bouge pas

Solution:Réécrit le constructeur de IfCar.java

3.Dans Partie5.1, le deuxième grenouill ne bouge jamais

Solution:Ajoute "break" dans tous les case sauf le dernier

Difficultés:

Si j'ai essayé d'affiche le temps d'exécution de jeu dans le EndScreen, il toujours change et sans arrêt d'affiche dans la même position

Et plus, on a essayé de faire "branche", mais si je change les contenus dans une partie, toute les partie changent en même temps, donc on a créé les packages pour remplacer les branches.