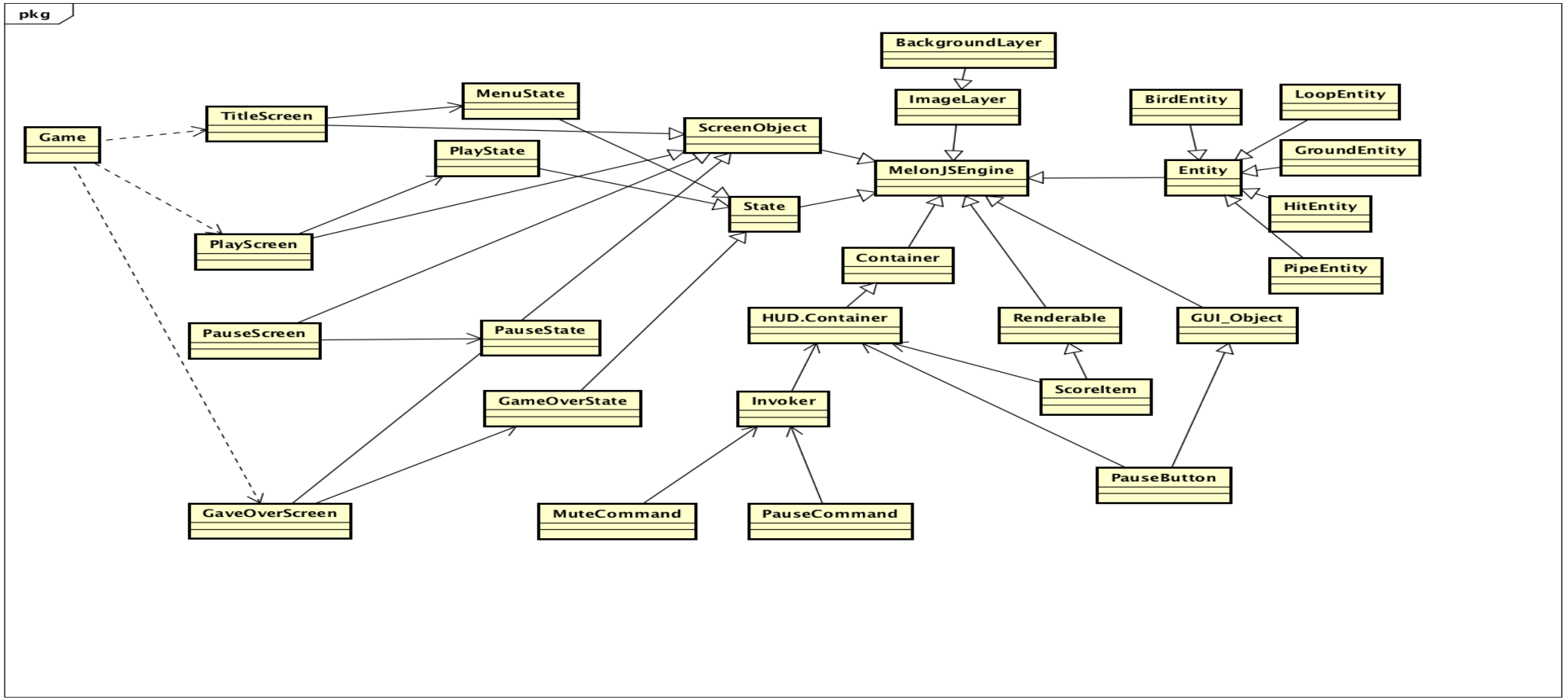

LOOPY BIRD

Loops

- Aartee Kasliwal
- Junaid Khan
- Prathmesh Parmar
- Shikhar Gaur

- UML Diagrams
- Final Task Board
- Cumulative Flow Diagram
- Sprint Burndown Chart
- Sprint Retrospective
- UI Wireframes

