Software Requirements and Design Document

For

Group <12>

Version 3.0

Authors(name, FSUID, GithubID):

Modibo Traore, mt16g, modibot

David Lee, dyl20b, dyl20b

Alejandro Osuna, ao18f, aosuna5861

Daniel Kovacs, djk19f, LivingBrovacs

Matthew Papageorge, mp20gu, mattpapa3

1. Overview (5 points)

An isometric top down adventure game based on the gameplay of the original Frogger. Super Lizard takes inspiration from roguelike games in conjunction with Frogger by adding various levels, a traditional limited life and retry system, as well as potential expansions in the form of power ups or other features.

Super Lizard will be written in C# and the game will be made within the Unity game engine which operates primarily in C#.

2. Functional Requirements (10 points)

*List the **functional requirements** in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. *When the game is started the start screen will be showing and there should be two buttons “play” and “quit”. This is a medium priority because this is how the game can be started.*
2. *When “play” is clicked the game will begin and there will be cars going across a road with the lizard starting on the bottom of the screen. This is a high priority requirement because this is how the start of the game should look like.*
3. *When any of the arrow keys are clicked the lizard will move up, down, left, or right based on the input. This is a high priority requirement because this is how the game will be played and how the player will be allowed to control the lizard.*
4. *If the lizard gets to the other side of the road it will be reset back to where it started and 100 points will be added to the score. This requirement is medium priority because this allows the player to continue playing the game without having to reset it and keeps track of the score that*

the player has. It is not high because the game would still be playable without it but this allows the player to continue playing the game without any interruptions.

5. *If the Lizard collects any of the bugs 10 points will be added to the score per bug collected. This is also not too high of priority because the game would still be playable without it but it adds more to the game.*
6. *If the Lizard comes in contact with a car or a raindrop a game over screen will appear with their final score and the user can select "new game" if they want to play again. If the player selects "new game" the game will restart and their score will be reset to 0. This is a high priority because if this is not working then it would be impossible to lose and you would not be able to restart the game.*

3. Non-functional Requirements (10 points)

*List the **non-functional requirements** of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

