
Bilkent University

Department of Computer Engineering

# CS319 – Object Oriented Software Engineering Project

*Project short-name: Donkey Kong Game*

## Final Report (Final Draft)

Group 3E
Fuad Ahmadov
Çağatay Küpeli
Sine Mete
Arkın Yılmaz

Supervisor: Bora Güngören

Final Report (Final Draft)
Dec 16, 2017

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS319/3.

**Contents**

# Final Report (Final Draft)

*Project short-name: Donkey Kong Game*

## 1. Modifications in the Implementation

### 1.1 Model Subsystem

Model Subsystem includes Direction, Enemy, EnemyType, ExtraLife, FallingBarrel, FireElemental, GameEngine, Girl, Hammer, Ladder, Monkey, Movable, MyObject, Nonmovable, Oil, Platform, Player, RollingBarrel classes.

GameEngine was not under this subsystem in Analysis Report; however, it is moved under Model Subsystem during Design Report.

Direction, EnemyType, ExtraLife, Hammer, Ladder, Movable, MyObject, Nonmovable and Platform classes remained same.

Enemy class is remained relatively same only exception is a new variable called scoreable is added to prevent player to get more points from that particular enemy.

FallingBarrel, FireElemental, RollingBarrel classes are remained relatively same except for the fact that animation() method.

GameEngine is the core class that construct the state of the level and handle the collisions. New methods are added and some of the old methods declarations changed.

**New Variables**

- BARREL_SPAWN_TIMER is a new constant integer that is used as a criteria for creating a new barrel.
- UPDATE_TIMER is a new constant integer that is used for updating Girl's and Oil's animations.
- MONKEY_ANIMATION_NUMBER is a new constant integer that is used for updating Monkey's animations.
- HAMMER_TIMER is a new constant integer that is used for updating Player's hammer animation.
- jump is a new Boolean variable that checks if the Player is jumping or not.
- barrels is a new ArrayList<Barrel> that stores all the barrel objects for collision checks.
- fireElementals is a new ArrayList<FireElemental> that stores all the fire elemental objects for collision checks.

- oils is a new ArrayList<Oil> that stores all the oils for collisions. Every level has 2 oils even though, one is showing. The reason behind this design choice is that it allow using rectangles for collisions.

- girls is a new ArrayList<Girl> that stores all the girls for collisions. Every level has 2 girls even though, one is showing. The reason behind this design choice is that it allow using rectangles for collisions.

- monkeys is a new ArrayList<Monkey> that stores all the monkeys for collisions. Every level has 4 monkeys even though, one is showing. The reason behind this design choice is that it allow using rectangles for collisions.

- barrelSpawnX and barrelSpawnY are integers to initialize the barrel starting point.

- finishX and finishY are integers to initialize the finishing point.

**Removed Variables**

- map is removed due to the fact that nonmovable is also doing the same operation.

- enemies is removed due to design purposes. barrels and fireElementals are used for similar purposes.

**New/Renewed Methods**

- collisionWithFinishPoint(): boolean, collisionCoinAndPlayer(): void, collisionBarrelAndPlayer(barrel : Barrel): boolean, collisionFireElementalAndBarrel(fireElemental : FireElemental): boolean, collisionBarrelAndFireElemental(barrel: Barrel) : void, collisionBarrelAndOil(barrel : Barrel) is added/renewed to handle the collision between system objects.

- reloadMap(): void is added to reload the current map when player dies and has enough lives.

- createBarrel(x : int, y : int, enemyType : EnemyType) : void is added to allow Control class to create new barrels.

- rollEnemyType() : EnemyType assign random enemy type to newly created barrel.

- moveBarrel(barrel : Barrel) : boolean creates movement for the barrel which it takes as parameter. Returns true if player dies else return false.

- moveFireElemental(fireElemental : FireElemental) : boolean creates movement for the fire elemental which it takes as parameter. Returns true if player dies else return false.

- fallBarrel(barrel : Barrel) : void is the barrel algorithm for barrel which it takes as a parameter should fall from the point or not.

- climbFireElemental(fireElemental : FireElemental) : void is the climb algorithm for the fire elemental which it takes as parameter

- updateMonkeyTimer(second : long) : boolean, updateObjectTimer(second : long) : boolean, updateHammerTimer(second : long) : boolean are used for decide if the nonmovable objects and player should change their image.

