

# PROJECT FAT CAT



Presented by Team Ocean's II:

Aditi Bhatia

Arshdeep Singh

Dipro Chowdhury

Dishant Kimtani

Suhel Mehta

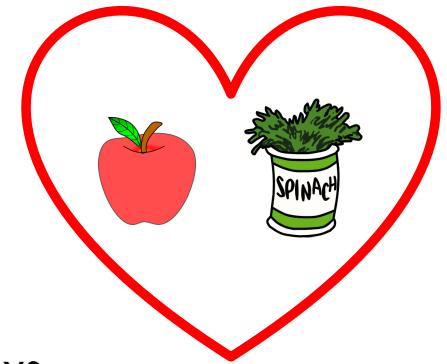
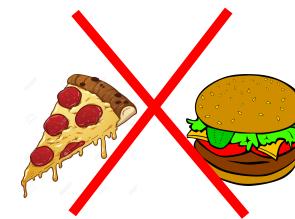
# AN OVERVIEW



- Inspired by “Mario” and lazy cat videos
- Written in JavaScript and built using MelonJS
- Structured with 5 design patterns
- **Main objective:** avoid obstacles and finish all levels as fast as possible for a highest possible score.

# FAT CAT: HOW TO PLAY

- All movement is mapped to the up, down, right, left, and spacebar keys
  - Press the up button multiple times to jump higher.
- Should avoid unhealthy food items: pizza and hamburgers.
  - Reduces character speed, health and lowers score.
- Should eat healthy food items: apples and spinach.
  - Increases character speed, health, and increases score.
- Must avoid all obstacles (river, metal spikes, holes) which kill the character and end the game.
- **Main objective:** avoid obstacles and finish all levels as fast as possible for a highest possible score.



Key	Action
↑ W Spacebar	Jump
← A	Walk Left
↓ S	Drop Down
→ D	Walk Right
F	Full-Screen Mode

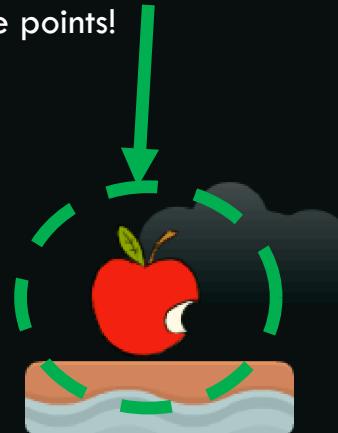
# FAT CAT

[Play Game](#)

