



**LOOPY BIRD**



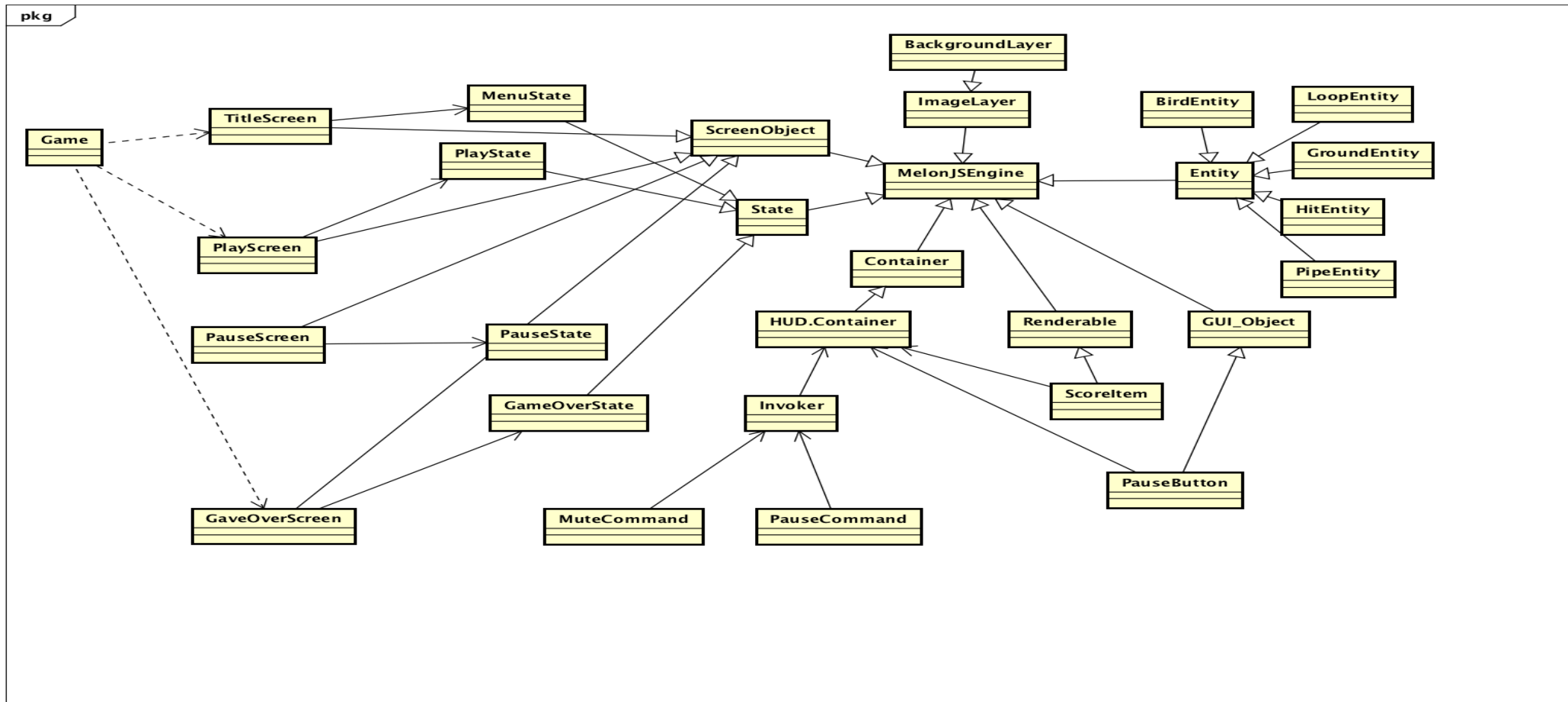
# Loops

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



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



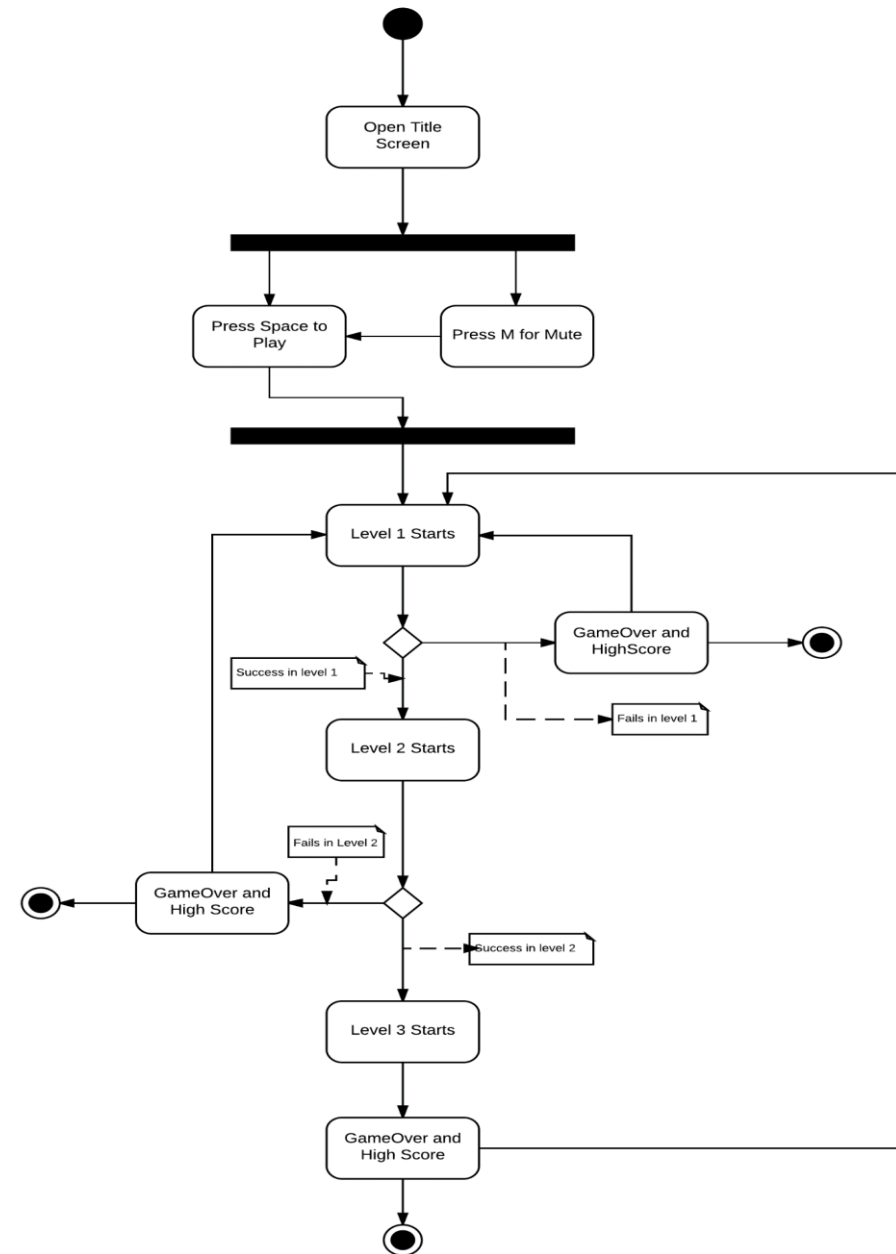


# Class Diagram: Objects inherited from MelonJS Engine

# Design Patterns Implemented in the Game

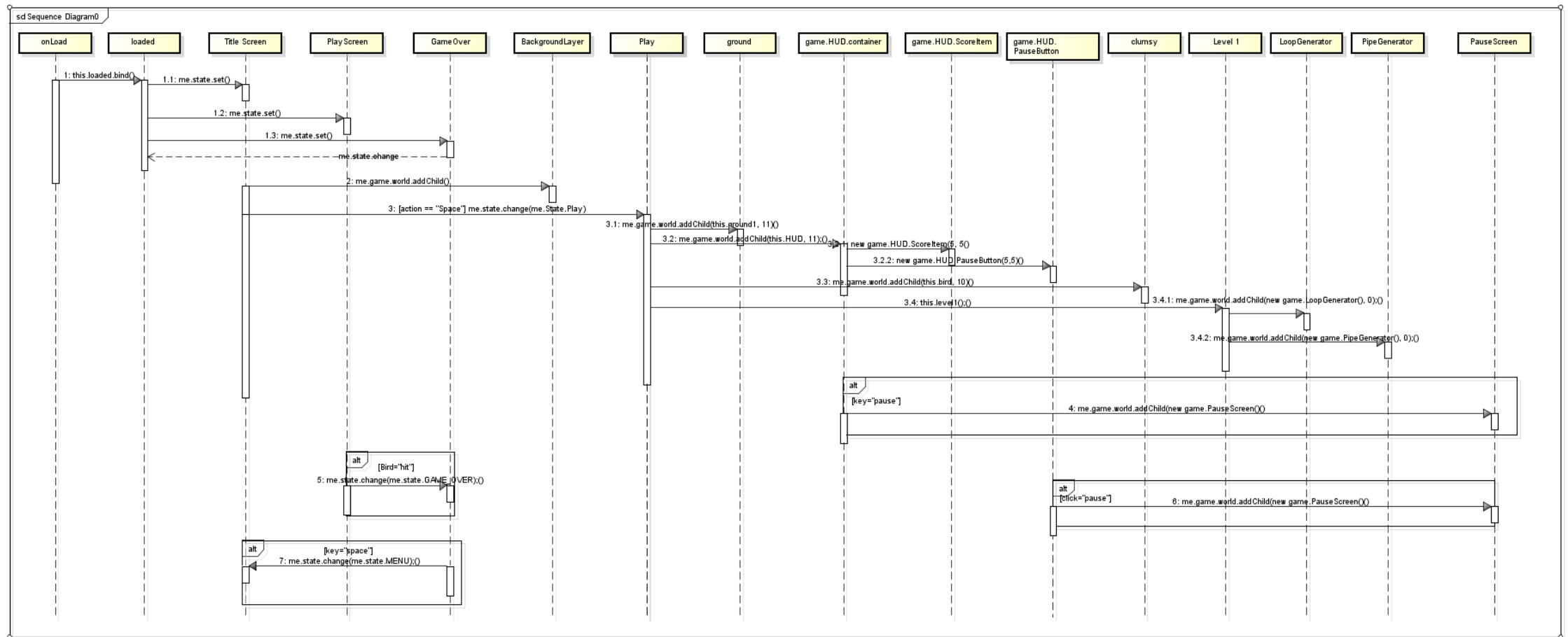
Design Pattern Name		Class/Interface
Command Pattern	Concrete Command	MuteCommand, PauseCommand
	Client	HUD.Container
	Invoker	Invoker
	Receiver	MelonJSEngine
State Pattern	Context	State
	States	MenuState/PlayState/PauseState/GameOverState
Composite Pattern	Client	Game
	Component	ScreenObject
	Leaf	TitleScreen/PlayScreen/PauseScreen/GameOverScreen
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>

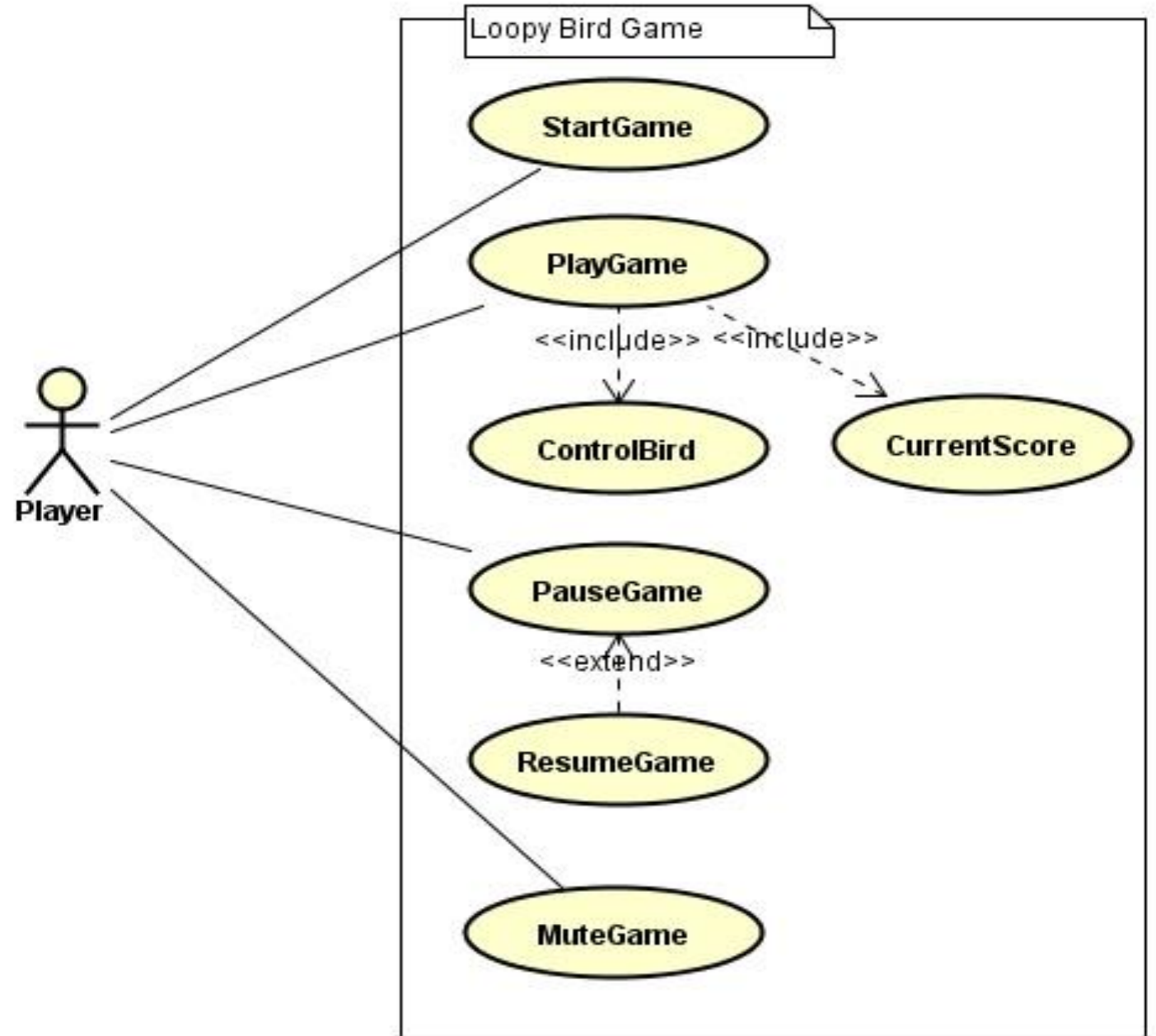
# Sequence Diagram



<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

<b>Use Case:</b> Play Loopy Bird Game	
<b>Brief Description:</b> Player wants to play the game.	
<b>Actors:</b> Player	
<b>Basic Flow:</b> <ul style="list-style-type: none"><li>I. Player starts the game by going to web link.</li><li>II. Player chooses the options from the title screen like start new game, mute the game.</li><li>III. Player presses the "Space" button on keyboard to start playing the game.</li><li>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.</li><li>V. Once in the play mode, player must save the bird from hitting various obstacles in the game to continue playing the game.</li><li>VI. If bird hits an obstacle, then game ends, player will be displayed the game score automatically on the screen.</li></ul>	
<b>Precondition:</b> Player must have the web browser to play Loopy bird game.	