- updateObjects() : void, updateMonkey() : void, updateHammer() : boolean are methods that are used for notifying nonmovable objects and player to create movement.

Girl, Monkey and Oil relatively remained same except for the fact that a new method is added to create animation.

Player class is also remained same except for animation methods which are animationJump() : void, animationHorizontal() : void, animationClimb() : void, animationClimbEnd() : void.

## 1.2 Controller Subsystem

Controller Subsystem includes Controller, MapData, ScoreData, SoundManager, UnlockData, classes.

MapData, ScoreData, UnlockData classes remained same except for the fact that they start using ArrayList which can be seen in Design Report.

SoundManager class is still causing bugs, thus we are not sure the design will remain same, but currently no changes in terms of methods.

Controller class is a newly added class that is initialized in Design Report, but not in Analysis Report. Its main purpose is to remove controller from view in GamePanel.

### 1.2.1 Controller

Controller creates a tread for the game and run it instead of GamePanel which is only responsible for drawing the game screen.

There is no method added since Design Report, but some methods are expanded.

## 1.3 View Subsystem

View Subsystem includes GUIPanelManager, HighscoresPanel, OptionPanel, MainMenuPanel, GamePanel, LevelSelection, HelpPanel, CreditsPanel, EndGamePanel classes.

HighscorePanel, OptionPanel, MainMenuPanel, CreditsPanel, HelpPanel classes are not changed.

GUIPanelManager remained relatively same. For every new panel, a new setVisible method is added to GUIPanelManager and a SoundManager as a parameter to send it to Controller and OptionPanel.

GamePanel is a new addition to this subsystem which is added in Design Report. Thread is removed from this class and moved to Controller class. Only drawing part is remained here. Only new function added to this class is notified(nonmovable : ArrayList<Nonmovable>, barrels : ArrayList<Barrel>, fireElementals :

ArrayList<FireElemental>, player : Player, score : int, remainingLives : int) : void which is used for notifying this function to render the screen by Controller.

EndGamePanel is recently added class to the system. It is used for updating the high scores. If player achieves a high score, instead of sending back to main menu, they will be send to this panel.

# 2. Current Status of the System

## 2.1 Model Subsystem

- Death animation is missing in Player class.
- Monkey does not know which colored barrel will spawn. Therefore it does not know the correct image to display.

## 2.2 Controller Subsystem

- SoundManager is not working correctly in Controller class. Monkey stop moving synchronized when player dies in Controller class.

## 2.3 View Subsystem

- View subsystem is currently done. There might be some bugs which we might not realize.

# 3. User Guide

## 3.1 System Requirements

Game requires 1000x1000 pixel screen.

## 3.2 Installation

Installation only requires JRE (Java Runtime Environment) to play the game.