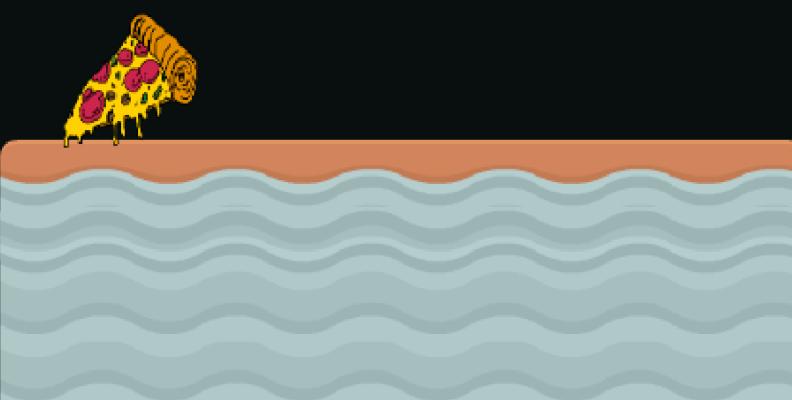


0:52

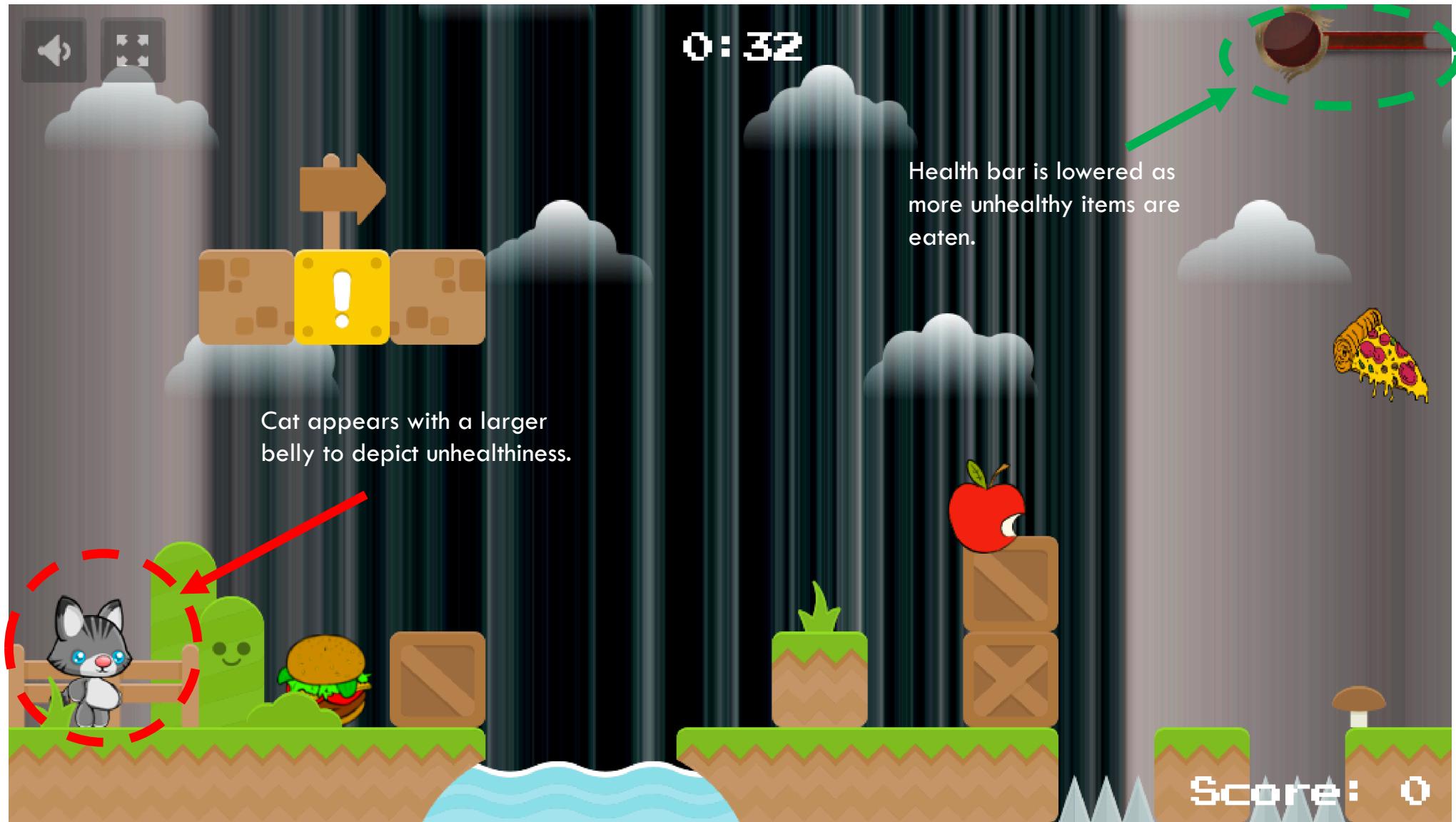
Healthy food item. Try to eat for better speed and more points!



Unhealthy food item. Try NOT to eat! Results in lower score and slower speed.