# Final Task Board

The screenshot shows a web browser window displaying the GitHub repository page for 'nguyensjsu / cmpe202-loops'. The browser's address bar shows the URL 'https://github.com/nguyensjsu/cmpe202-loops/projects/1'. The repository page includes tabs for 'Code', 'Issues', 'Pull requests', 'Projects', and 'Insights'. The 'Projects' tab is active, showing a 'Task Board' with columns for different stages of development: 'Backlog', 'Ready To Start', 'In Development', 'In Testing', 'In Review', and 'Done'. Each column contains a list of tasks, each represented by a card with a title, a number, and the user who opened it. The 'Done' column is the largest, containing 81 tasks. The browser's top bar shows the time as 'Sat 10:30 AM' and the battery level as '47%'. The bottom of the screen shows a macOS dock with various application icons.

Chrome File Edit View History Bookmarks People Window Help

Loopy E x CMPE 2 x cmpe20 x cmpe20 x cmpe20 x design/ x 202-Lo x WordAr x Word Ar x 202\_Prc x Task Bo x CMPE 2 x Project x Aartee

GitHub, Inc. [US] | <https://github.com/nguyensjsu/cmpe202-loops/projects/1>

Fun MS News Python Go Aartee 2018-Int-applications SJSU Fall 2017 202-Project 281-Project 272-Project Technical-Interviews Coding Other Bookmarks

This repository Search Pull requests Issues Marketplace Explore

nguyensjsu / cmpe202-loops Private

Unwatch 5 Star 0 Fork 0

< Code Issues 22 Pull requests 0 Projects 1 Insights

Task Board

Filter cards + Add cards Fullscreen Menu

Backlog 0 + ...

Ready To Start 0 + ...

In Development 2 + ...

In Testing 9 + ...

In Review 8 + ...

Done 81 + ...

Design Pattern: Implement State Pattern #86 opened by prathmesh-parmar

Command Design Pattern #103 opened by shikhargaur1990

Add levels to game #104 opened by junasjsu

Display dynamic score at the play screen #43 opened by prathmesh-parmar

Add a proper win and loss screen #24 opened by shikhargaur1990

Code: js/screens/gameover.js #65 opened by Aartee

Deciding the game features #23 opened by junasjsu

Creating the level up screen(additional feature) #22 opened by Aartee

Design Pattern: Implement Factory Method #83 opened by Aartee

Design Pattern: Implement Observer #85 opened by Aartee

Code: build/loopy-min.js #73 opened by Aartee

Create the loading screen for our game #20 opened by Aartee

Analyze project Deliverable - Activity Diagram #38 opened by prathmesh-parmar

Design Pattern: Implement Command #97 opened by junasjsu

UI Wireframe : Loading Screen #31 opened by Aartee

Implement Decorator Pattern which changes levels in the game #102 opened by Aartee

Code: js/screens/title.js #61 opened by Aartee

Add Loop generator functionality #101 opened by Aartee

Code: Audio feature in the game #84 opened by Aartee

XP values : Simplicity & See The Whole(Week#1) #1 opened by Aartee

Study Design Patterns so that the best ones are implemented in our game #18 opened by Aartee

Code : Index.html #66 opened by shikhargaur1990

Adding basic objects(bird) and obstacles(pipes) #19 opened by prathmesh-parmar

XP Value: Courage (Week 6) #88 opened by junasjsu

UI Wireframe: Adding the title screen #21 opened by prathmesh-parmar

Add toggle fullscreen option in the game #99 opened by Aartee

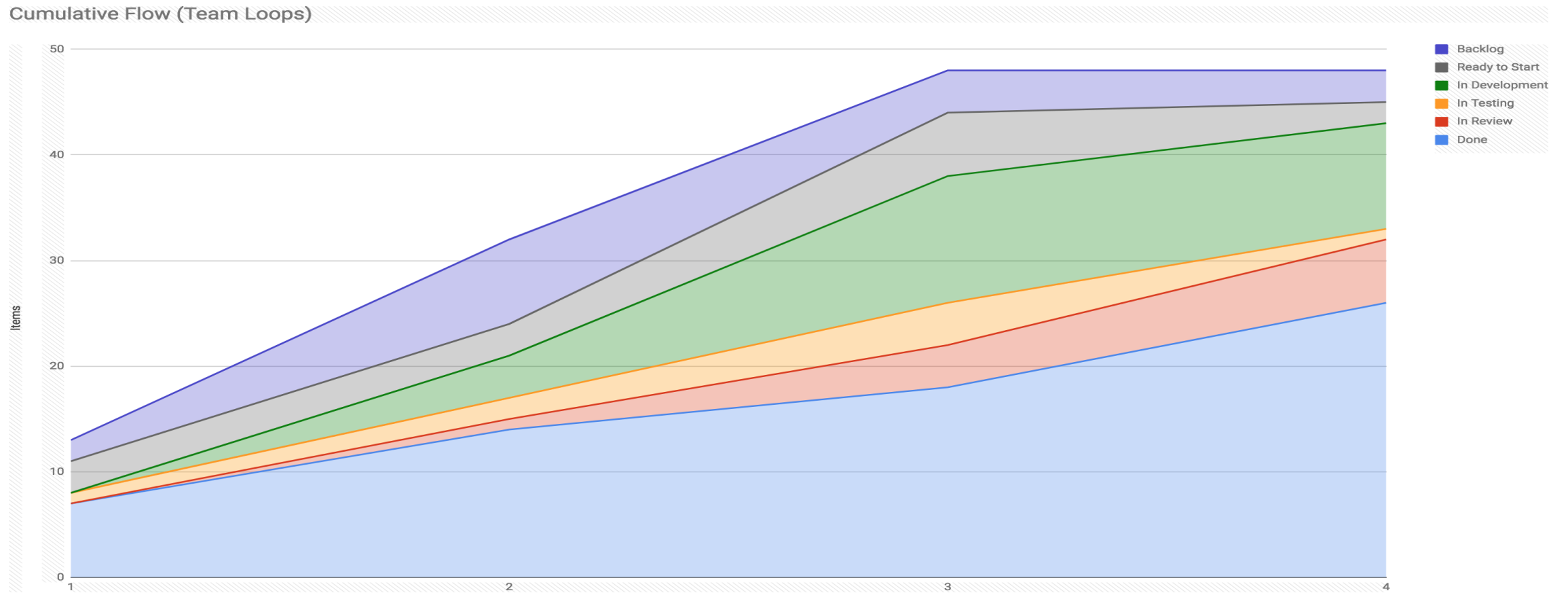
Add new entity called loop in the game #100 opened by Aartee