https://github.com/nguyensjsu/cmpe202-loops/tree/master/Class-Diagram

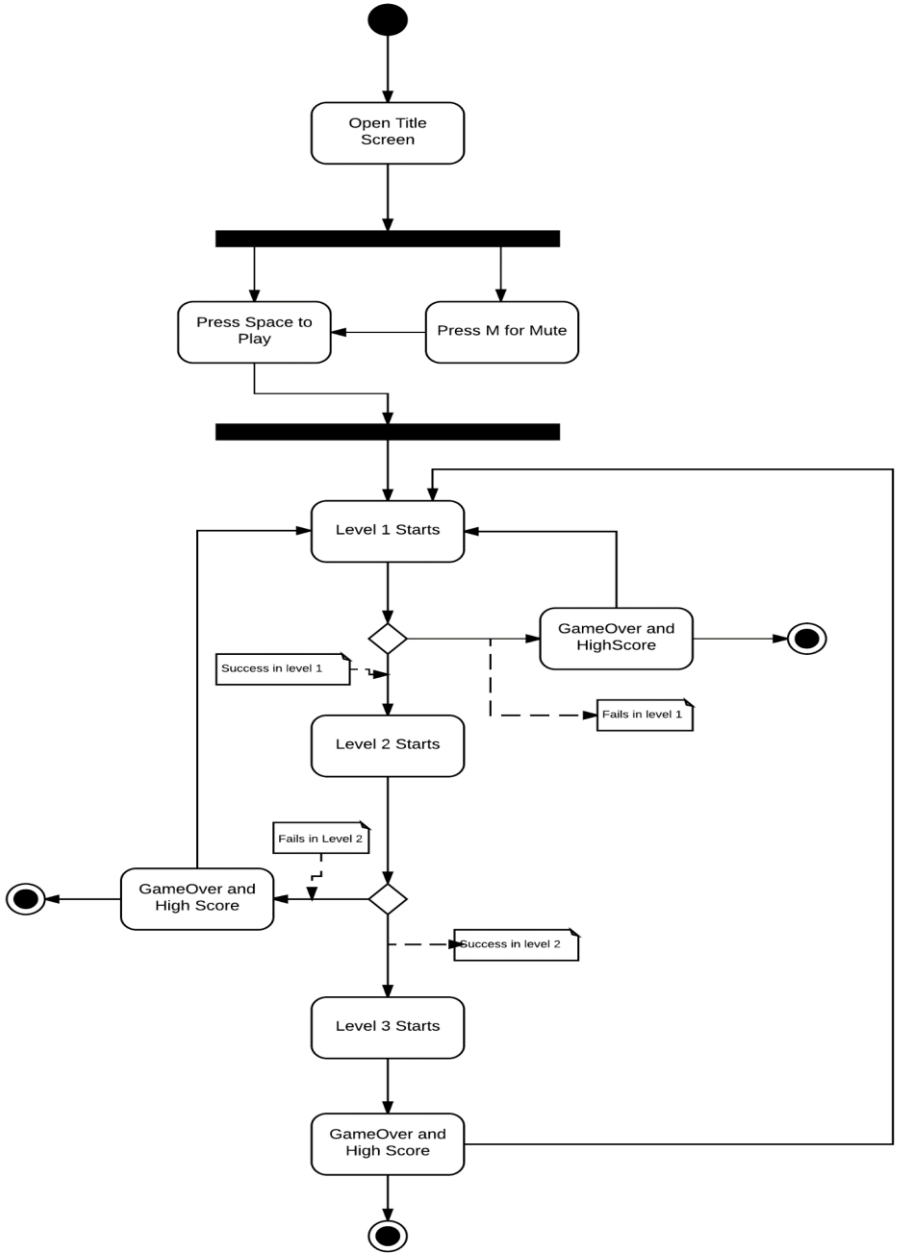


Class Diagram: Objects inherited from MelonJS Engine

Design Patterns Implemented in the Game

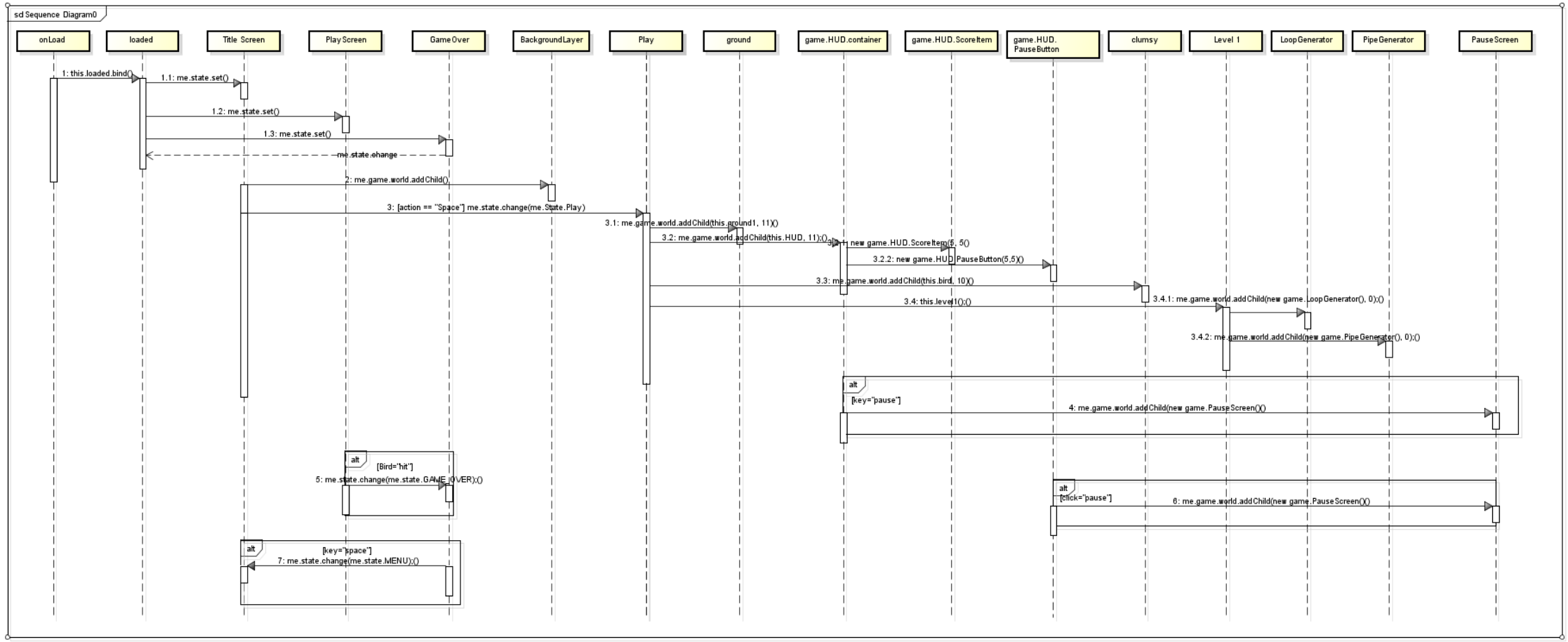
Design Pattern Name		Class/Interface
Command Pattern	Concrete Command	MuteCommand, PauseCommand
	Client	HUD.Container
	Invoker	Invoker
	Receiver	MelonJSEngine
State Pattern	Context	State
	States	MenuState/PlayState/PauseState/GameOverState
Composite Pattern	Client	Game
	Component	ScreenObject
	Leaf	TitleScreen/PlayScreen/PauseScreen/GameOverScreen
Proxy Pattern	Client	ScreenObject
	Subject	MelonJSEngine
	Real Subject	Game
	Proxy	ScoreItem
Observer Pattern	Subject	MelonJSEngine
	Concrete Subject	BirdEntity/LoopEntity/PipeEntity/GroundEntity/HitEntity
	Observer	MelonJSEngine
	Concrete Observer	ScoreItem
Strategy Pattern	Concrete Strategy	Strategy
	Context	Level1, Level2, Level3

Activity Diagram



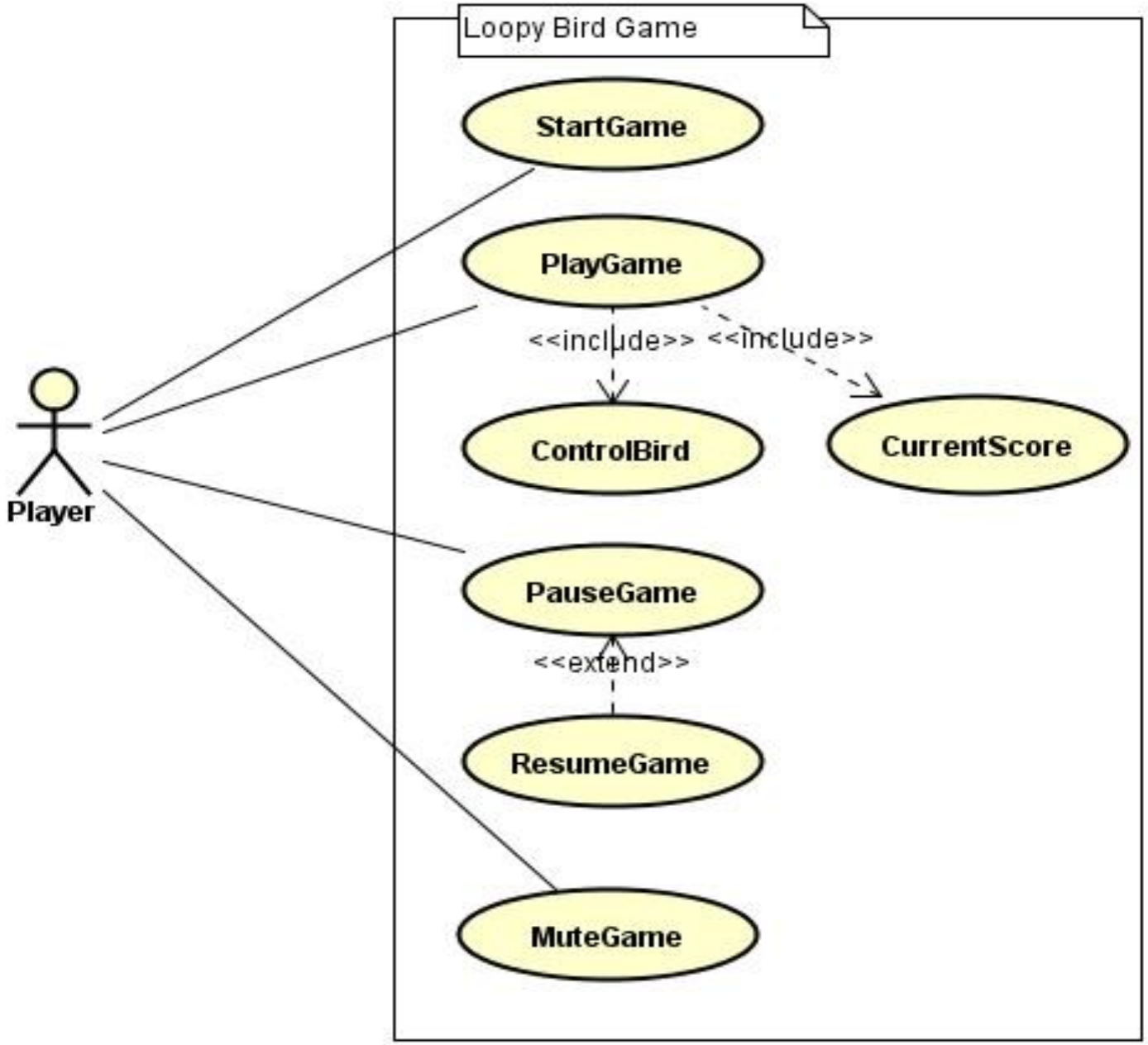
https://github.com/nguyensjsu/cmpe202-loops/blob/master/Activity%20Diagram%20Loops%20.png

Sequence Diagram



https://github.com/nguyensjsu/cmpe202-loops/tree/3ce471770673b5e09e6aefb61968234477f157ab/sequence%20diagram

Use Case Diagram



User Story-1

Story: User plays the "Loopy Bird" game

In order to beat the high score

As a Player

I want to play the game

Scenario 1: Player correctly move the bird to cross levels in the game

Given: that the game is set up

And: player is playing the game

When: player correctly move the bird in the game

And: grabs the loops or pass the pipes successfully without hitting on ground

Then: beats the game

https://github.com/nguyensjsu/cmpe202-loops/blob/master/User-Story1.png

User Story-2

Story: User plays the "Loopy Bird" game

In order to beat the high score

As a game application

I want to display screen

Scenario 1: Player correctly move the bird to cross levels in the game

Given: that the game is set up

And: player is playing the game

When: player correctly move the bird in the game

Then: loads the next screen of the stage

https://github.com/nguyensjsu/cmpe202-loops/blob/master/User-Story2.png

Use Case Specification

Use Case: Play Loopy Bird Game

Brief Description:
Player wants to play the game.