Score: 0





0:20



**Oh no, you died! :(**  
**Your total score is 0 points.**

**Want to try again?**

[Restart Game :D](#)

[See Leaderboard!](#)



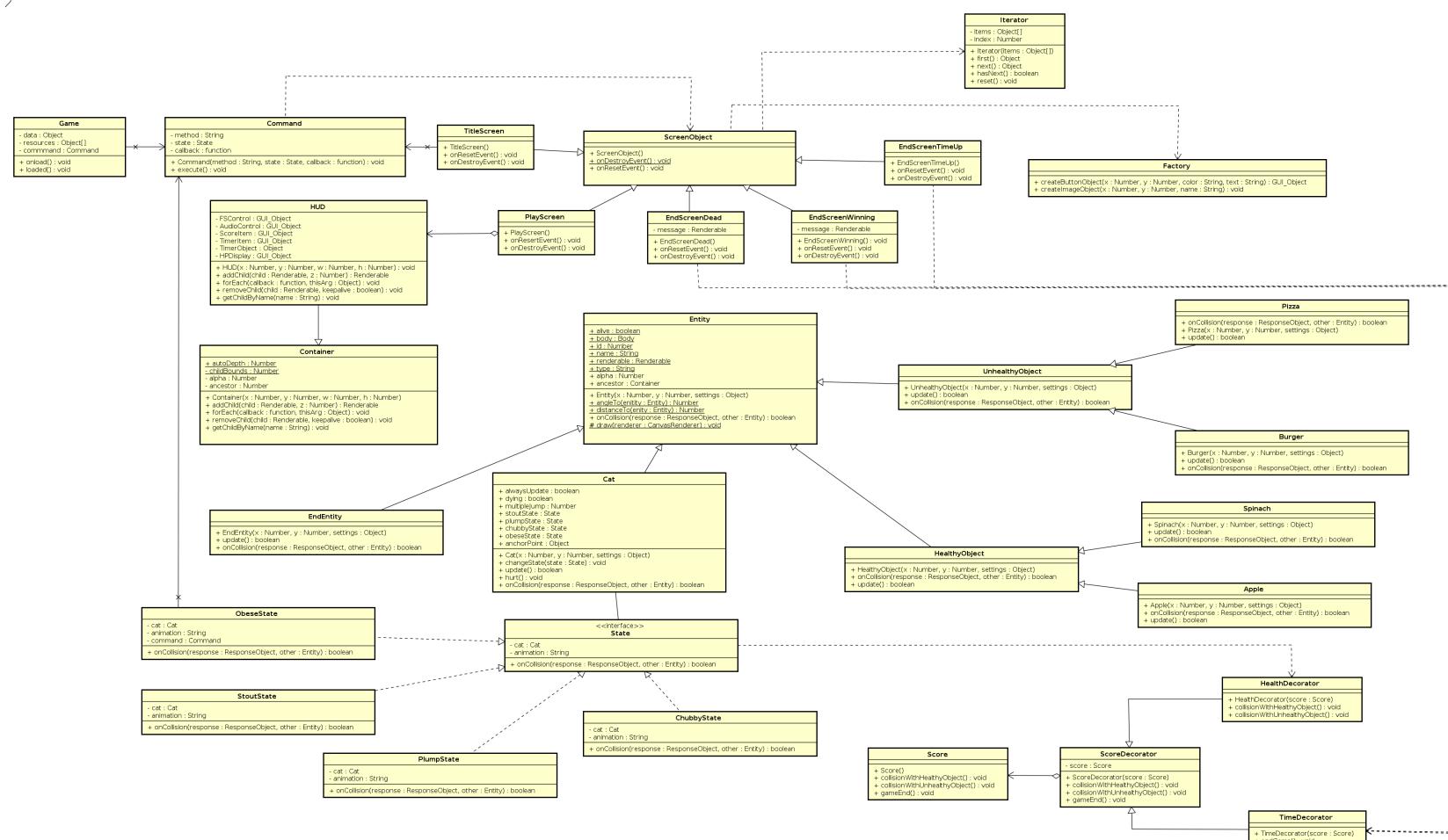
**Score: 0**



# LIVE DEMO

# DESIGN PATTERNS USED

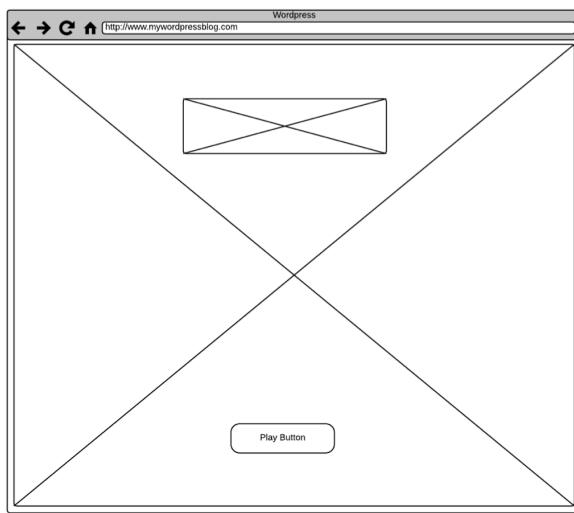
- **Decorator design pattern:** For score calculation
- **Iterator design pattern:** For displaying top 3 scores in our leaderboard
- **Command design pattern:** For changing screens and updating states during screen changes
- **State design pattern:** Used for state changes of main character
- **Factory design pattern:** Used for



