PROJECT **FAT CAT**

Presented by Team Ocean's 11:

Aditi Bhatia

Arshdeep Singh

Dipro Chowdhury

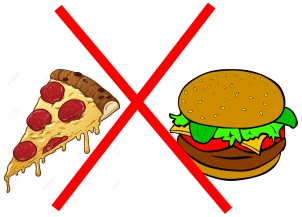
Dishant Kimtani

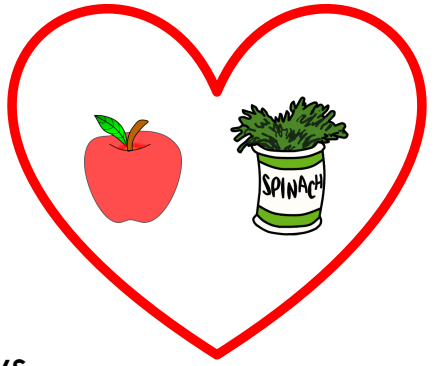
Suhel Mehta

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



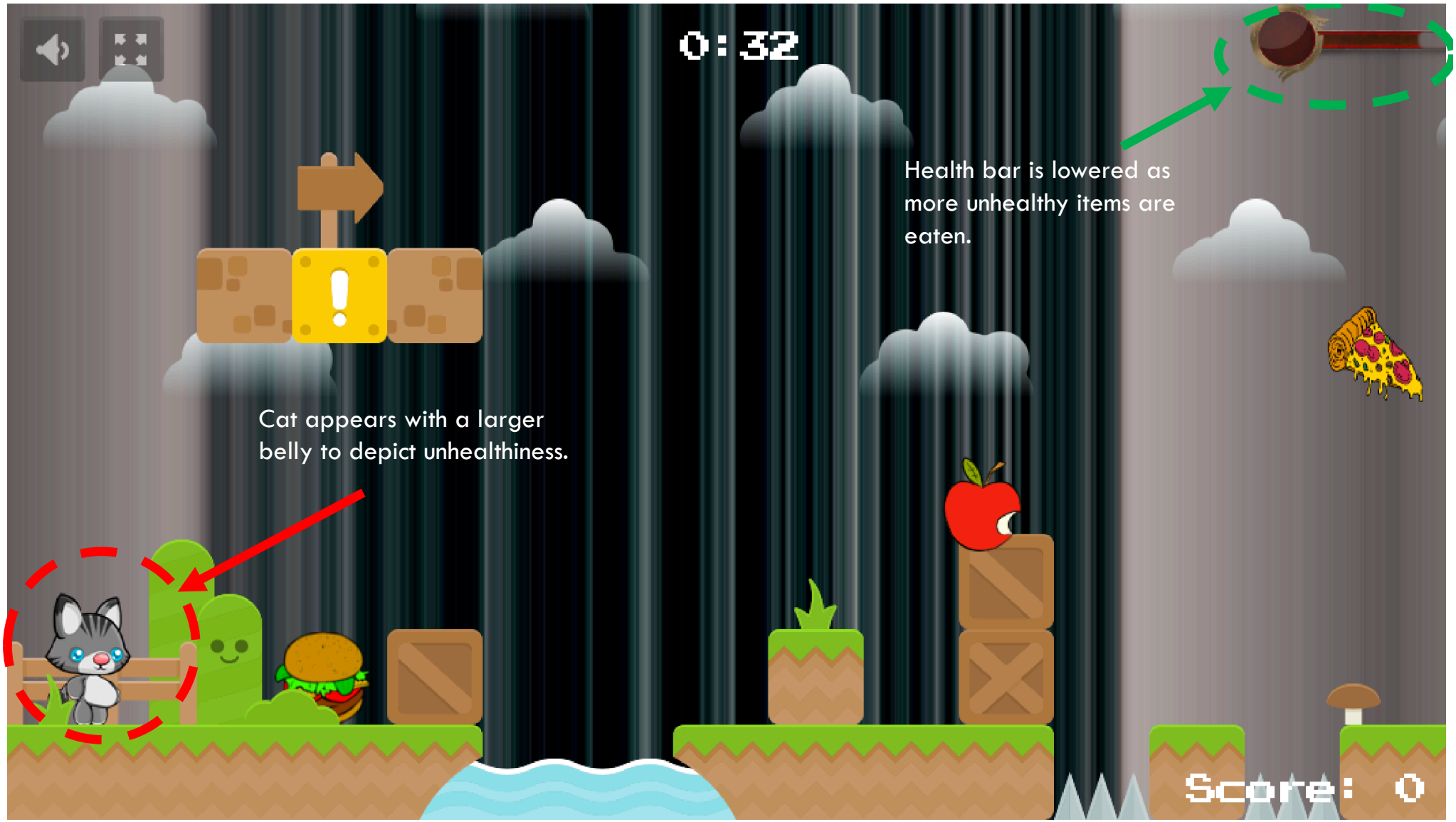


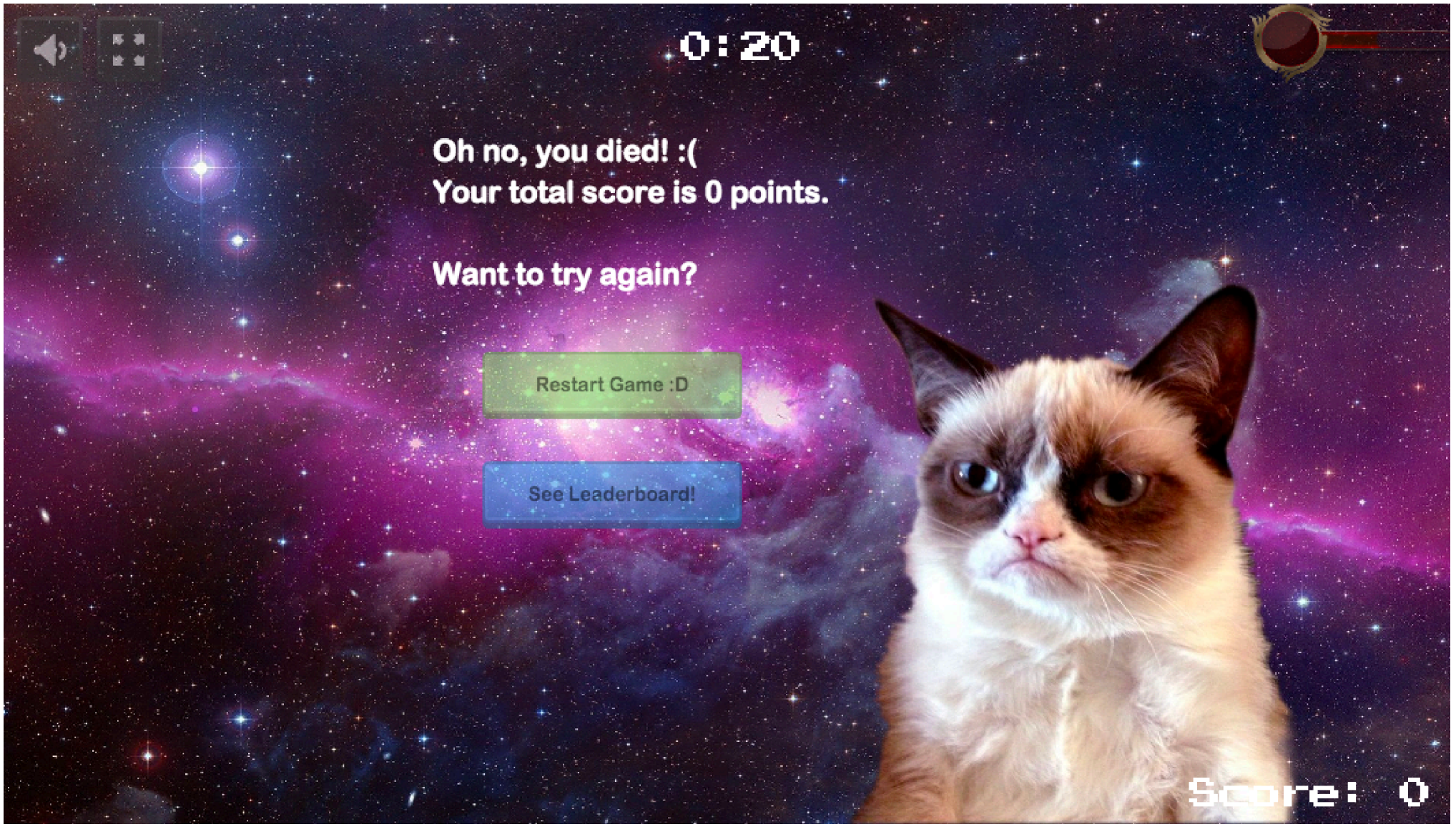
- All movement is mapped to the up, down, right, left, and spacebar keys
  - Press the up button multiple times to jump higher.
- Should avoid unhealthy food items: pizza and hamburgers.
  - Reduces character speed, health and lowers score.
- Should eat healthy food items: apples and spinach.
  - Increases 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