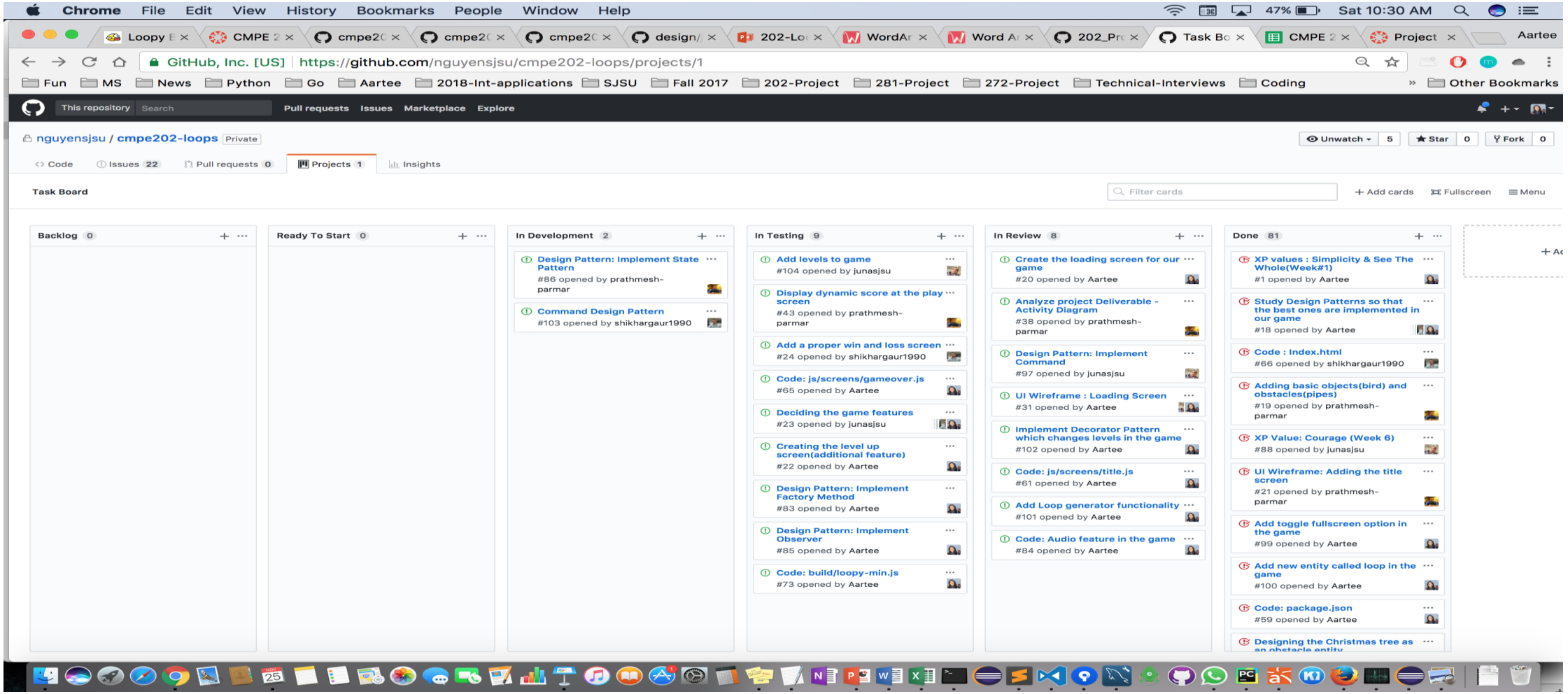
Actors: Player

Basic Flow:

I. Player starts the game by going to web link.
II. Player chooses the options from the title screen like start new game, mute the game.
III. Player presses the "Space" button on keyboard to start playing the game.
IV. In the play mode, player can use feature to mute the game or pause the game by using specified keys on the keyboard, or through mouse clicks on specified buttons on screen. Player can resume game later using "S" key.
V. Once in the play mode, player must save the bird from hitting various obstacles in the game to continue playing the game.
VI. If bird hits an obstacle, then game ends, player will be displayed the game score automatically on the screen.

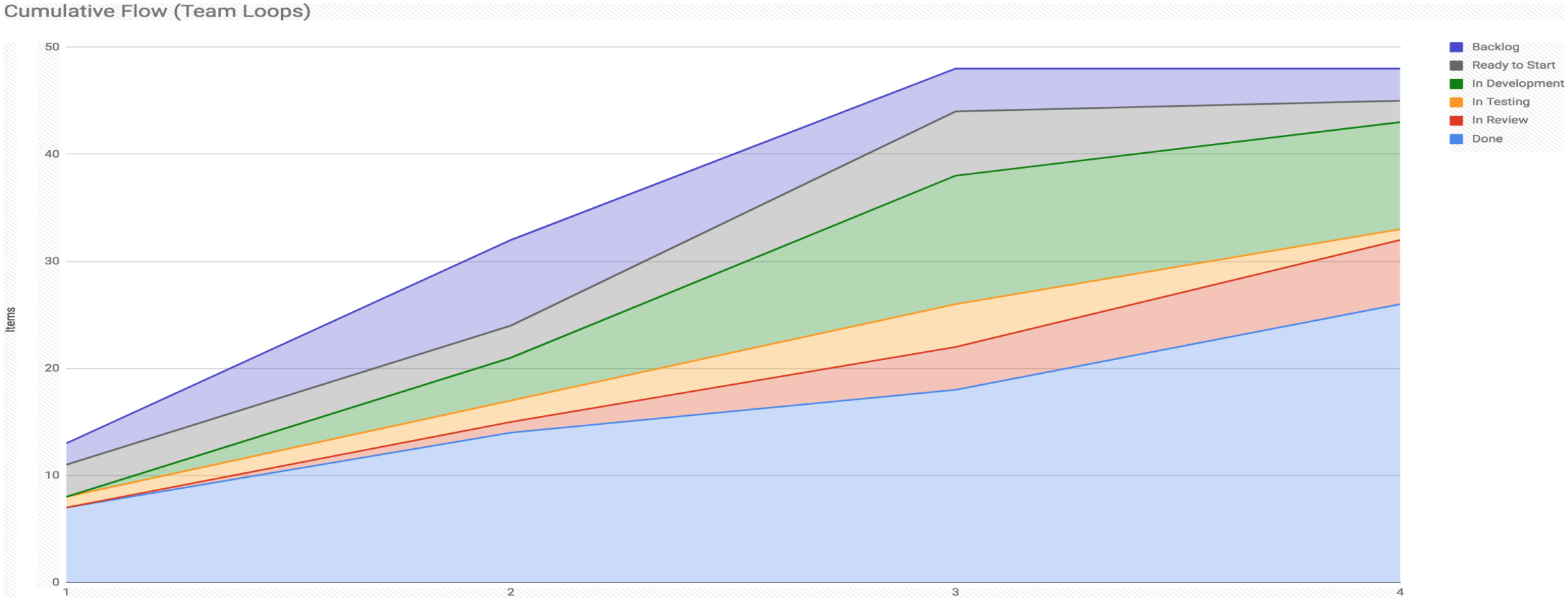
Precondition: Player must have the web browser to play Loopy bird game.

Final Task Board



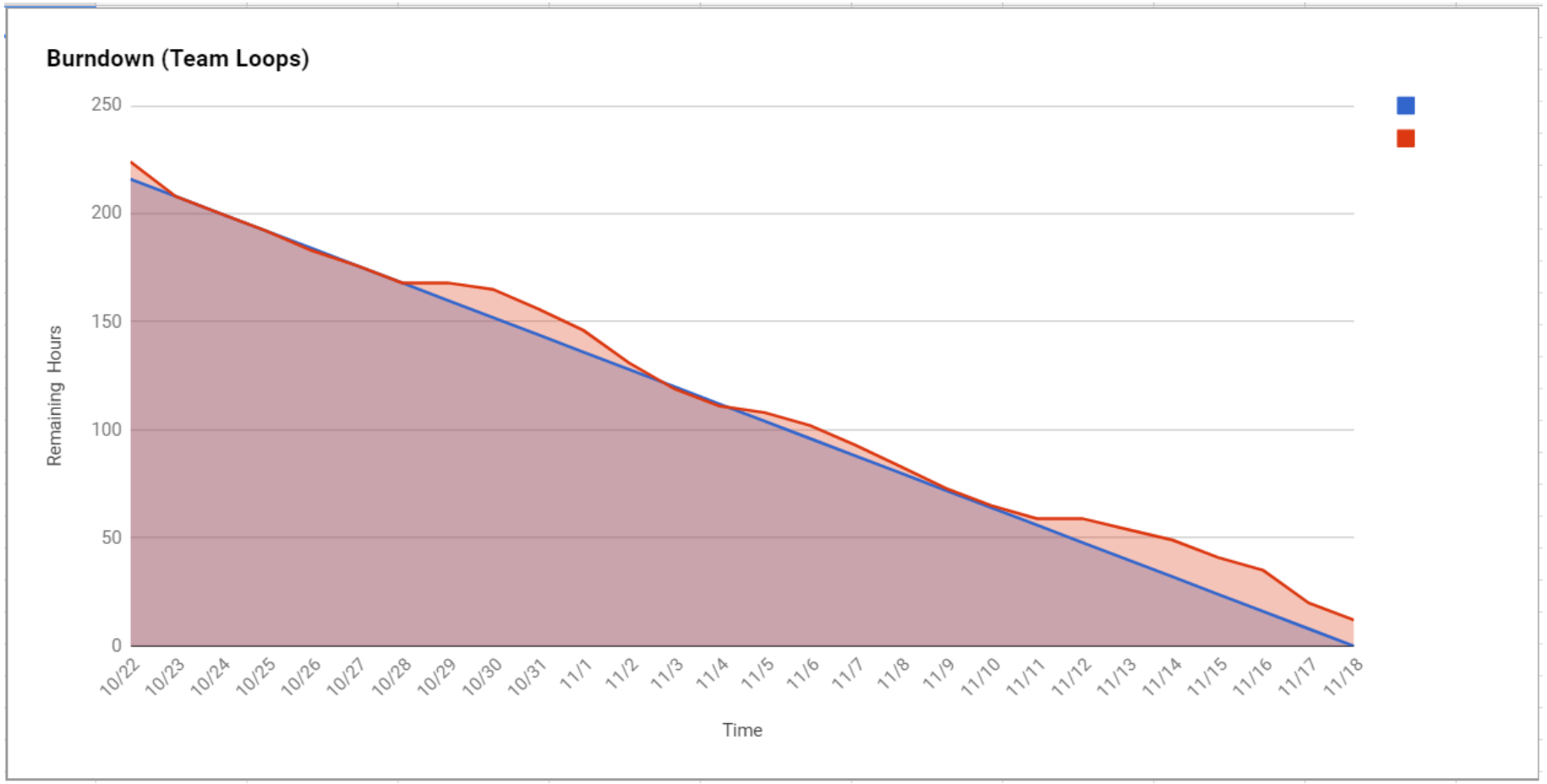
https://github.com/nguyensjsu/cmpe202-loops/projects/1