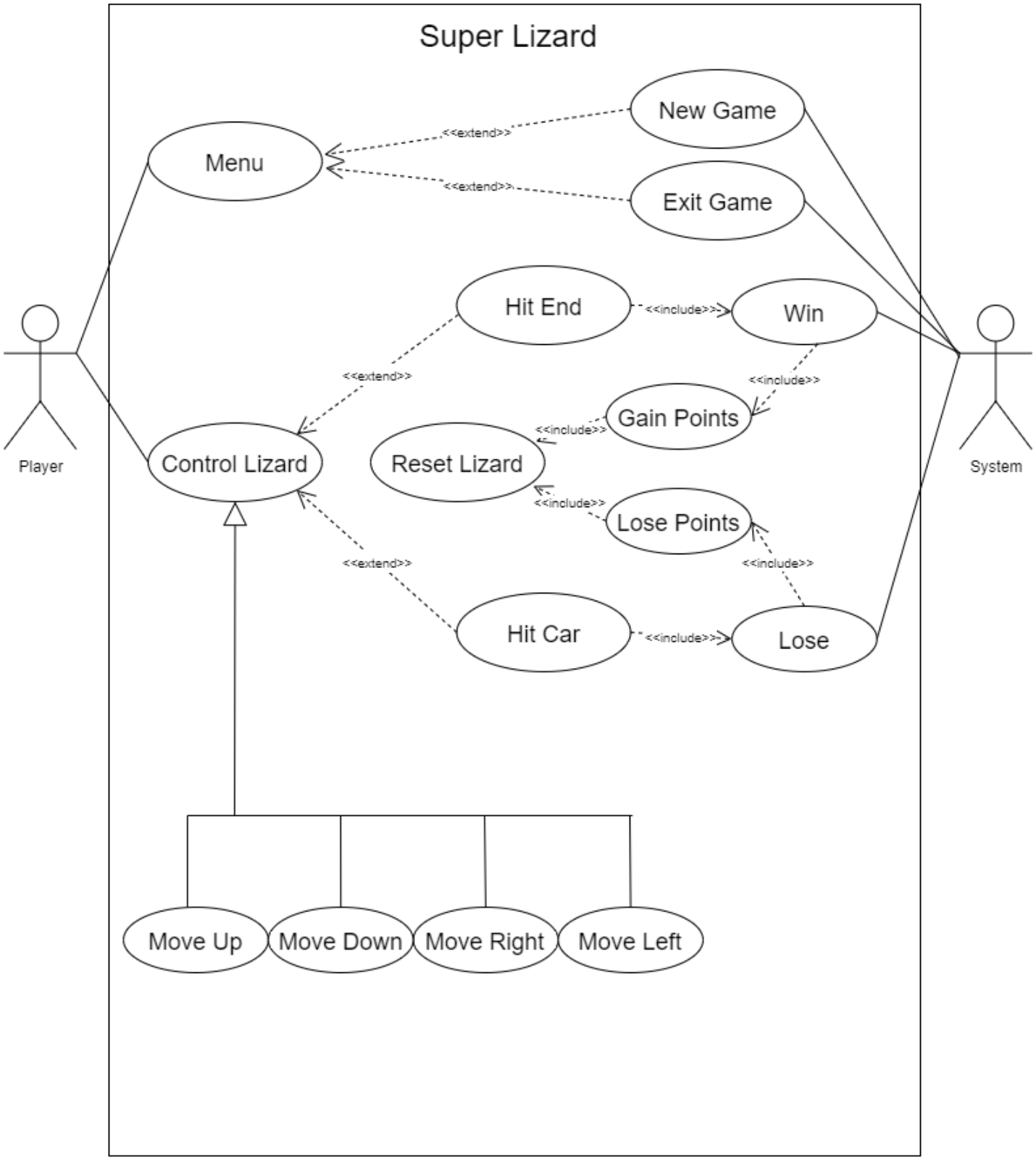
The game should efficiently manage memory from allocation to deallocation of game objects.

The game should be simple in that it doesn't require a really strong computer to run.

The user interface should be appealing, clean, and readable.