Code: package.json #59 opened by Aartee

Designing the Christmas tree as an obstacle entity

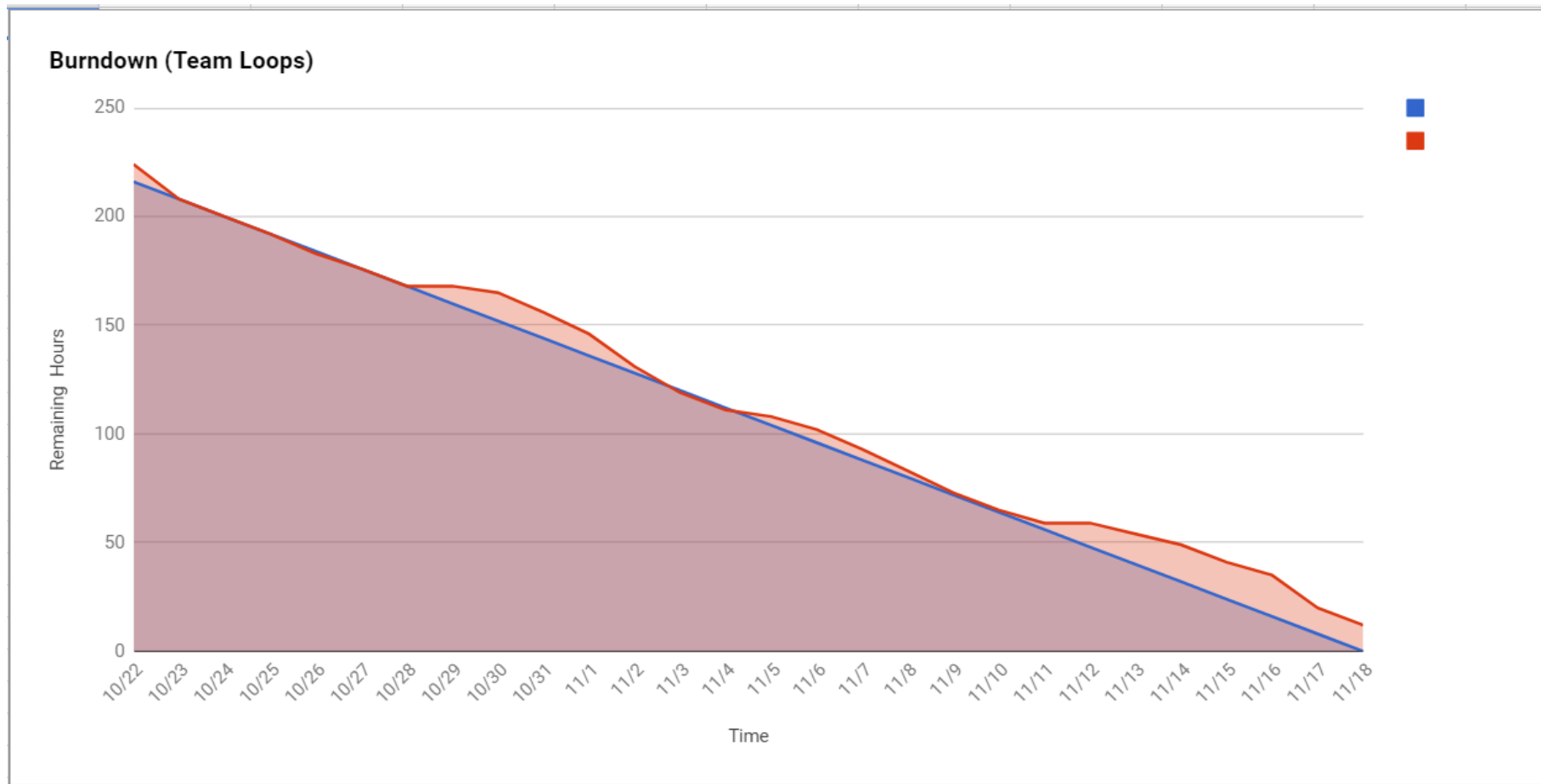
<https://github.com/nguyensjsu/cmpe202-loops/projects/1>

# Cumulative Flow Diagram



<https://docs.google.com/spreadsheets/d/1UmRQTEI2Ciksiykl0AxxsgKBSQMjxMpq187mC-V3RS8/edit#gid=2>

# Burndown chart



[https://docs.google.com/spreadsheets/d/1IMXWemXAJ-7vbFc5gqzqQhYOF1j\\_ZS1fGVpb0VUqfik/edit#gid=0](https://docs.google.com/spreadsheets/d/1IMXWemXAJ-7vbFc5gqzqQhYOF1j_ZS1fGVpb0VUqfik/edit#gid=0)



The illustration shows a top-down view of six team members seated around a large, dark brown rectangular table. The team consists of three men and three women, each with a distinctively colored head (brown, orange, or red). They are all wearing dark blue business suits. The team is arranged in two rows of three. The top row (from the viewer's perspective) includes a man with a brown head using a tablet, a man with a brown head using a laptop, and a woman with an orange head reviewing a stack of papers. The bottom row includes a woman with a brown head holding a newspaper, a man with a brown head writing on a notepad, and a woman with a red head writing in a notebook. Various office items are scattered on the table: a coffee cup, a pencil, a watch, and a stapler. The background is a solid teal color.

# SPRINT RETROSPECTIVE



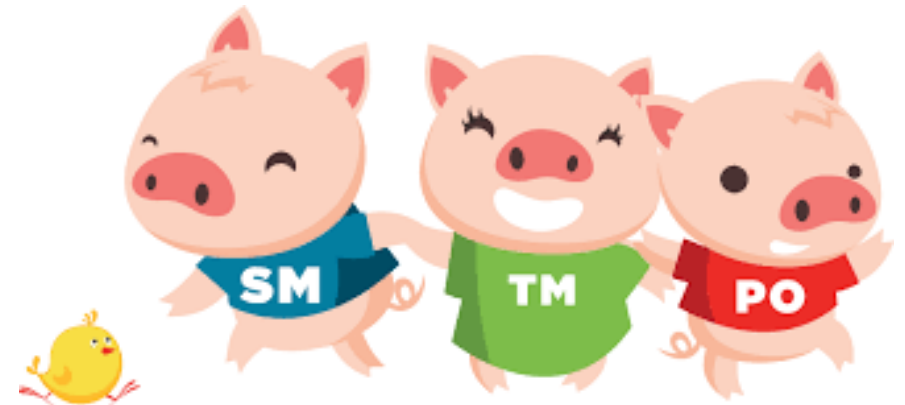
# Loop Team's Retrospective Report

Team Loops Project Retrospective Report		
<b>What did we do well?</b>	<b>What should we have done better?</b>	
<ul style="list-style-type: none"><li>• Regular team meetings.</li><li>• Discussing pros and cons of a game engine before using it in the game.</li><li>• Research on all the tasks for feasibility before starting implementation.</li><li>• Taking regular feedbacks from all team members.</li><li>• On-time completion of all tasks discussed in the weekly meetings.</li><li>• Asked for help from team members when stuck with any task for more than 3-4 hours.</li><li>• Good game design.</li></ul>	<ul style="list-style-type: none"><li>• Discuss on project requirements regularly to make sure all team members have the same understanding of the project.</li><li>• Get the code functionalities reviewed by all team members.</li><li>• Start on the UI development early in the project development.</li><li>• Proper documentation on how to run, use and test the individual game features.</li></ul>	
<b>What should we start doing?</b>	<b>What should we stop doing?</b>	<b>What should we keep doing?</b>
<ul style="list-style-type: none"><li>• Start developing UI as soon as wireframes are ready along with backend.</li><li>• 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.</li><li>• Ask a team member to test your functionality for all cases.</li><li>• Make sure all team members understand the requirements properly by discussing the requirements regularly.</li><li>• Document the testing and setup steps regularly.</li></ul>	<ul style="list-style-type: none"><li>• Testing only the modules that a team member is responsible for implementing.</li><li>• Not adding other team members in code review.</li></ul>	<ul style="list-style-type: none"><li>• Help each other in a team in case a team member is stuck with any task.</li><li>• Discuss any requirements if unclear.</li><li>• Have regular team meetings and MOM to keep track of things discussed.</li><li>• Divide the tasks equally among team members.</li><li>• Complete the tasks on time as planned.</li></ul>