Cumulative Flow Diagram



https://docs.google.com/spreadsheets/d/1UmRQTEI2CiksiykI0AxxsgKBSQMjxMpq187mC-V3RS8/edit#gid=2

Burndown chart



https://docs.google.com/spreadsheets/d/1lMXWemXAj-7vbFc5gqzqQhYOF1j_ZS1fGVpb0VUqfik/edit#gid=0


SPRINT RETROSPECTIVE

Loop Team's Retrospective Report

Team Loops Project Retrospective Report

What did we do well?

- Regular team meetings.
- Discussing pros and cons of a game engine before using it in the game.
- Research on all the tasks for feasibility before starting implementation.
- Taking regular feedbacks from all team members.
- On-time completion of all tasks discussed in the weekly meetings.
- Asked for help from team members when stuck with any task for more than 3-4 hours.
- Good game design.

What should we have done better?

- Discuss on project requirements regularly to make sure all team members have the same understanding of the project.
- Get the code functionalities reviewed by all team members.
- Start on the UI development early in the project development.
- Proper documentation on how to run, use and test the individual game features.

What should we start doing?

- Start developing UI as soon as wireframes are ready along with backend.
- Ask 2-3 team members to review the code so that everybody is aware of the functionality and any bugs or missed requirements can be detected early.
- Ask a team member to test your functionality for all cases.
- Make sure all team members understand the requirements properly by discussing the requirements regularly.
- Document the testing and setup steps regularly.

What should we stop doing?

- Testing only the modules that a team member is responsible for implementing.
- Not adding other team members in code review.

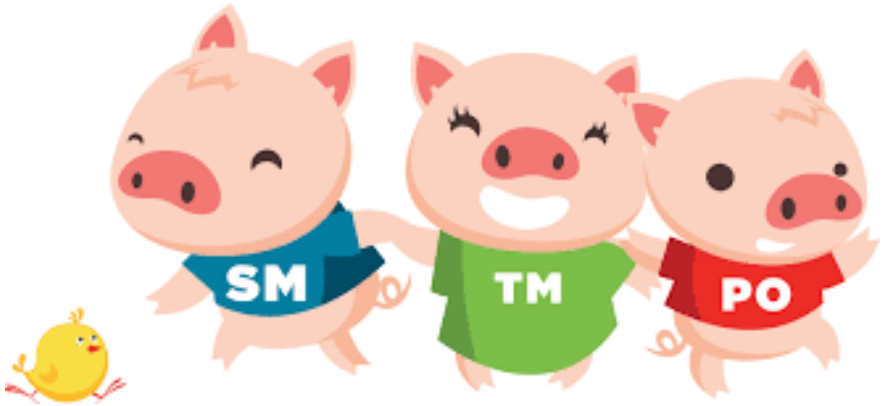
What should we keep doing?

- Help each other in a team in case a team member is stuck with any task.
- Discuss any requirements if unclear.
- Have regular team meetings and MOM to keep track of things discussed.
- Divide the tasks equally among team members.
- Complete the tasks on time as planned.

https://github.com/nguyensjsu/cmpe202-loops/blob/master/Team%20Retrospective%20Report.png

Scrum team

- Scrum master:
    - Shikhar Gaur
- Product Owner:
    - Prathmesh Parmar
- Development team:
    - Junaid Khan
    - Aartee Kasliwal



XP Core Values

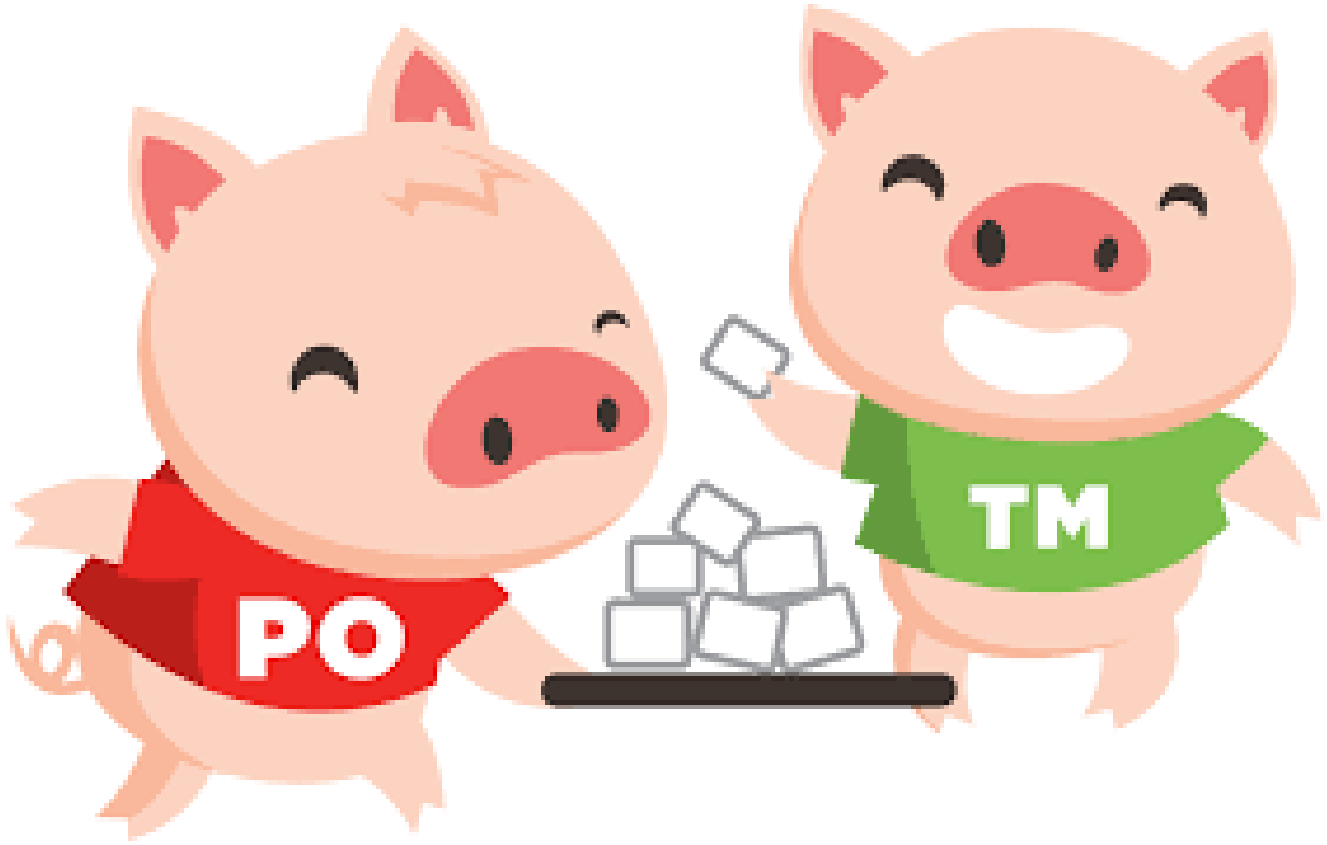
https://github.com/nguyensjsu/cmpe202-loops/blob/master/Values.md