# Class Diagram

## FIRSTWIREFRAME

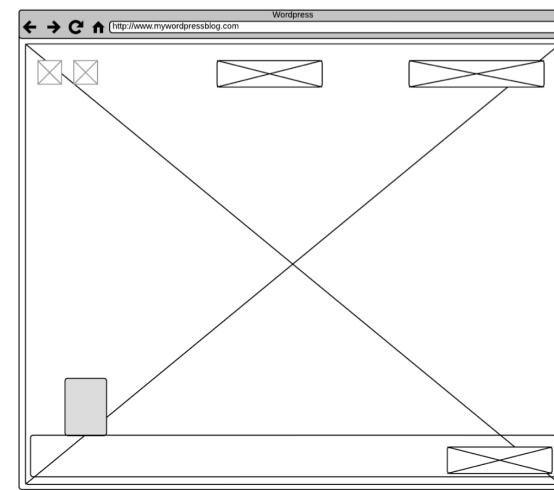
Arshdeep Singh | November 25, 2017



Select the "Layers" icon to start Demo mode and use clickable hotspot buttons

## PLAYSCREEN

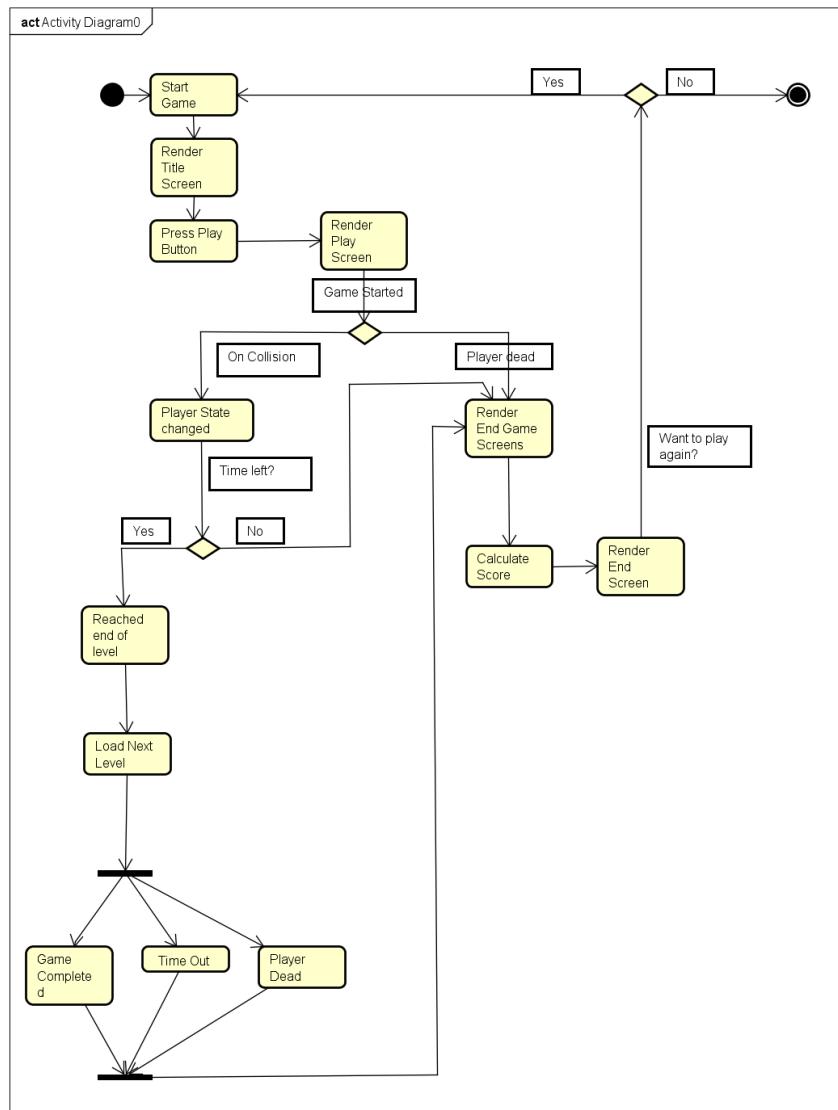
Arshdeep Singh | November 25, 2017



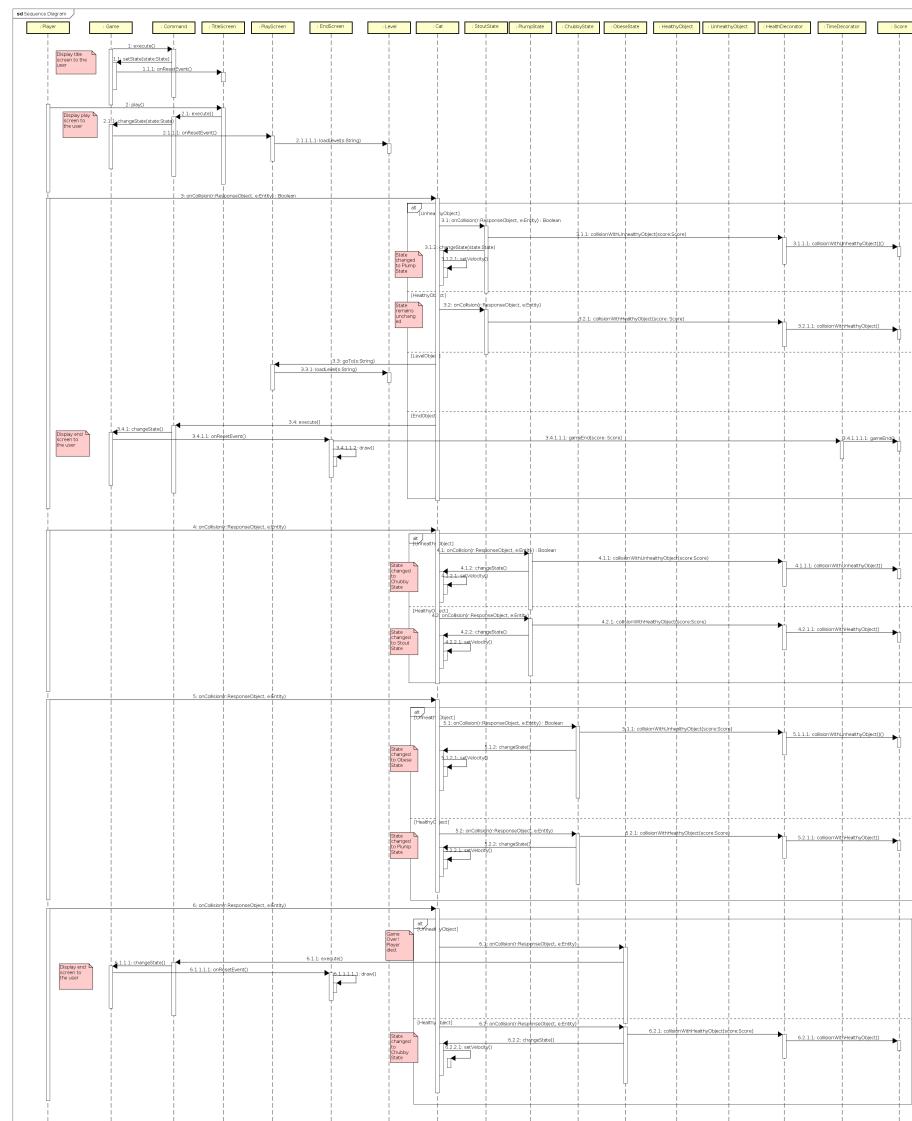
Select the "Layers" icon to start Demo mode and use clickable hotspot buttons

