# Team: Fenrir

Kiourtsis, Dean Joesph, Project Manager  
Zhang, Dabai, Release Manager  
Earisman, Jesse Kyle, Quality Assurance Manager  
Charron, Justin Scott, Technical Design Lead  
Finelli, Thomas Michael, Scribe  
Liu, Yang, Pizza Guy

## Summary

There were two meetings this week: one on Saturday, March 21st and one on Monday, March 23rd. On 3/21 the user stories and use cases for the game and level builder were drafted. On 3/23 the use cases and user stories were refined, fundamental objects for the game were identified using the use cases as a basis, quality assurance procedures were discussed, weekly meeting times were addressed, and a rotation for group roles was determined.

## User Stories

### Game User Stories

* As a player, I want to Make a move so that I can complete the game objectives.
* As a player, I want to Reset the Board so that I can clear an unfavorable configuration of numbers.
* As a player, I want to swap two numbers so that I can put the board into a better configuration.
* As a player, I want to clear a number so that I can put the board into a better configuration.
* As a player, I want to restart the level so that I can try the level again from the beginning.
* As a player, I want to see all accessible levels so I can view which levels I have unlocked, and what my scores were.
* As a player, I want to play a past level so that I can get a better score, or replay an enjoyable level.
* As a player, I want to view my achievements so that I can check what achievements I have.
* As a player, I want to reset my achievements so that I can re-unlock them.
* As a player, I want to continue with highest unlocked level so that I can quickly resume my game progress.
* As a player, I want to quit a level so that I can do something else.
* As a player, I want to quit the game so that …
* As a player, I want to exit a menu so that I can go back to the previous menu

### Level Builder User Stories

* As the programmer, I want to make a new level so that there are levels that can be played.
* As the programmer, I want to load an existing level I can edit it.
* As the programmer, I want to set the mode of a level so that the mode will be set.
* As the programmer, I want to set a specific square so that initial positions can be set.
* As the programmer, I want to specify generation probabilities so that levels can vary.
* As the programmer, I want to save the level so that it will be playable.
* As the programmer, I want to set the number of moves/time limit of the level so that the player will have a limit.
* As the programmer, I want to set multiplier chances so that levels can vary.
* As the programmer, I want to set score boundaries (1,2,3 star) so that the levels will be hard.
* As the programmer, I want to re-order the levels so that …
* As the programmer, I want to delete a level so that it can’t be used

## Objects

### Game Objects

Some objects that are needed for the game are: a board to play on; multiple levels with varying difficulties and varying level type; a tile object for each number on the board; and an achievement object to keep track of the player’s progress towards badges. We created a table of classes [below] that would pair with these objects. Additionally, we define some possible attributes and behavior for each class/object in the table below.

|  |  |  |
| --- | --- | --- |
| **Class/Object** | **Attributes** | **Methods** |
| **Board** | Tiles[] | initialize()  reset() |
| **Tile** | value  multiplier  marked |  |
| **Level** | score  stars  numMoves |  |
| **Achievement** |  | show() |
| **SixesWild** | level |  |
| **Move** | tiles  isValid |  |
| **SpecialMove** | type  amount |  |

### Level Builder Objects

Some objects that are needed for the level builder are: a buildable level object, where the level is designed; multiple tile objects which can be inserted, moved, or deleted into/from the level being designed; and a move object which can be used to validate moving tiles around on the level. We created a table of classes [below] that would pair with these objects. Additionally, we define some possible attributes and behavior for each class/object in the table below.

|  |  |  |
| --- | --- | --- |
| **Class/Object** | **Attributes** | **Methods** |
| **LevelBuilder** | level |  |
| **Level** | Tiles[]  type  timeLimit  movesLimit  multiplierChance | new()  load()  delete()  save() |
| **Tile** | value  marked  multiplier | new()  delete() |
| **Move** | tiles  isValid |  |

## Quality Assurance

To ensure quality for the first deliverable (i.e. use cases) we will: review each use case to make sure it is valid and necessary, and we will make sure that everyone understands each use case so that the big picture makes sense to all group members. An outline of the process that we can follow is:

1. Create a use case
2. Have 1-2 other group members review the use case for: necessity, validity, conciseness, and overlap with other use cases
3. Have everyone read all use cases to ensure a group understanding
4. Brainstorm about any missing use cases, and go back to step 1, if necessary

## Schedule

We have found that Wednesday afternoons, weekends, Mondays, and Fridays work best as meeting times for our group.