<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	SPRINT 1
A fully designed Loopy Bird Game using Design Patterns	
<b>GOAL:</b> <ul style="list-style-type: none"><li>• Design front end UI design as per mockups for the Loopy Bird Game.</li><li>• Implement Design Patterns in the game.</li><li>• Implement score calculation feature according to levels in the game.</li><li>• Implement different States in the game.</li><li>• Implement different activities on different entities like bird, loop, pipe.</li></ul>	
<b>METHOD:</b> <ul style="list-style-type: none"><li>• Using MelonJS Game Engine</li></ul>	
<b>METRICS:</b> <ul style="list-style-type: none"><li>• Burndown Chart</li></ul>	

## Product Backlog Refinement



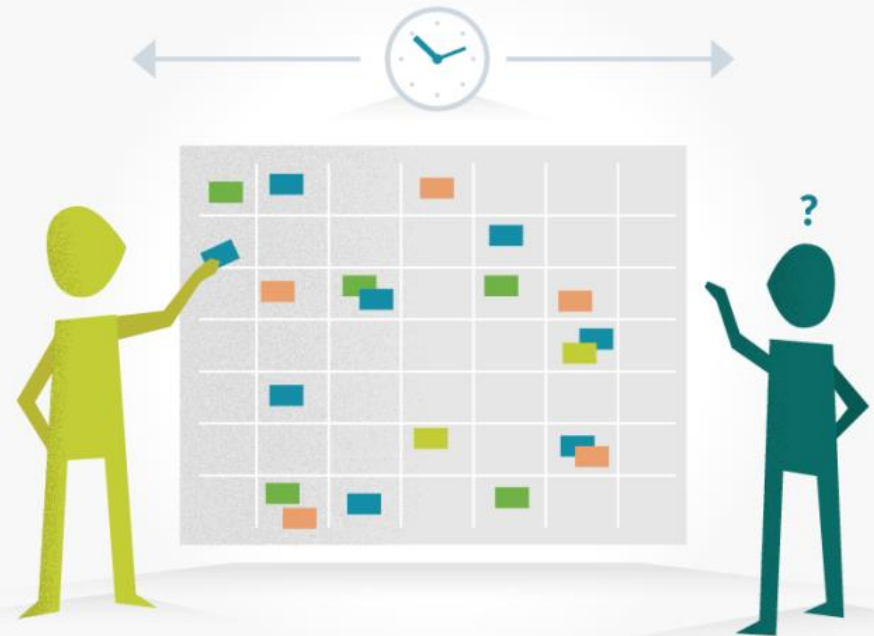
Backlog Item	Task	Task Owner
Develope Game and implement Design Patterns	Implement 2 Design Patterns	Prathmesh
	Implement - Title and Loading Screen	Prathmesh
	Implement 2 Design Patterns in the game	Aartee
	Implement - Play and Gameover Screen	Aartee
	Implement 2 design patterns	Shikhar
	Implement - 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:

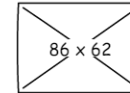
- Complete the Game Over Screen Code.
- Restarting the game once it's over.

What Blockers I had:

- Displaying high score in game over screen.