Sprint planning

Sprint planning

PRODUCT A fully designed Loopy Bird Game using Design Patterns	SPRINT 1
GOAL: • Design front end UI design as per mockups for the Loopy Bird Game. • Implement Design Patterns in the game. • Implement score calculation feature according to levels in the game. • Implement different States in the game. • Implement different activities on different entities like bird, loop, pipe.	
METHOD: • Using MelonJS Game Engine	
METRICS: • Burndown Chart	

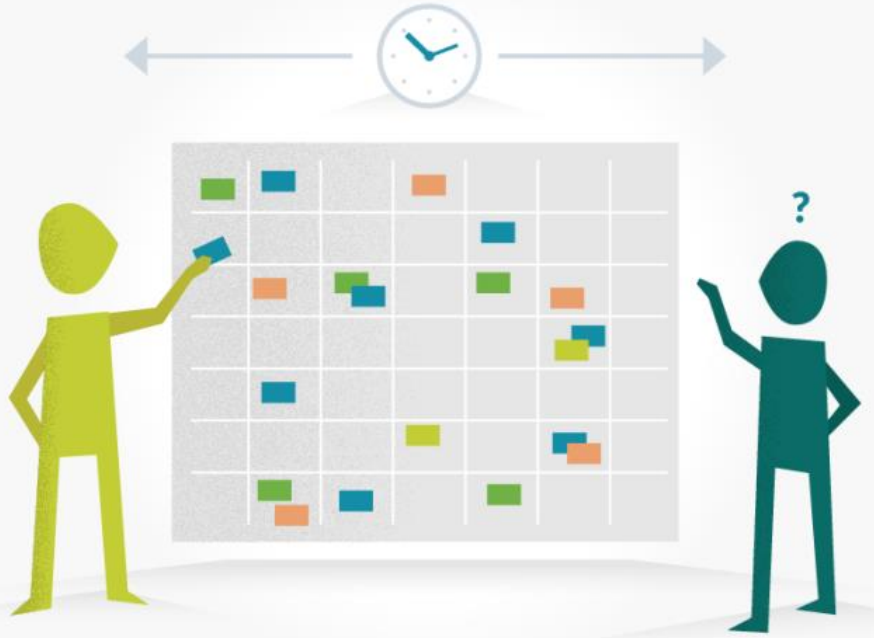
Product Backlog Refinement



Backlog Item	Task	Task Owner
Develope Game and implement Design Patterns	Implement 2 Design Patterns	Prathmesh
	Implement - Title and Loading Screen	Prathmesh
	Implement 2 Design Patterns in the game	Aartee
	Implement - Play and Gameover Screen	Aartee
	Implement 2 design patterns	Shikhar
	Implement - Levels in game	Shikhar
	Implement 2 Design Patterns	Junaid
	Entities in our Game	Junaid
Creating UML Class Diagram	Proxy, State	Prathmesh
	Observer, Factory Method	Aartee
	Command and Strategy	Shikhar
	Modular, Menu	Junaid

Backlog items

Weekly Standup



Week – 1 Standup

Loops, Sprint #1

Weekly Scrum #1 – 10/22/1017

Team Member Name: Prathmesh Parmar

What I did since the last weekly scrum:

- Started on the UI for Title Screen.
- Implemented Space button functionality for stating the game.
- Implemented Mute button functionality for Mute Sound.
- Implemented Proxy Design Pattern.

What do I plan to do this week:

- Complete the Title Screen Code.
- Complete sound implementation for Title screen.
- Implemented State Design Pattern.

What Blockers I have:

- To integrate sound in the game.
- To Mute the game on “M” button press.

Week – 2 Standup

Loops, Sprint #2

Weekly Scrum #2 – 10/29/2017

Team Member Name: Aartee Kasliwal

What I did since the last weekly scrum:

- Started on the UI for Play Screen.
- Implemented Level 1 functionality in the game.
- Implemented Observer and Factory Design Pattern.
- Created game over screen for displaying the high score.

What do I plan to do this week:

- Complete the Play Screen Code.
- Complete sound implementation for Play screen.
- Start implementing Level 2 in the game.

What Blockers I have:

- To integrate sound in the game.
- To notify observers of their respective high scores.

Week – 3 Standup

Loops, Sprint #3

Weekly Scrum #3 – 10/05/2017

Team Member Name: Shikhar Gaur

What I did since the last weekly scrum:

- Started on the UI for Play Screen.
- Implemented Level 3 functionality in the game.
- Implemented Command Design Pattern.
- Created game Pause and Resume functionality.

What do I plan to do this week:

- Complete the Play Screen Code.
- Implement Strategy Design Pattern.

What Blockers I had:

- Implementing pause/resume functionality
- Implementing mouse control is UI
- to invoke the assigned commands

Week – 4 Standup

Loops, Sprint #4

Weekly Scrum #4 – 10/12/2017

Team Member Name: Junaid Khan

What I did since the last weekly scrum:

- Started on the UI for Game Over Screen.
- Implemented Menu Design Pattern.
- Created game Pause and Resume functionality.

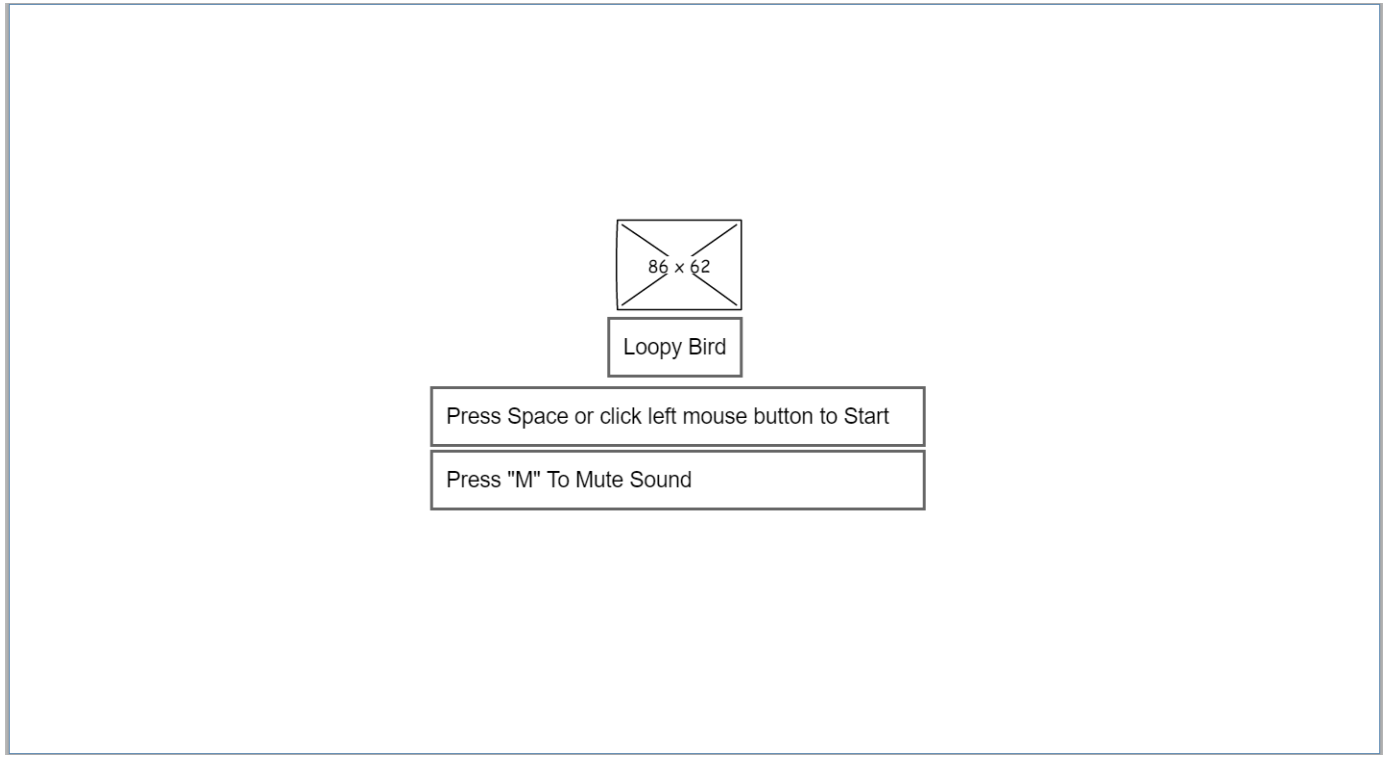
What do I plan to do this week:

- Complete the Game Over Screen Code.
- Restarting the game once it's over.