# UI Wireframe

# Activity Diagram



# Design Sequence Diagram



## USE CASE SPECIFICATION

### Primary actors:

- Game Players
- Burger
- Metal spikes, river bed, holes
- Burger, Pizza, Spinach, Apples

### Pre-Condition:

- Latest Web Browser (Chrome, Firefox, safari)
- Internet connection
- Platforms:
  - Laptop
  - Desktop
  - Mobile Browser

### Basic Flow of Events:

1. Player goes to the Url from browser
2. Player presses the Play button
3. Player starts controlling the cat character using up, left, down and right arrow keys in the keyboard
4. Player tries to avoid harmful obstacles (Pizza, Burger) that will get the cat fat and slow.
5. Player also tries to avoid the empty space and glass spikes that is harmful for its health.
6. Player tries to move forward as fast as possible while avoiding the obstacles to finish the level on time.
7. If Player consumes the harmful objects, it can consume good objects like apple or spinach to get its previous health back.
8. Consuming harmful objects will decrease the overall score of the player as well as health bar.
9. Consuming beneficial objects like apple and spinach will increase the score as well as health bar.
10. If the timer runs out before the player finishes a level, player will see a screen that says 'you lose' with the highest score
11. Player will also have options to see highest score for the game.
12. If player can complete all levels on time, then a new screen will show up saying 'you win' with leaderboard option as well.

# Use Case Specification

# 1

---

As a game player,  
*I want to be able to "win" the game*  
*When the game is running*  
*So that my name is at the top of the leaderboard.*

**Scenario 1:**

*Given that the Fat Cat game is running*  
*When a high score is achieved after completing all obstacles*  
*Then the player will win and be placed on the leaderboard.*

**Scenario 2:**

*Given that the Fat Cat game is running*  
*When the player dies from an obstacle but a high score is achieved*  
*Then the player will win and be placed on the leaderboard.*

**Scenario 3:**

*Given that the Fat Cat game is running*  
*When the time runs out but a high score is achieved*  
*Then the player will win and be placed on the leaderboard.*

**Scenario 4:**

*Given that the Fat Cat game is running*  
*When all obstacles are completed and a high score is not achieved*  
*Then the player will not win.*

**Scenario 5:**

*Given that the Fat Cat game is running*  
*When time runs out and a high score is not achieved*  
*Then the player will not win.*