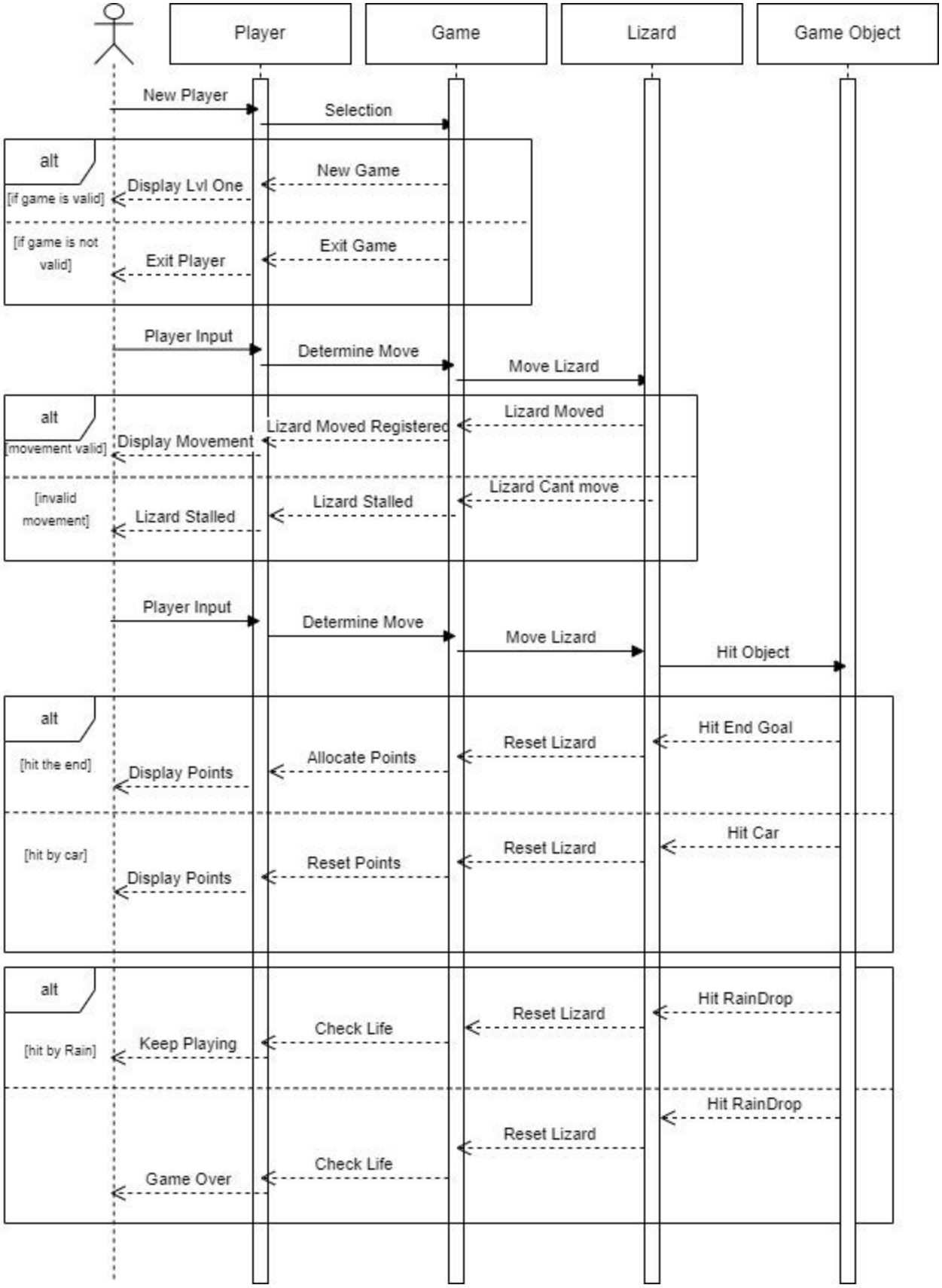
Hit boxes for obstacles should be justifiable, there shouldn't be too harsh of hit boxes so that it annoys the player.

4. Use Case Diagram (10 points)



Use Case	Description
Menu	When entering the menu the player would be prompted to either begin a new game or exit the game entirely in which the system will execute.
Control Lizard	The lizard can be controlled by the player by inputting directional keys. If the lizard collides with either obstacle, car or end goal, then the following points would be determined and the lizard would be reset. The system will handle the allocation or deallocation of points.

5. Class Diagram and/or Sequence Diagrams (15 p


Player
Game
Lizard
Game Object
New Player
Selection
alt
[if game is valid]
Display Lvl One
New Game
[if game is not valid]
Exit Player
Exit Game
Player Input
Determine Move
Move Lizard
alt
[movement valid]
Display Movement
Lizard Moved Registered
Lizard Moved
[invalid movement]
Lizard Stalled
Lizard Cant move
Lizard Stalled
Player Input
Determine Move
Move Lizard
Hit Object
alt
[hit the end]
Display Points
Allocate Points
Reset Lizard
Hit End Goal
[hit by car]
Display Points
Reset Points
Reset Lizard
Hit Car
alt
[hit by Rain]
Keep Playing
Check Life
Reset Lizard
Hit RainDrop
Game Over
Check Life
Reset Lizard
Hit RainDrop

6. Operating Environment (5 points)

The software will be its own application that will be run on a computer that is rather a Windows or Mac. The operating system will be any Mac OS or if run on Windows, Windows 7 would be preferred. The software will be used by using the arrow keys on the keyboard. The only application required will be the SuperLizard application.

7. Assumptions and Dependencies (5 points)

List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

Computer has enough storage and RAM to run application and the user has a working keyboard.

The game at its current state is “unfinished” in that we were unable to meet our set goals from the original plan. The game SuperLizard does not contain randomized levels and doesn’t feature any powerups or anything of that sort. It does accomplish the many lives and scoreboard keeping however.