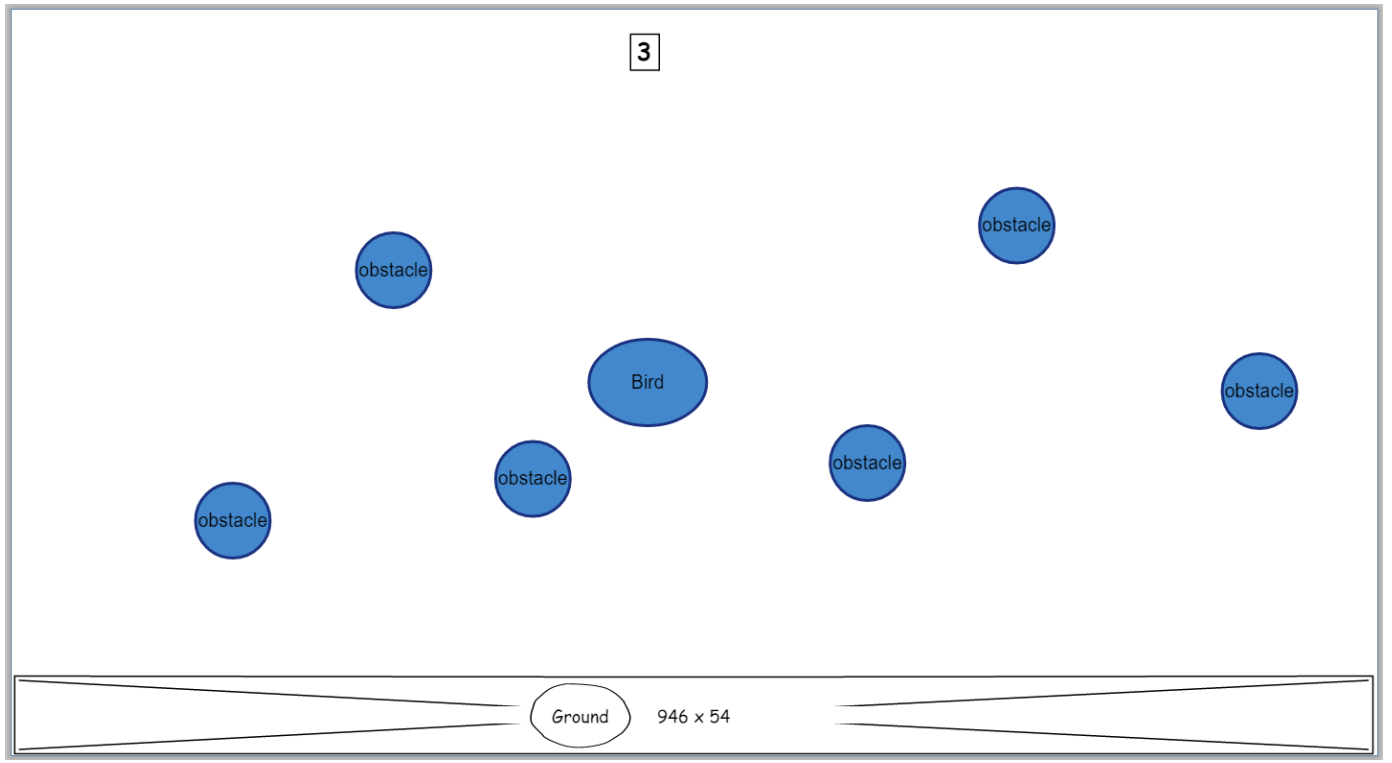
What Blockers I had:

- Displaying high score in game over screen.