## 4. User Interface


Figure 1: Main Menu

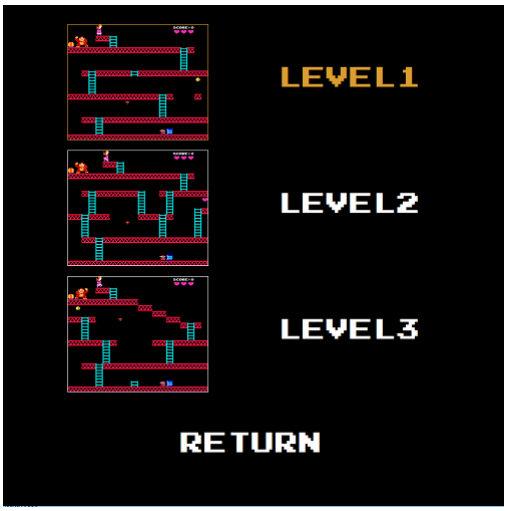


Figure 2: Level Selection Menu


Figure 3: High scores Menu


Figure 4: Options Menu

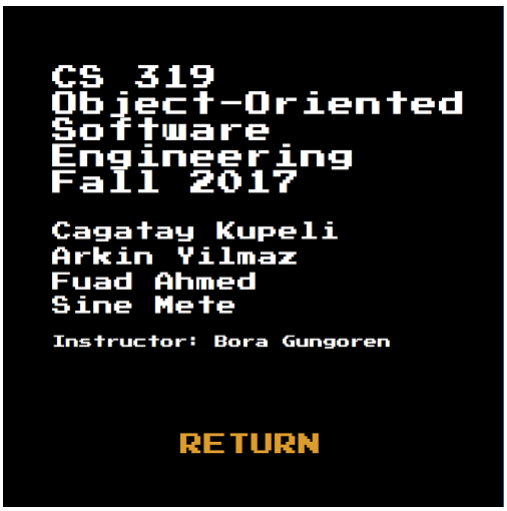


Figure 5: Credits Menu


Figure 6: Help 1 Menu


Figure 7: Help 2 Menu


Figure 8: Help 3 Menu


Figure 9: Help 4 Menu


Figure 10: Help 5 Menu

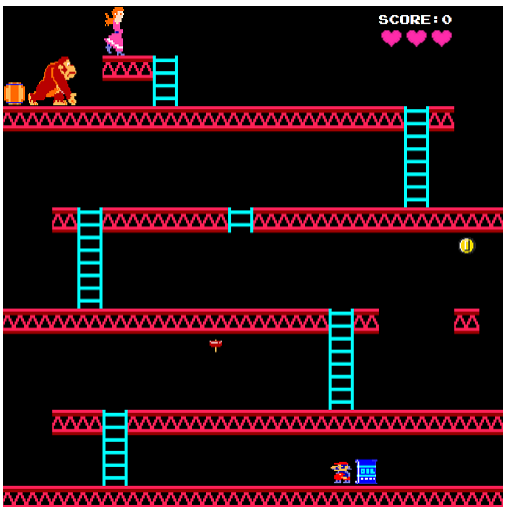


Figure 11: Level 1

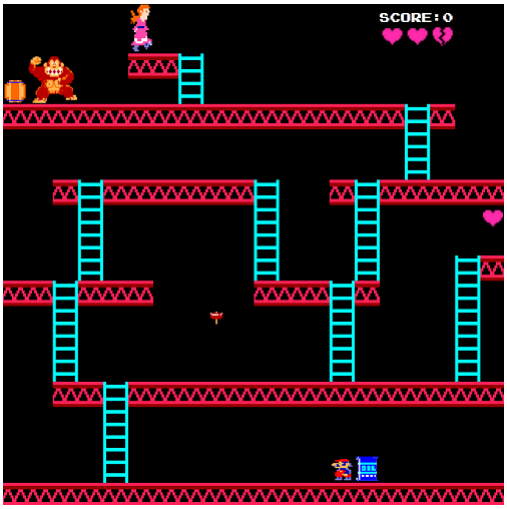


Figure 12: Level 3

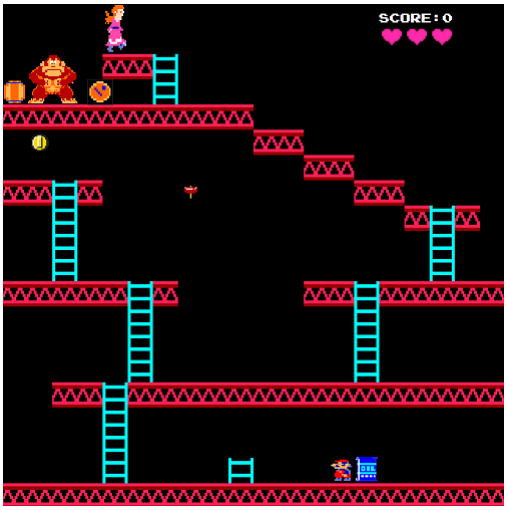


Figure 13: Level 2

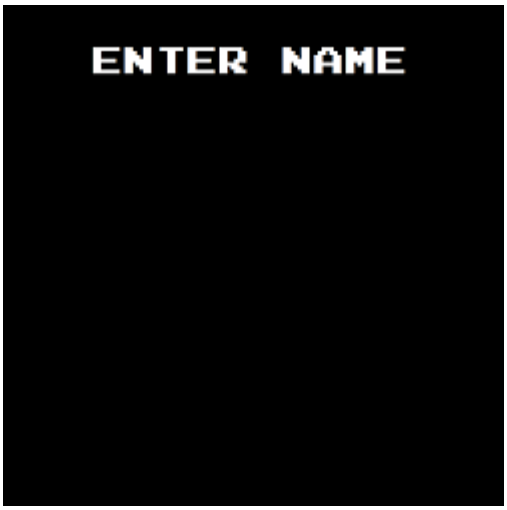


Figure 14: Update Score