0:32
Health bar is lowered as more unhealthy items are eaten.
Cat appears with a larger belly to depict unhealthiness.
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

Ul Wireframe

Design Sequence Diagram

Use Case Specification

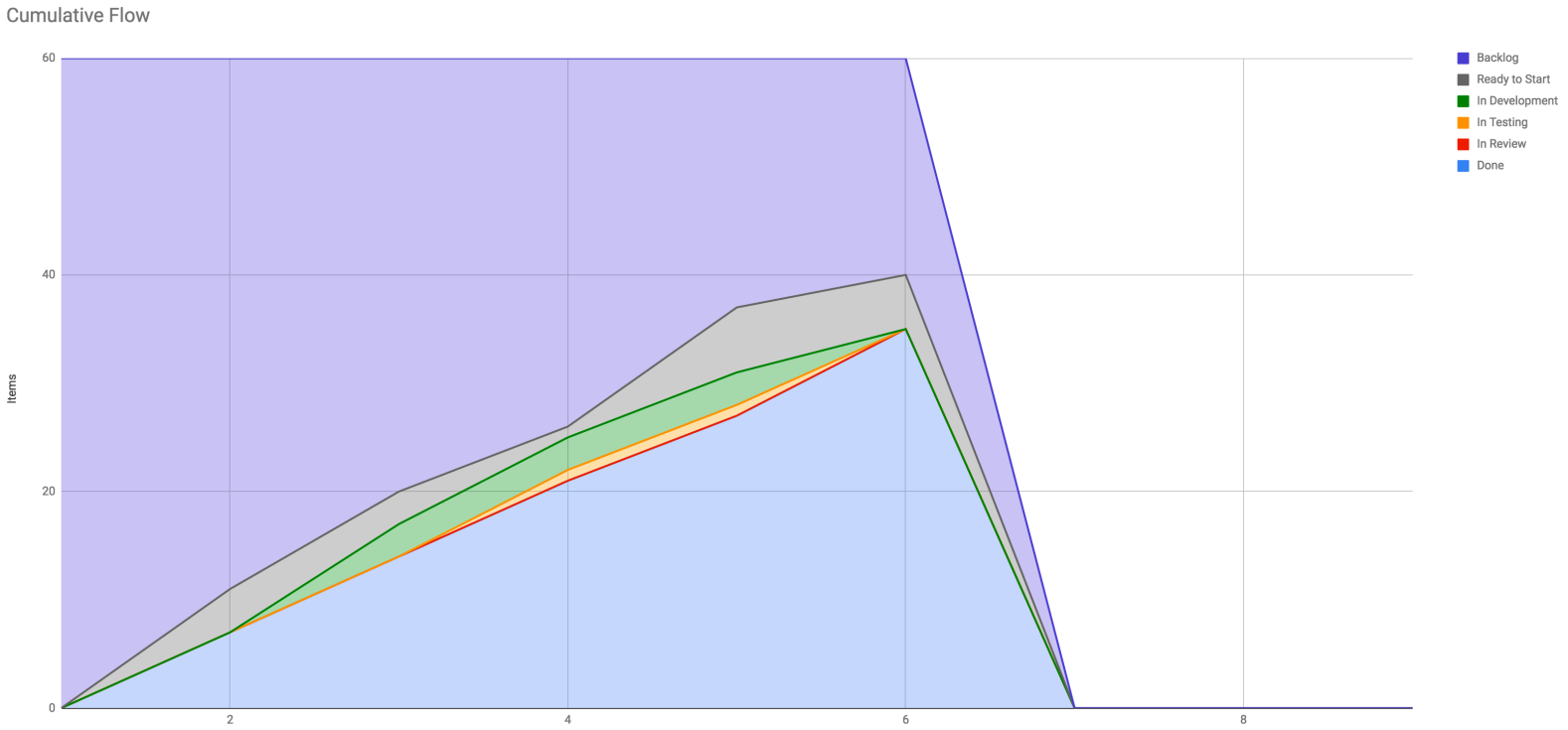
User Story

(additional stories available in GitHub repo)