For the duty rotation: we have decided to use the initial role list as a basis. After each week, we will shift everyone’s name 1 space down on the list, resulting in a new duty for each person.

## Appendix A: Game Use Cases

We have already developed many of the use cases from the user stories, so we thought we would include them in this report:

|  |  |
| --- | --- |
| Name | Make Move |
| Participating Actor | Player |
| Entry Condition | A game is ongoing, there are 2 or more adjacent tiles to use |
| Exit Criteria | If the tiles form a valid move, they are removed and replaced  If they do not form a valid move, the system notifies the user of their error |
| Flow of Events | 1) User requests to use 2 or more tiles to make a move  2) Either the move is executed, or SixesWild informs the user that the move is invalid |

|  |  |
| --- | --- |
| Name | Reset Board |
| Participating Actor | Player |
| Entry Condition | There is a board to reset, the user has 1 or more “Reset Board” special moves available |
| Exit Criteria | All numbers on the board (except 6's) are randomized to different numbers, moves are incremented by one |
| Flow of Events | 1) User requests to reset the board  2) The board is reset |

|  |  |
| --- | --- |
| Name | Swap Numbers |
| Participating Actor | Player |
| Entry Condition | Their are two adjacent tiles to swap, the user has 1 or more “Swap numbers” special moves available |
| Exit Criteria | The two numbers are swapped, the “Swap numbers” move is used up |
| Flow of Events | 1) The Player requests to swap two number  2) The two numbers are swapped |

|  |  |
| --- | --- |
| Name | Clear Number |
| Participating Actor | Player |
| Entry Condition | There is a number eligible for deletion (eg. 6's in release mode are not eligible), The player has 1 or more “Clear number” Special moves available |
| Exit Criteria | The selected number is removed and replaced |
| Flow of Events | 1) The user requests to clear a number  2) The number is cleared and the board is updated |

|  |  |
| --- | --- |
| Name | Restart Level |
| Participating Actor | Player |
| Entry Condition | The player is currently playing a level |
| Exit Criteria | The level is reset, the number of moves mode and score are set to 0 |
| Flow of Events | 1) The player requests to reset the level  2) The level restarts |

|  |  |
| --- | --- |
| Name | See all accessible levels |
| Participating Actor | Player |
| Entry Condition | The player is at the main menu |
| Exit Criteria | The player is taken to a view where they can see all accessible levels |
| Flow of Events | 1) The player requests to see all accessible levels  2) SixesWild opens a view where the player can see their accessible levels |

|  |  |
| --- | --- |
| Name | Play accessible level |
| Participating Actor | Player |
| Entry Condition | The player has accessible levels, the player is in a view where they can see all accessible levels |
| Exit Criteria | The player is taken to the chosen level, the level begins |
| Flow of Events | 1) The player requests to play an accessible level  2) The chosen level starts |

|  |  |
| --- | --- |
| Name | View achievements |
| Participating Actor | Player |
| Entry Condition | Player is at the main menu |
| Exit Criteria | Player has been taken to a view showing all of their achievements |
| Flow of Events | 1) User requests to view their achievements  2) SixesWild opens a view where they can see their achievements |

|  |  |
| --- | --- |
| Name | Reset achievements |
| Participating Actor | Player |
| Entry Condition | None |
| Exit Criteria | All achievements are set to an “un-achieved” state |
| Flow of Events | 1) User requests to reset their achievements  2) SixesWild resets the user achievements |

|  |  |
| --- | --- |
| Name | Continue highest level |
| Participating Actor | Player |
| Entry Condition | Player is at the main menu |
| Exit Criteria | The highest level that the player has unlocked has started |
| Flow of Events | 1) User requests to continue with their highest unlocked level  2) The game takes the user to the highest unlocked level, and lets them begin playing |