# UI Wire Frames- Title Screen

---

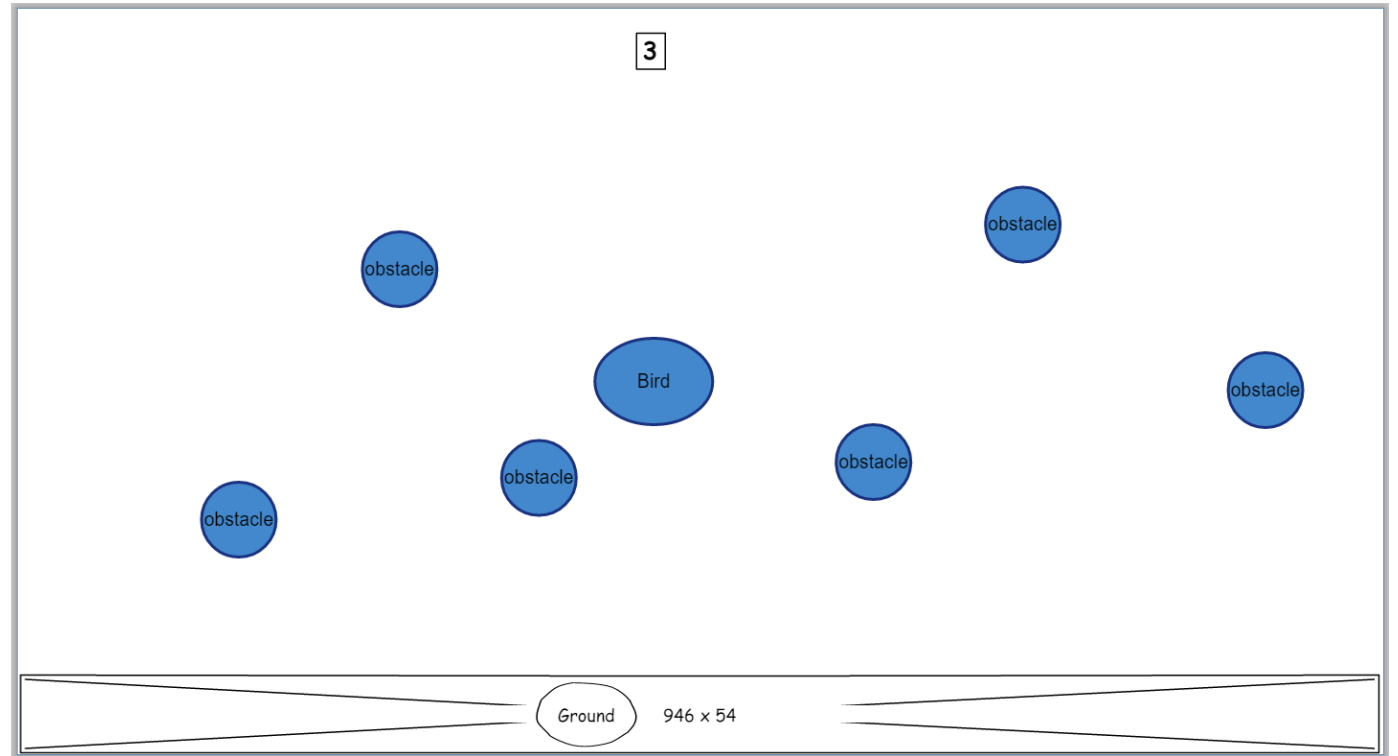


Loopy Bird

Press Space or click left mouse button to Start

Press "M" To Mute Sound

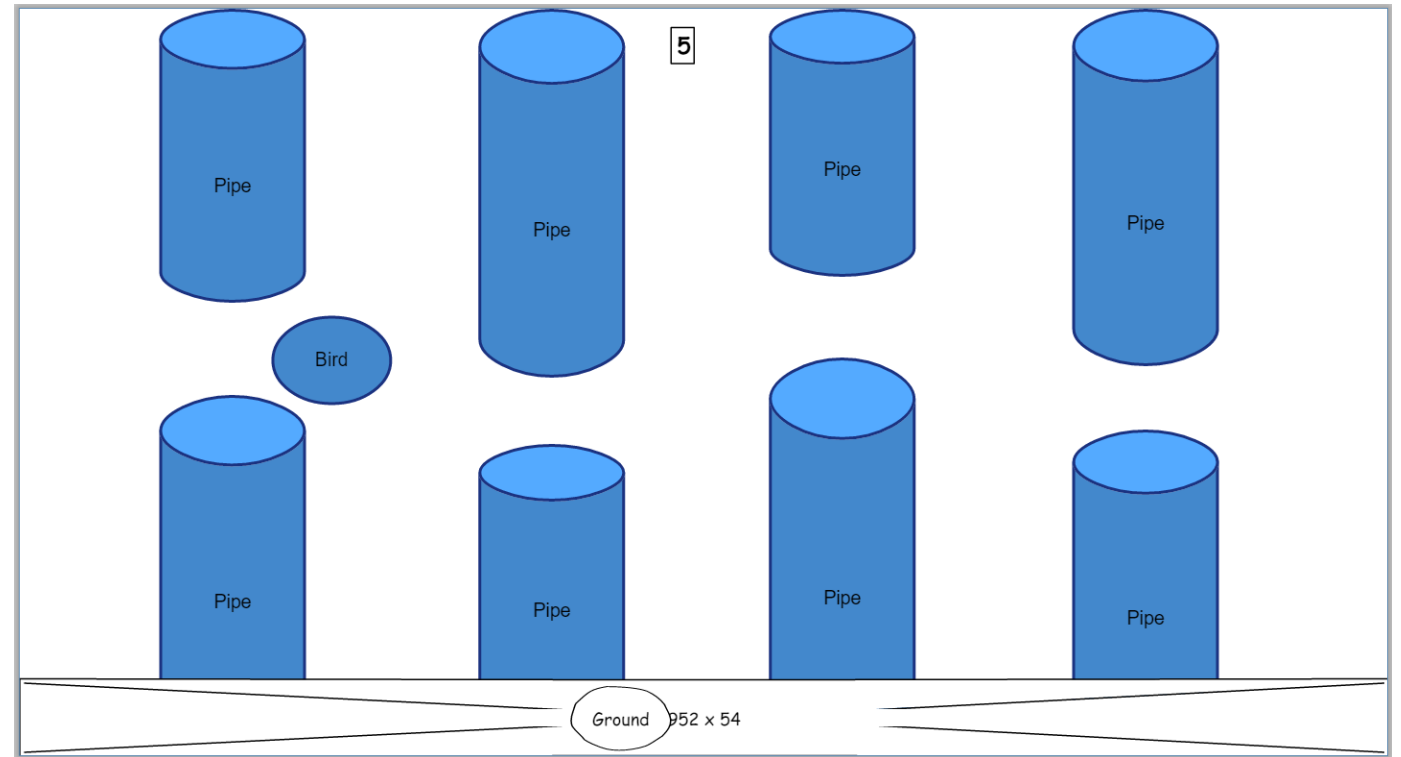
# UI Wire Frames- Play Screen





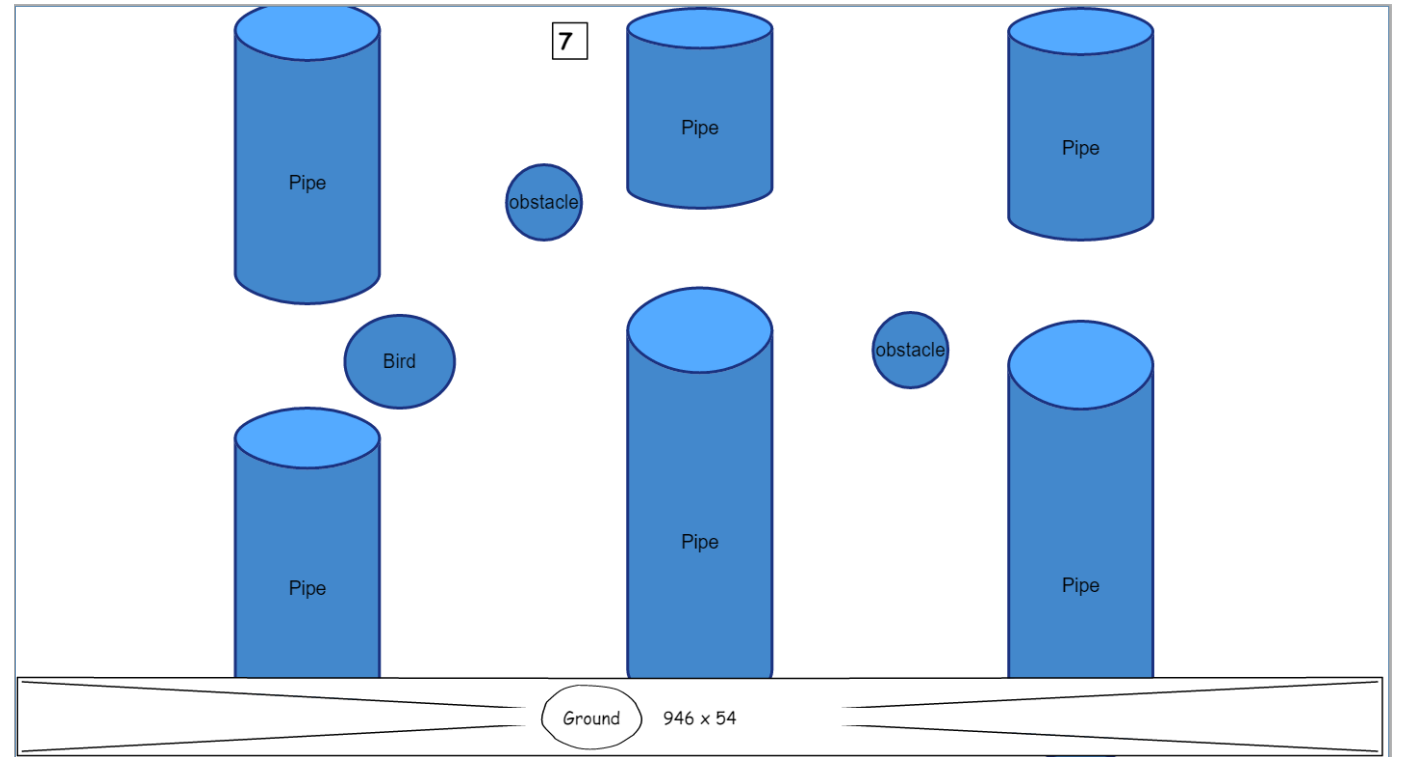