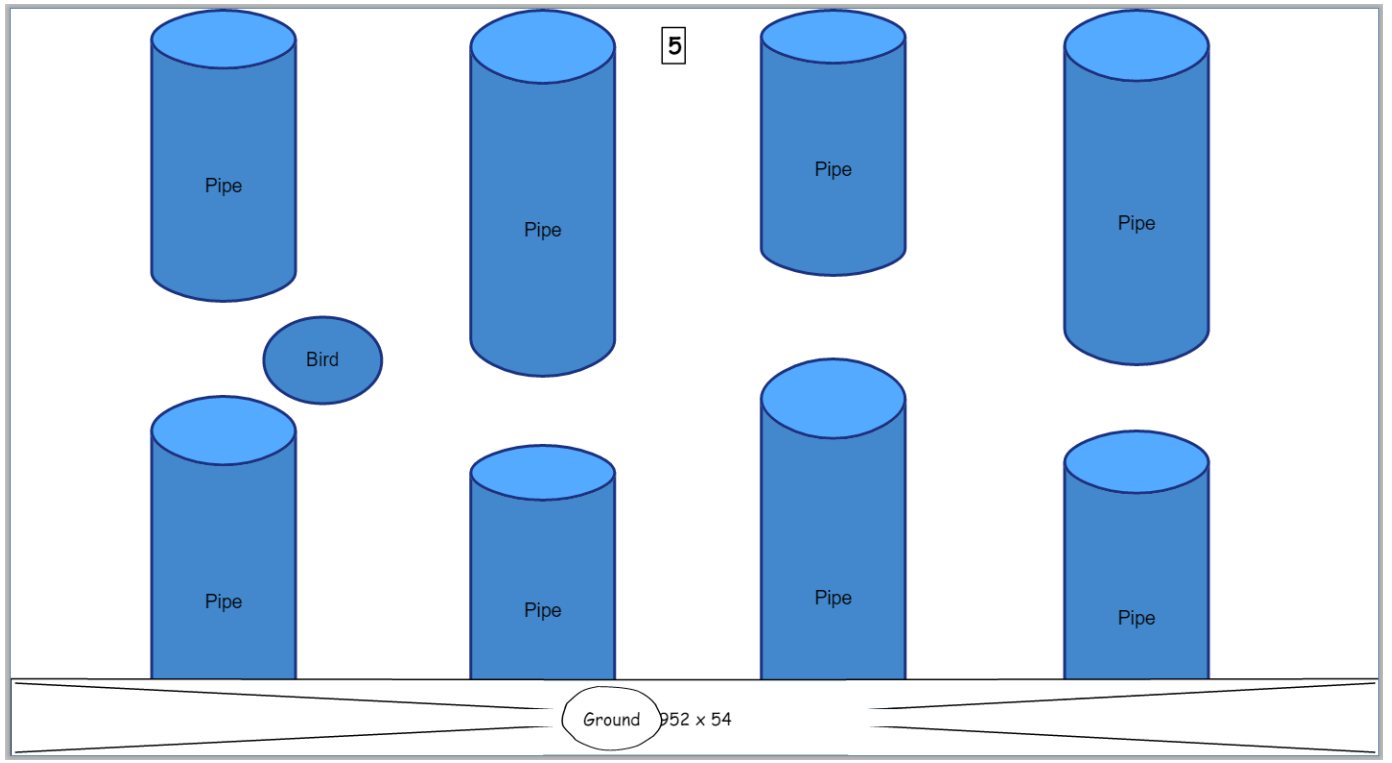
UI Wire Frames- Title Screen



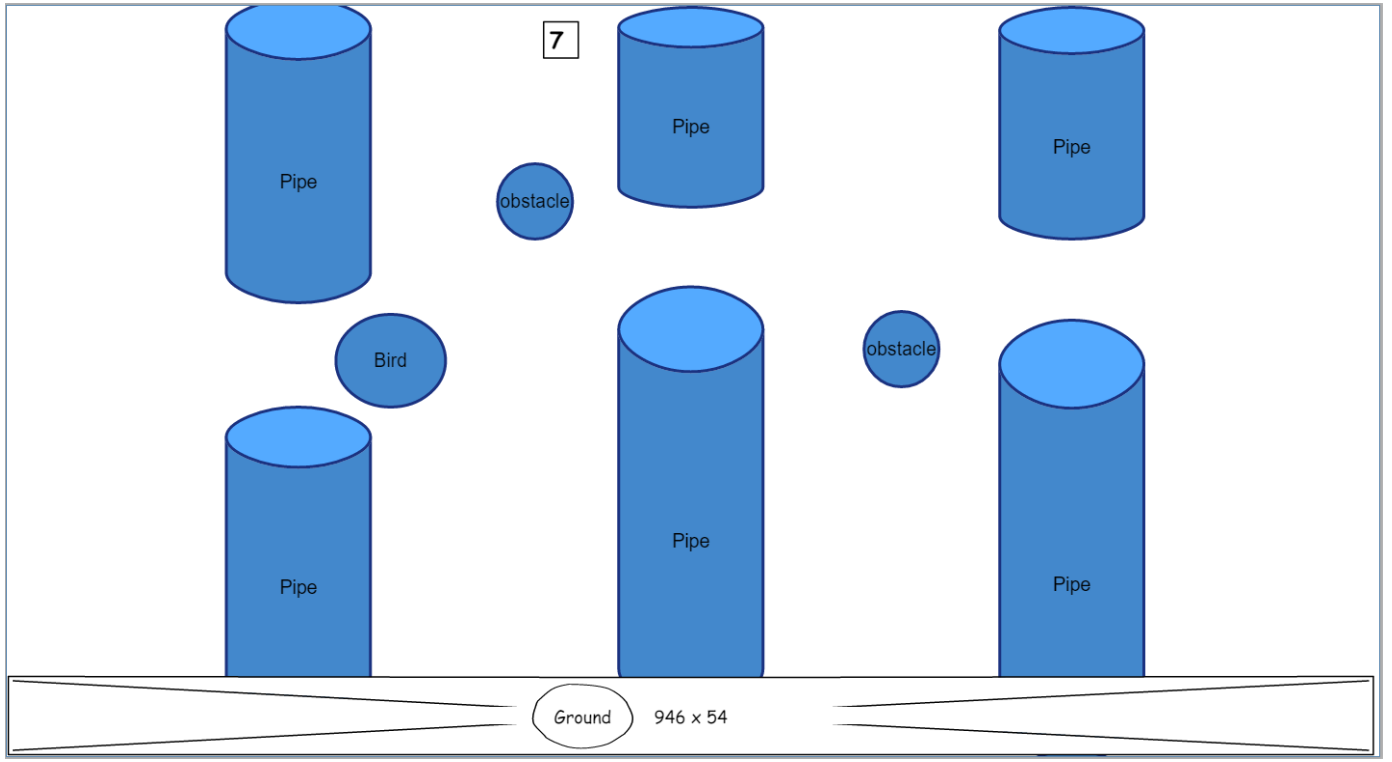
UI Wire Frames- Play Screen



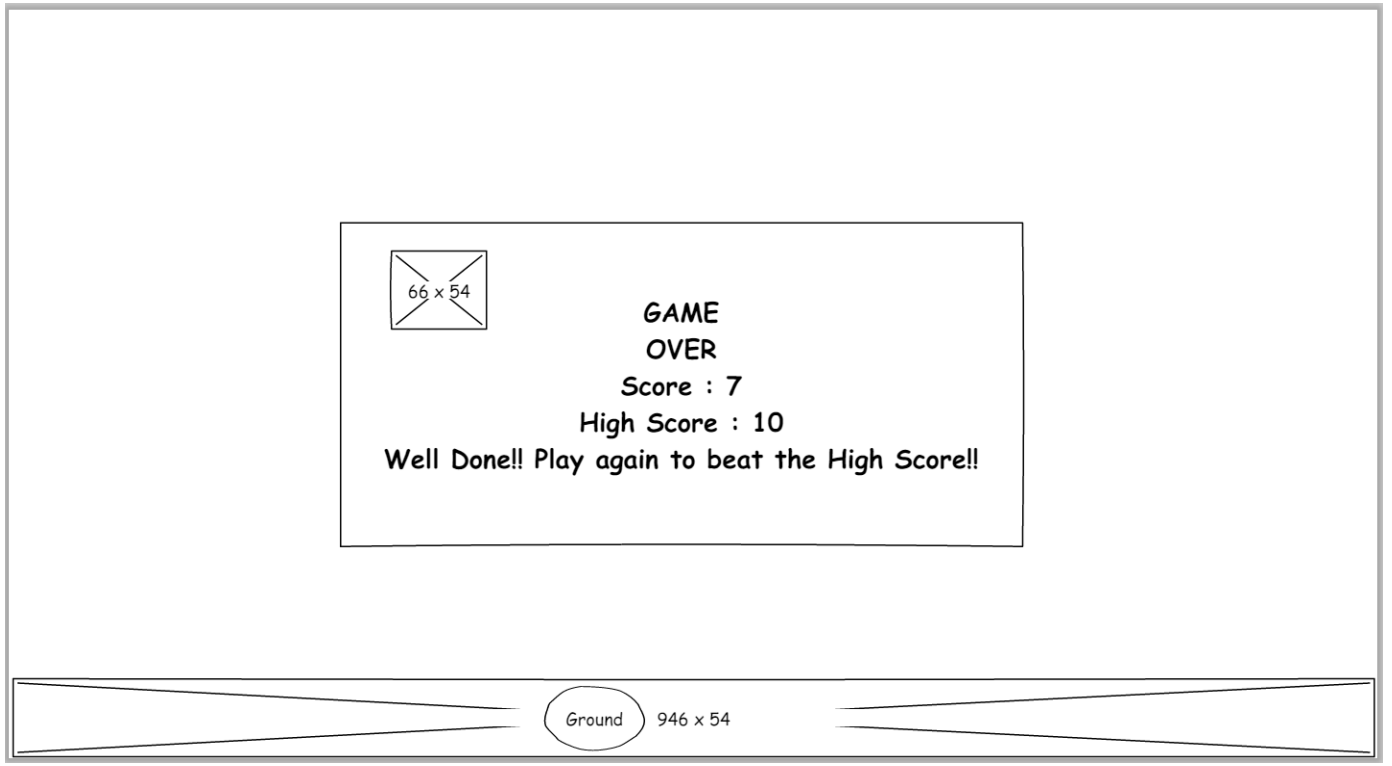
UI Wire Frames- Play Screen

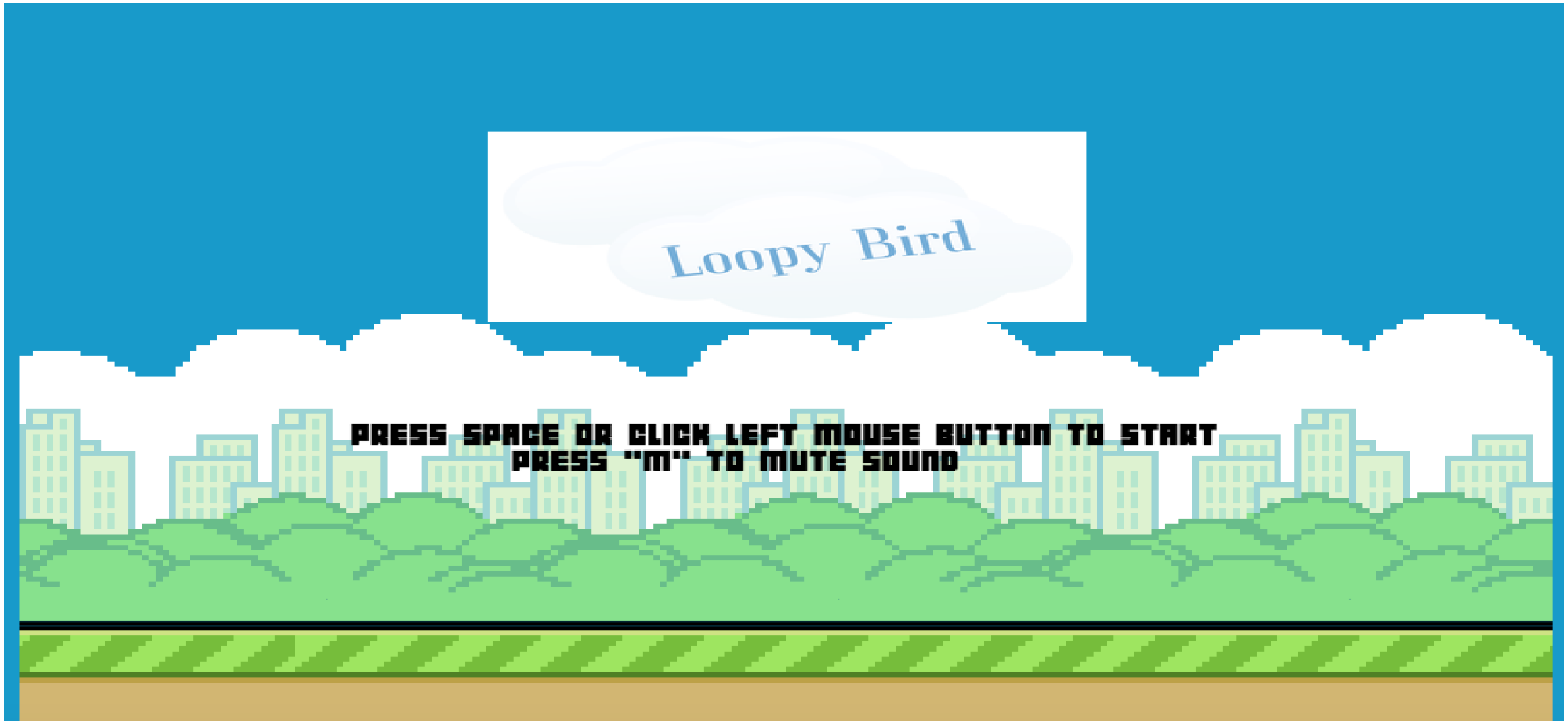


UI Wire Frames- Play Screen



UI Wire Frames- Game Over

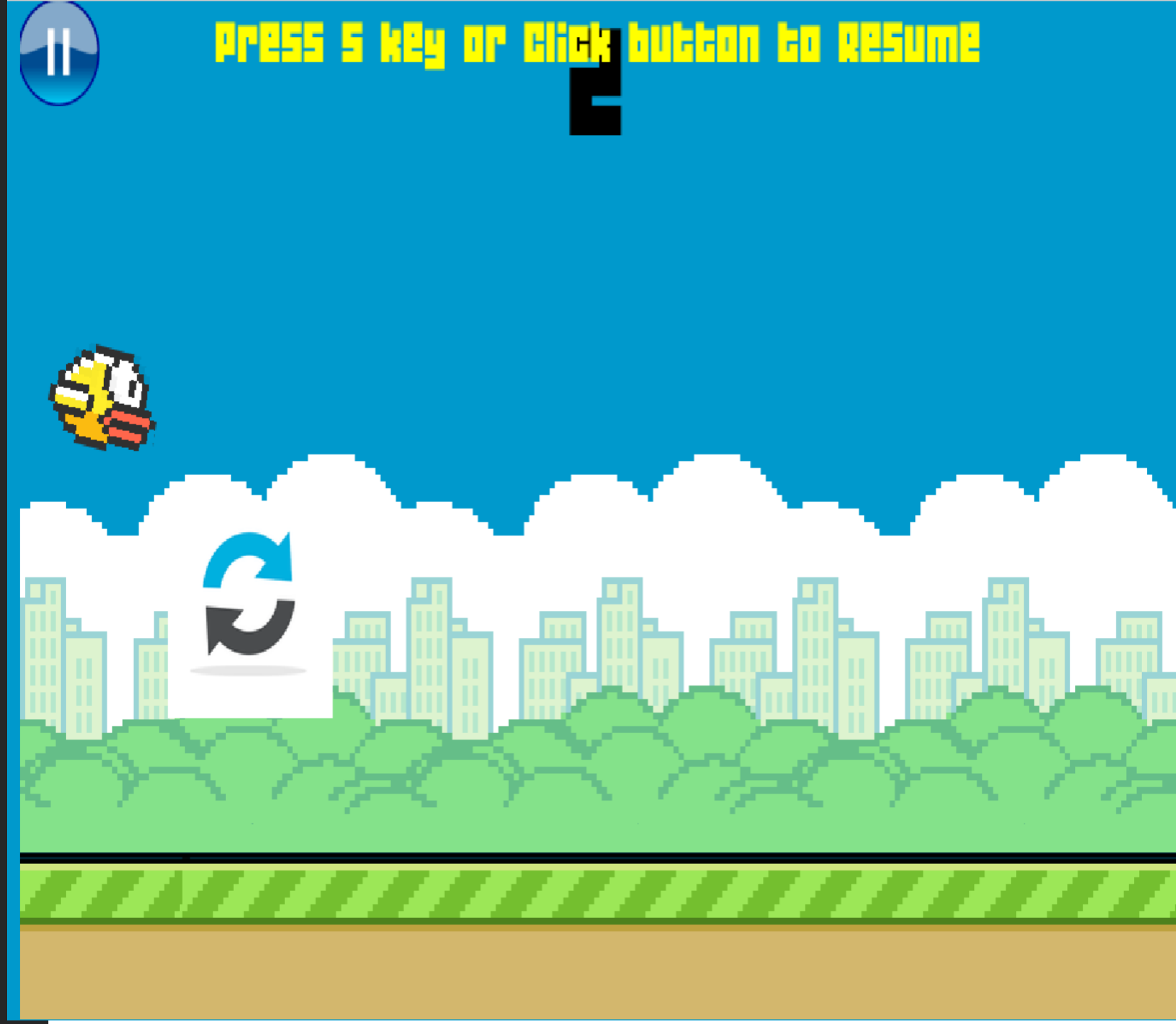


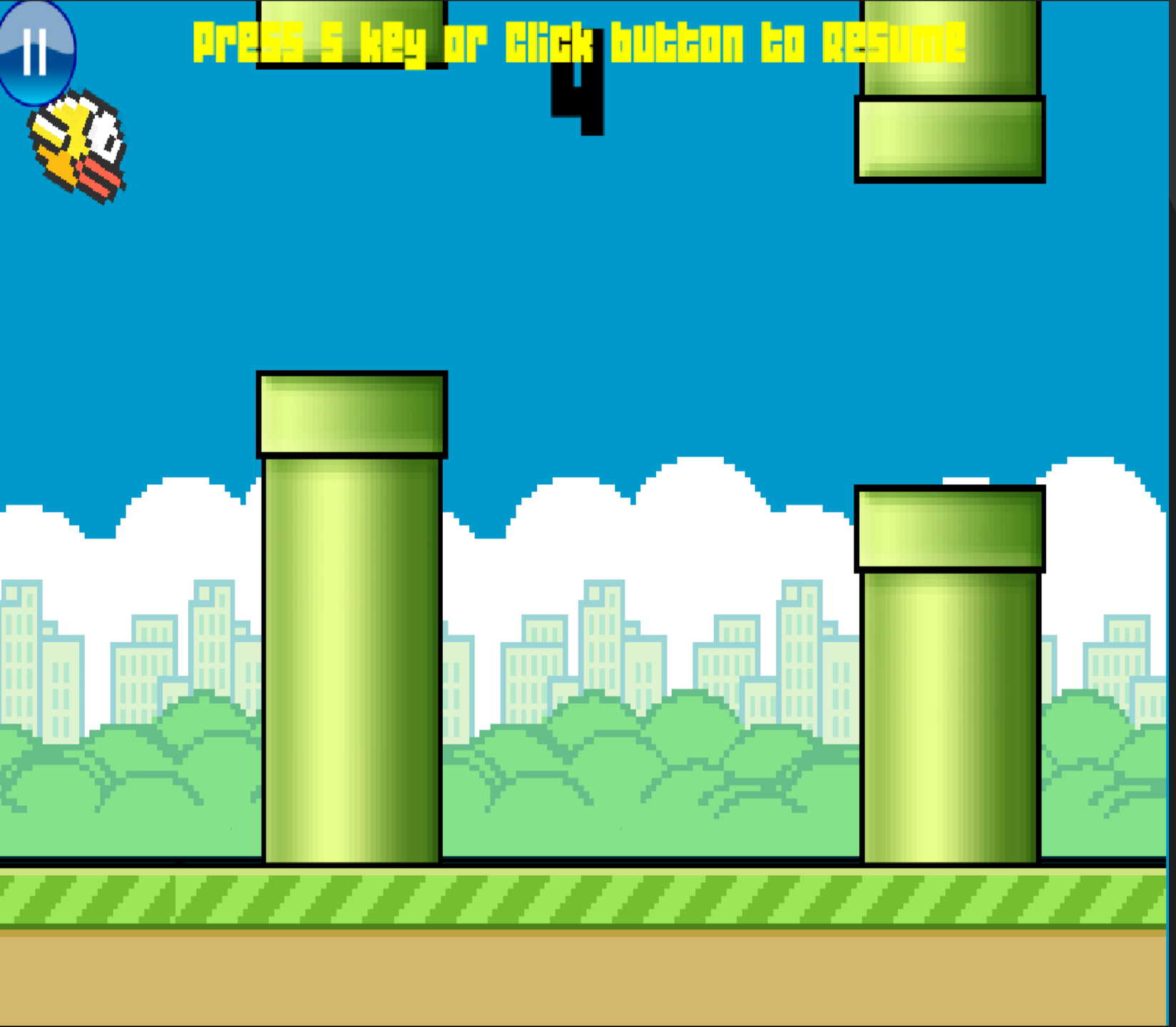


The Loopy Bird Game and the whole MelonJS world..

Level 1

- Press space key to keep the bird flying
- If bird touches the ground, the game is over
- Player needs to collect loops to increase the score





Level 2

- After you achieve certain score in level 1, you enter level 2.
- Keep flying by making sure you don't touch any of the pipes or the ground.
- Your score keeps on incrementing when you pass through a pipe.

Level 3

- This is the final stage in the game.
- Collect the loops while carefully flying through the pipes.
- The score increases when you collect a loop and pass through a pipe

Game Over Screen

- Oops! You hit the bird. If you get a high score, it will be displayed here. Press space to start all over again.