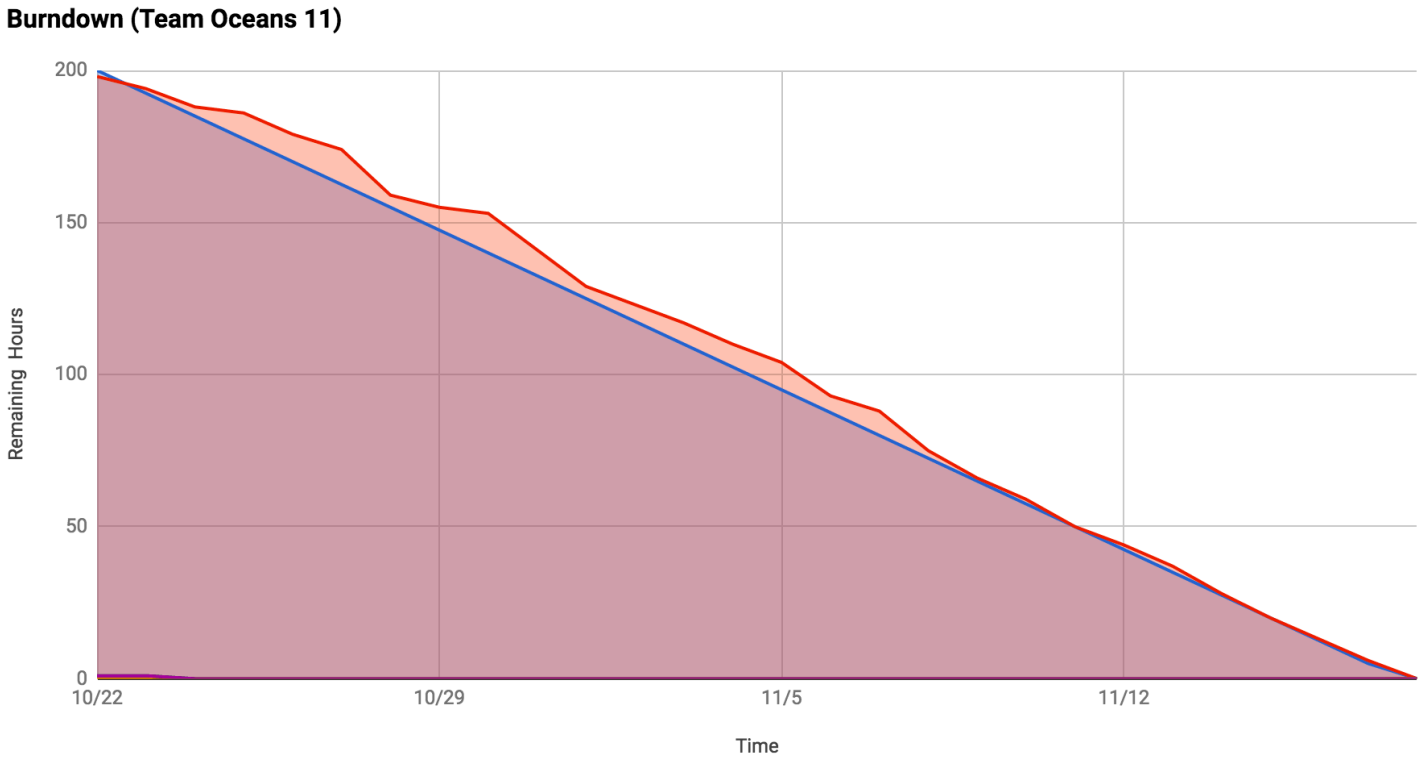
1

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.



Kanban: Cumulative Flow Diagram



Scrum: Burndown Chart

Extra Credit Demo Video

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!