# UI Wire Frames- Play Screen

---

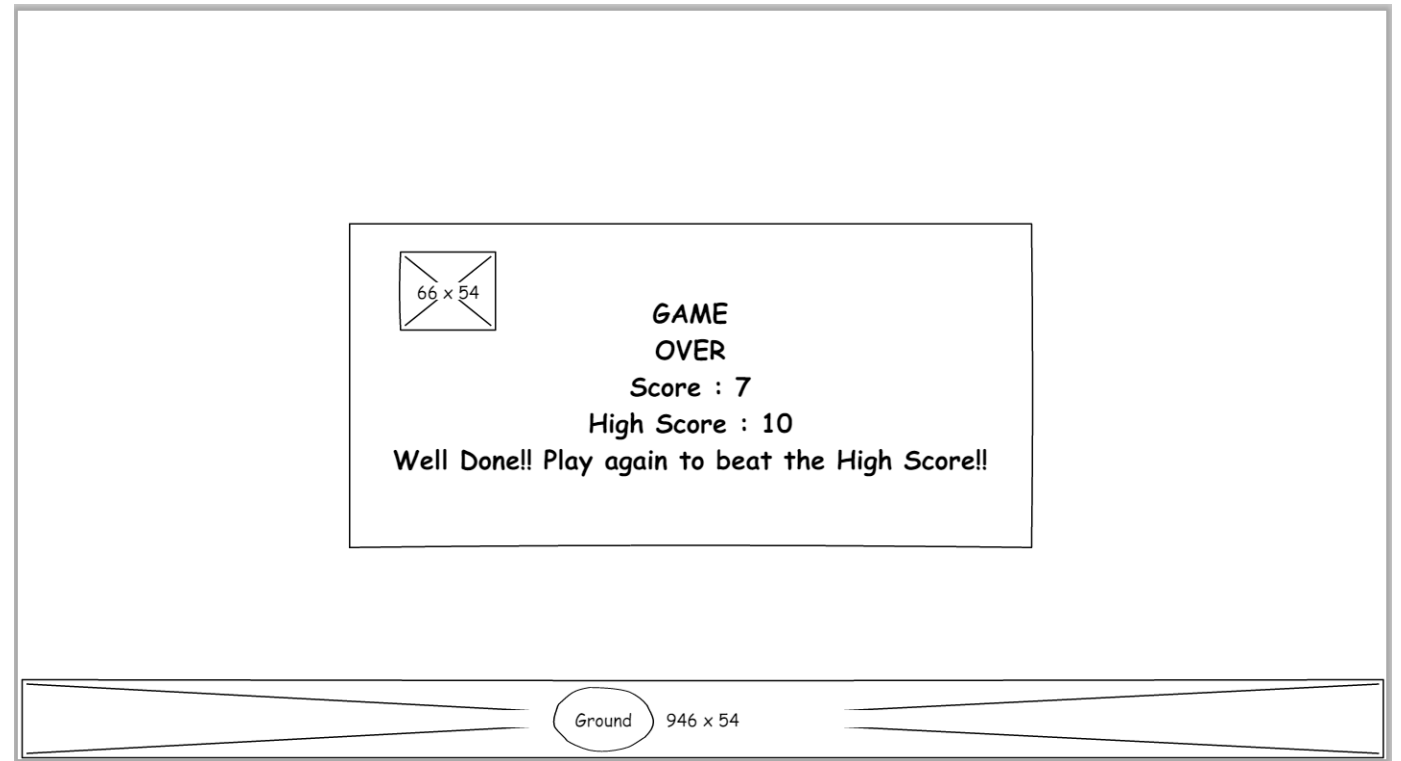


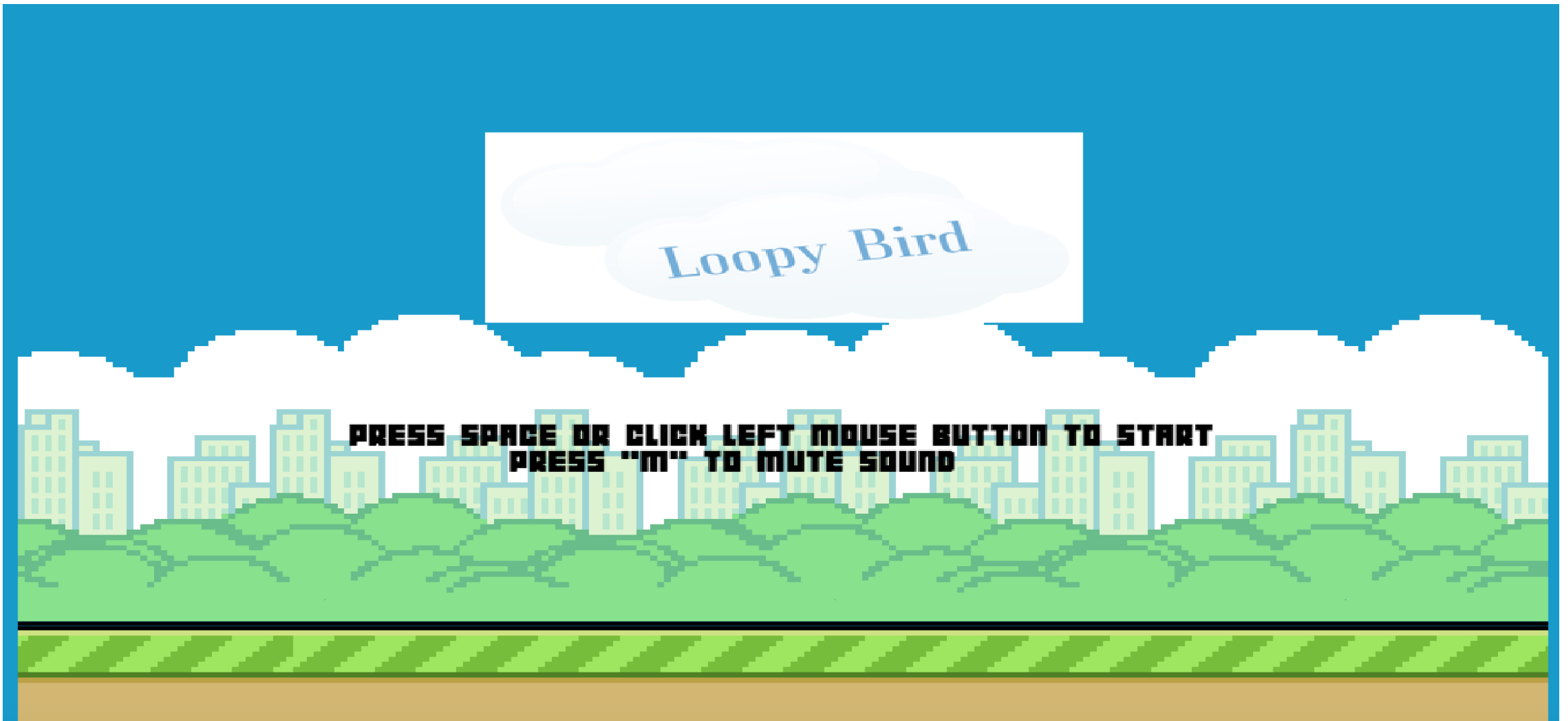
# UI Wire Frames- Play Screen

---



# UI Wire Frames- Game Over

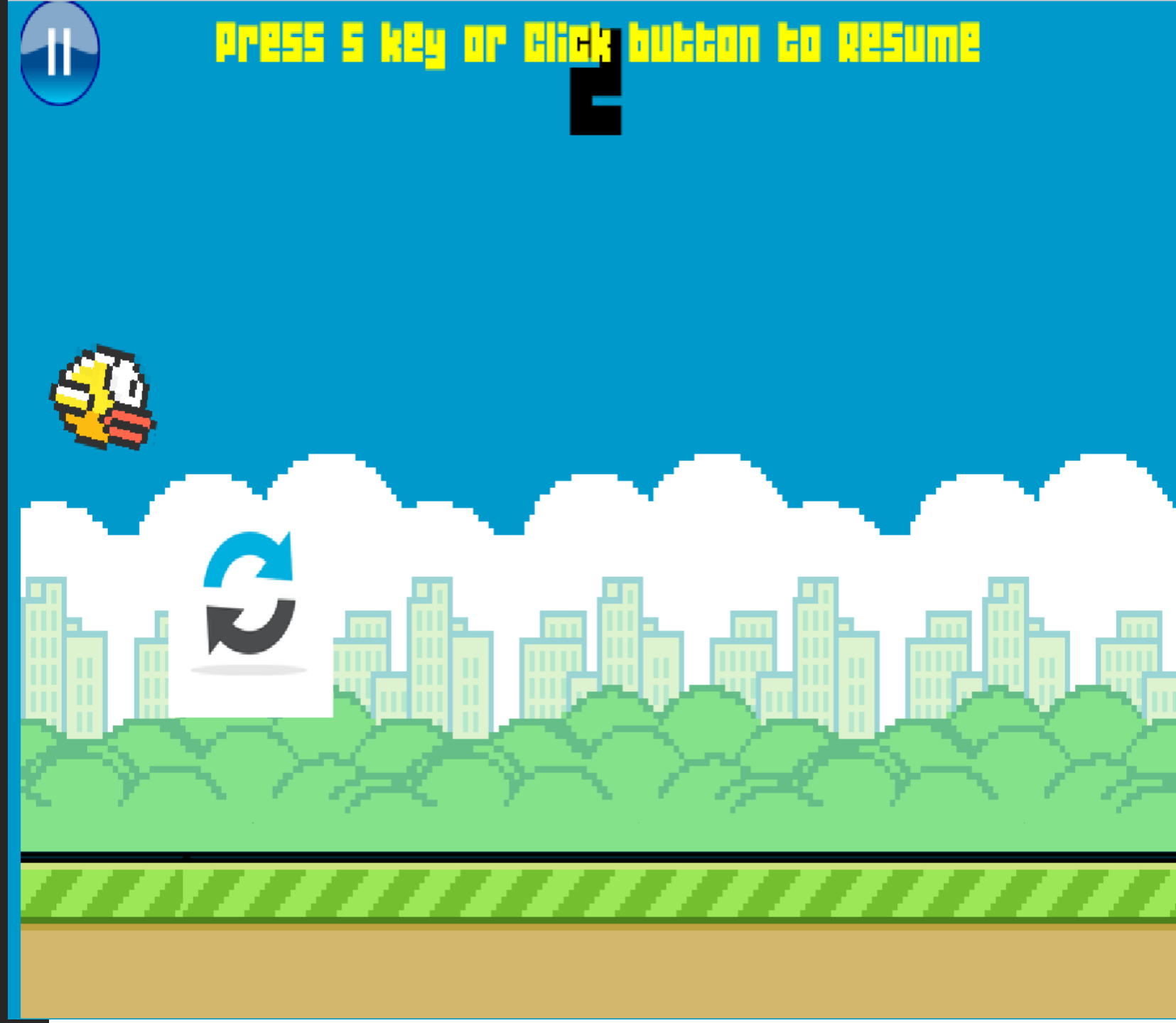


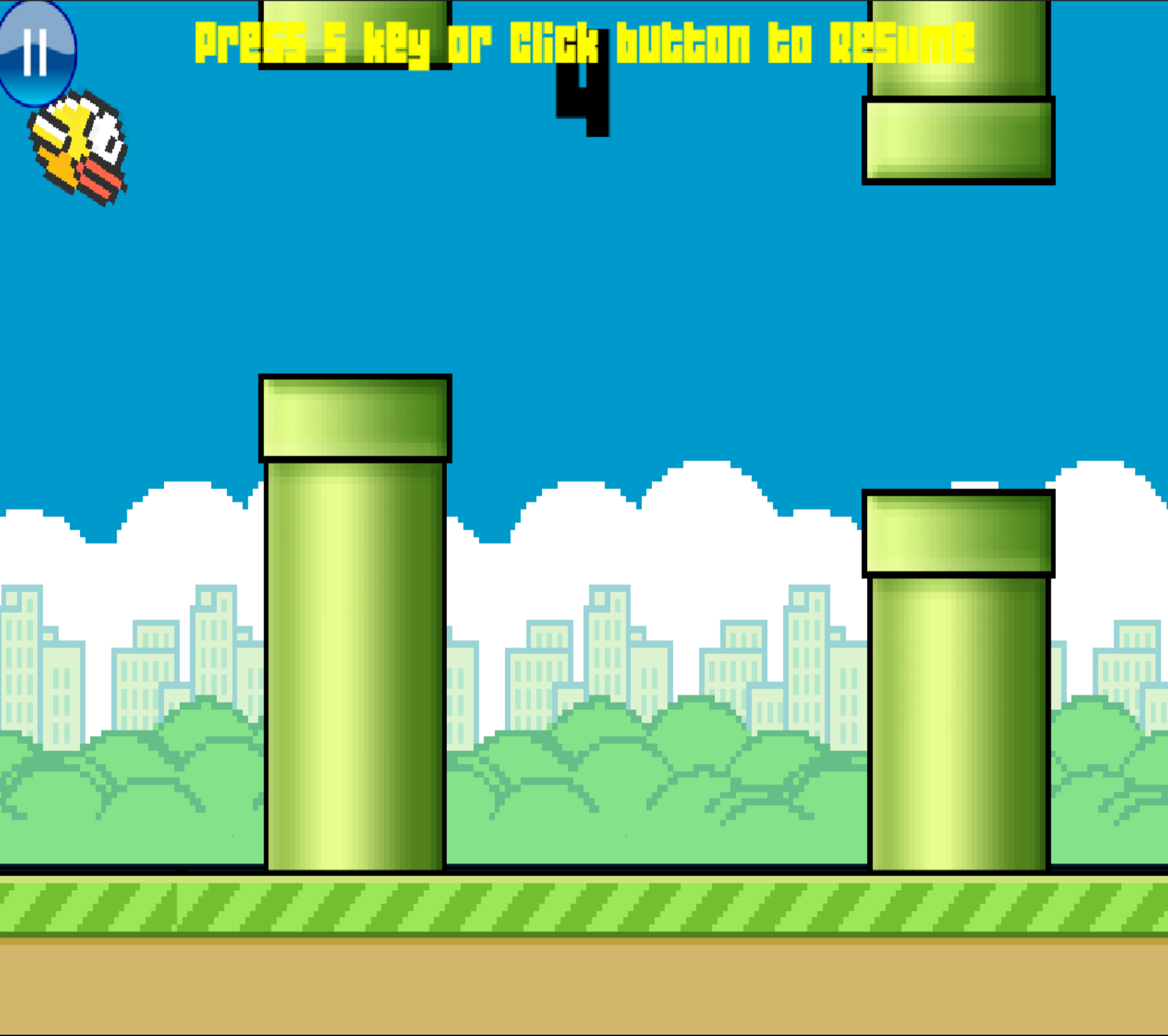


The Loopy Bird Game and the whole MelonJS world..

## 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





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

## Game Over Screen



**game  
over**

**Score: 10**

**Higher Score: 10**

**Well Done! Play again to beat HiScore**