**Scenario 6:**

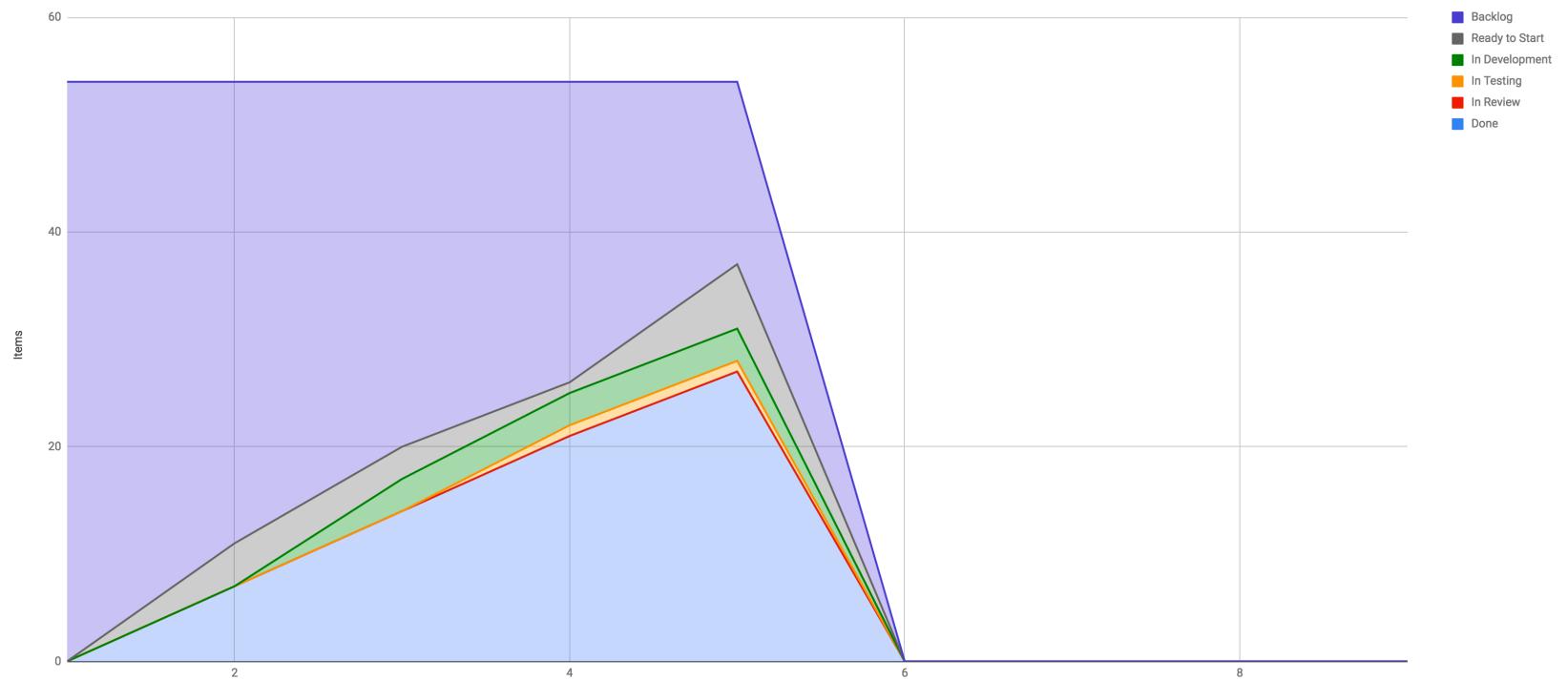
*Given that the Fat Cat game is running*  
*When the player dies from an obstacle and a high score is not achieved*  
*Then the player will not win.*

**Scenario 7:**

*Given that the Fat Cat game is running*  
*When a high score achieved has already been achieved earlier and is already on the leaderboard*  
*Then the player will not win.*

