**Design document**

**Kazak and Alien, short interactive story**

**By Oleksandr Bondarev**

Kazak and Alien is a short interactive story. Application will start with short instructions screen. In next screen a picture of Kazak and nature will appear with a small UFO in left top corner. User will have to control it and land on hill on the right. If landed on a hill – an alien will move towards Kazak. After that another instructions menu will appear and user will have to solve small logic game. If he succeeds – Alien will fly away.

**Design**

**Main**

Contain global variables. Setup screen and call WorldModel class. Display WorldModel. Handle key and mouse events and send them to WorldModel class.

Properties

* Main colors
* Global Boolean variables
* Global coordinates
* Variable for WorldModel

Methods

setup()

* Initialize WorldModel class
* Setup screen size and background.

draw() - display WorldModel class.

keyPressed()

* handle keyboard arrow keys.
* handle space button.
* handle “P” (pause) and “L” (show where to land UFO) buttons.

keyReleased() - set all Boolean variables that store information about keys from “keyPressed()” to “false”.

mousePressed() - handle mouse events.

**WorldModel**

Contain information about current act number (information about acts can be seen in Initial Proposal in the end of the document). Depending on current act - call different classes, display them and send mouse and key events to appropriate classes

Properties

* current Act
* variables for the following classes: Instructions, Background, Alien, Foreground, LandingArea, UFO, Kazak, Tree, LogicGame

Methods

display() – display classes from current act

handleKeyPressed() – handle keyPressed() from main class and send to classes depending on current act.

handleKeyReleased() - handle keyRelesed() from main class and send to classes depending on current act.

handleMousePressed() - handle mousePressed() from main class and send to classes depending on current act.

getCurrentAct() – get current act number.

nextAct() – switch current act to next act.

**Instructions**

Show instructions with all the text of the same style

Properties

* button style properties
* texts for different instructions
* variables for WorldModel and LogicGame classes

Methods

showTextWindow(float x, float y) – shows text window in given coordinates.

showText(float x, float y, float width, float height, String text) – show given text in given coordinates, of given size, with one preset style.

displayInstructions() – display instructions depending on current act.

displayAct1(), displayAct5() – display instructions for appropriate acts.

continueBtn(float x, float y) – display continue button in given coordinates.

handleMousePressed() – handle mouse event from WorldModel class.

isContinuePressed() – return true if mouse was pressed on continue button, else – false.

**Background**

Draw sky, background hills and river

Properties

* main colors
* variable for Tree class

Methods

display() – draw a sky and nature.

drawSky() – self explaining.

drawNature() – draw hills and river.

**Tree**

Display a tree in given coordinates

Properties

* main colors
* coordinates

Methods

display(float x, float y) – display tree in given coordinates

**Alien**

Draw an alien of different size, depending on act

Properties

* main colors
* time variables
* alien coordinates and velocity

Methods

reset() – reset coordinates to starting point.

inBackground(),inForeground() – move small and later big alien first in background, later in foreground (size changes when he approaches).

moveAlienToCoordinate(float x, float y) – smoothly move alien to given coordinates.

moveToCoordinate(float cur, float tgt) – smoothly changes current coordinate to target coordinate.

smallAlien(float x, float y), bigAlien(float x, float y) – draw small or big alien in given coordinates

setStatus() – receive landing status from UFO

**Foreground**

Show foreground image with Kazak and big Hill

Properties

* main colors
* variable for Kazak and Alien

Methods

display() – display front hill and Kazak and, after some point, an Alien

drawFrontHill() – self explaining

**Kazak**

Draw Kazak in red suite.

Properties – main colors

Methods

display() – display Kazak

**LandingArea**

Highlight all landing places. Correct place is highlighted in blue

Properties

* main colors
* Boolean variables for highlight status

Methods

display() – display landing field if “L” button was pressed

hide() – set status to switched off

opposite() – change status to opposite

**UFO**

Draw UFO. In act #2 it can be controlled by arrow keys on keyboard and mouse clicks

Properties

* main colors
* Boolean variables for UFO’s status
* messages for different landing areas
* variables for calculating OFO’s movement
* variable for alien class

Methods

display() – display UFO

init() – initial setup of UFO

reset() – set position to start position

moveUFO() – calculate next coordinates of UFO, based on current position and player controls

setLandingStatus() – provide Alien object with information about landing

pausedOpposite() – pause and un-pause flight

handleKeyPressed() – handle keyPressed() from WorldModel class

handleKeyReleased() - handle keyRelesed() from WorldModel class

handleMousePressed() - handle mousePressed() from WorldModel class

**LogicGame**

Play small logic game, using a mouse

Properties

* variables for game logic
* variables with button styles
* variable for WorldModel class

Methods

play() – start playing the game

makeGuess() – control pressed buttons and update game memory with chosen answer

displayUpBtn(float x, float y), displayDownBtn(float x, float y), displayConfirmBtn(float x, float y) – display Up, Down and confirm buttons in given coordinates

btnUp(), btnDown() – display square with arrow

guessBox(int x, int y, int val) – display square in given coordinates with chosen value

changeFillColor(float x, float y), changeCircleFillColor(float x, float y) – change buttons color based on given coordinates and mouse coordinates

givenBtnIsSelected(float x, float y), confirmIsSelected(float x, float y) – return true if buttons in given coordinated intersect with mouse coordinates

reset(WorldModel model) – reset game and ensure that it has reference to WorldModel

generateRandomCorrectVariant() – self explaining

guessToZero() – set all values of guess array to 0

setHistory() – save guess to array with history

handleMousePressed() - handle mousePressed() from WorldModel class

setGuess() – update guess value based on mouse coordiantes

confirmGuess() – check current guess after confirm button was pressed

processGuess() – process guess values

processWinScenario(), processLooseScenario() – finish game with one of the scenarios

noFirstTime() – is needed to prevent winning from the first try

increaseDifficulty(), decreaseDifficulty() – change game difficulty

**Initial proposal**

**Concept**: This is going to be a short interaction story-game about a Kazak and an alien. It will be divided in several acts.

**Act#1**, “Start screen”: User will see some text with short story, objective and control keys that are needed to fly UFO.

**Act#2**, “Landing UFO”: Some nature is shown. A Kazak in red suite is drawn in the left bottom corner of the screen. UFO is drawn in the left top corner of the screen and starts to fall down. User will control UFO. User has to land it according to an objective (far right hill). He or she will use controls described in act#1. If objective is complete small alien appears from the UFO and starts moving towards Kazak.

**Act#3**, “Meeting”: Alien comes to Kazak and appears to be bigger then expected (he was small, because he was far).

**Act#4**, “Greeting”: Alien and Kazak have a small conversation. Dialog clouds will appear next to each other and will be shown until user clicks button “Next”. Alien asks for help. He run out of gas.

**Act#5**, “Instructions”: User will see instructions for small game and its rules.

**Act#6**, “Guess ingredients”: Small logic game where user has limited tries to guess which ingredients are used for gas for UFO. Every game will need random ingredients. There will be hints: how many ingredients were guessed, but user will have to guess which are correct. It is impossible to guess in first turn.

**Act#7**, “Finish”: After guessing all ingredients alien flies away.

**Interaction**: There will be arrow keys and mouse clicks for UFO controlling. Clicking mouse on screen will also control everything else. There will be different buttons, depending on act number.

**Objectives**: Help an alien to get gas and fly away in space. Learn how to use arrays, classes and complex logic of switching between acts.