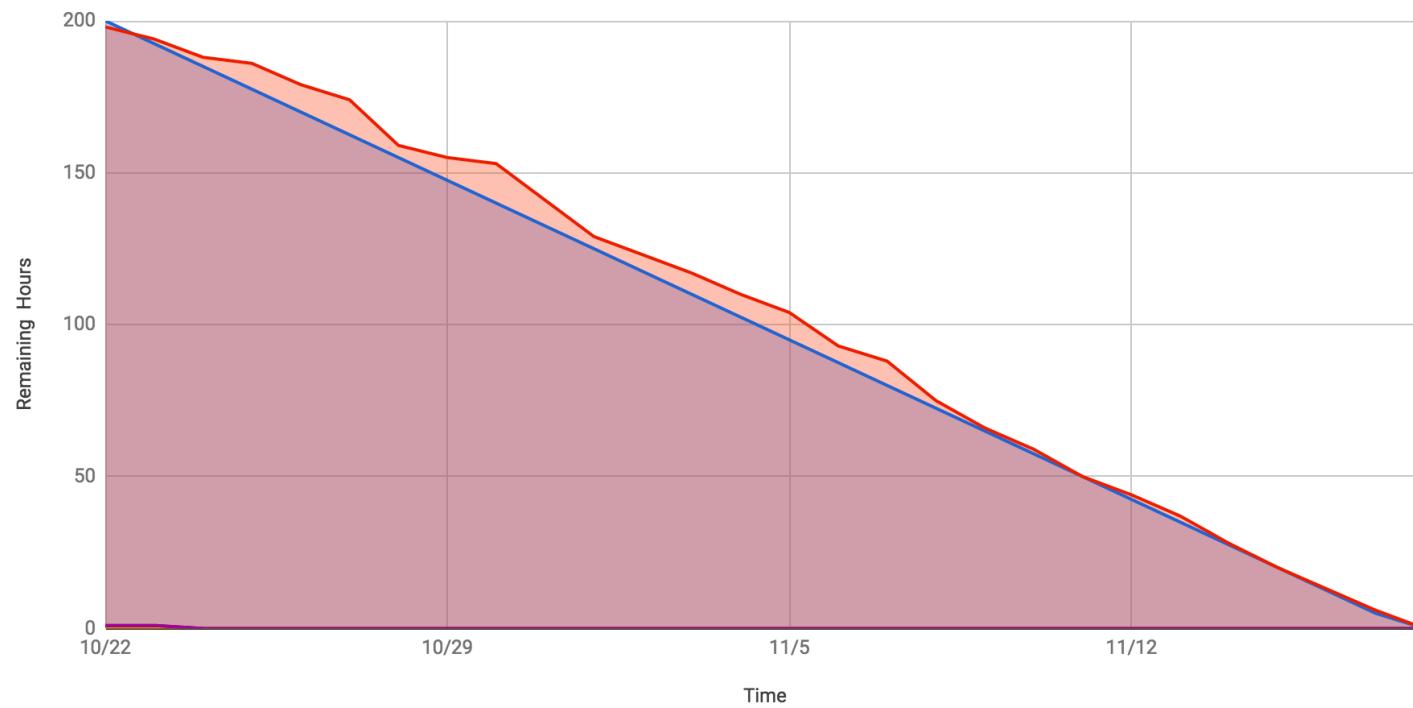
**User Story**  
(additional stories  
available in GitHub repo)

Cumulative Flow



## Kanban: Cumulative Flow Diagram

**Burndown (Team Oceans 11)**



## Scrum: Burndown Chart

**Extra Credit Demo Video**

# TEAM RETROSPECTIVE: WHAT WE DID WELL

- We believe the organization and structure of our project was executed well
  - Using Kanban, Scrum, and Agile methodologies, we were able to create tasks, assign tasks, and complete tasks within an allotted timeframe.
  - Were able to take large to-do items and split them into smaller tasks
- Organized our project so that the first couple of weeks were for research, learning, and brainstorming: so that we could have a strong game plan set up for the development weeks thereafter
- Weekly meetings allowed us to communicate about project issues and future project tasks + provide feedback to one another about current project status

# TEAM RETROSPECTIVE: WHAT WE COULD HAVE IMPROVED ON

- We believe we can improve in future development by having more specific backlog items
  - Allow for better project integration and testing throughout the week
- For future development, this will be vital since each project component will be more complex and harder to test and integrate unless testing is done for every small component
- Breaking each backlog item into even smaller items would help detect errors and issues earlier on and would help make the development process faster

# TEAM RETROSPECTIVE: FUTURE WORK

- Adding more levels of difficulty to the game with increasing complexities
- Add an option to select different characters within the game, each with different special attributes or abilities
  - Such as a tiger that can regenerate its health or a lion that starts off with higher health than normal
- Add new objects and obstacles to the game
  - different "healthy" and "unhealthy" objects
  - Other objects that would have different and special properties (i.e. an orange which gives the character 2x health or speed once consumed)
  - Will make the game more engaging and challenging

# RUN ON OUR GAME ON LOCALHOST

1. Clone our git repository.
2. Download and install [MelonJS](#)
3. Using the command line, enter the directory where the game is cloned.
4. Run "git serve" and open <http://localhost:8000/> in your browser.

**THANK YOU!**