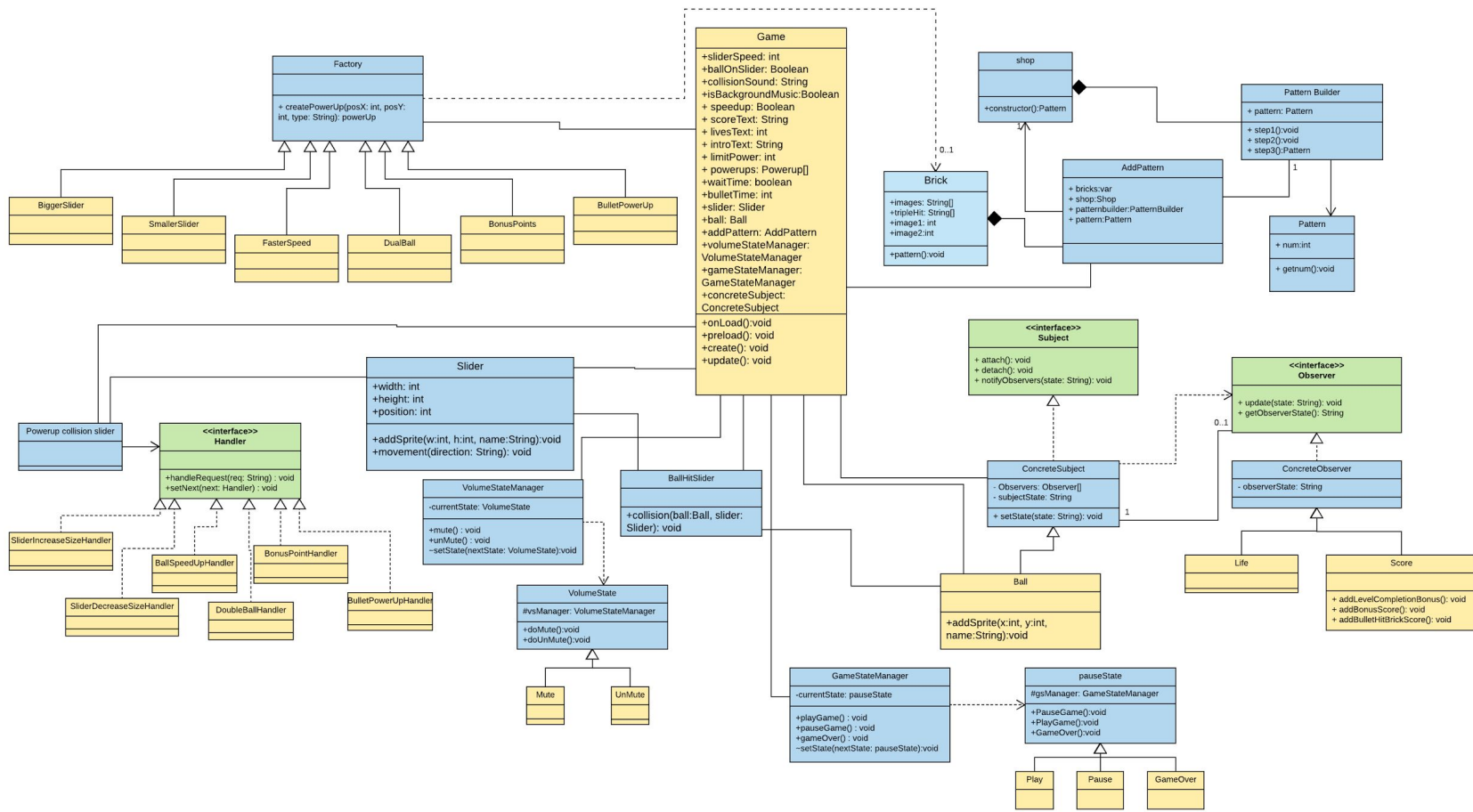

Team 2 iRobot

Breakout Game

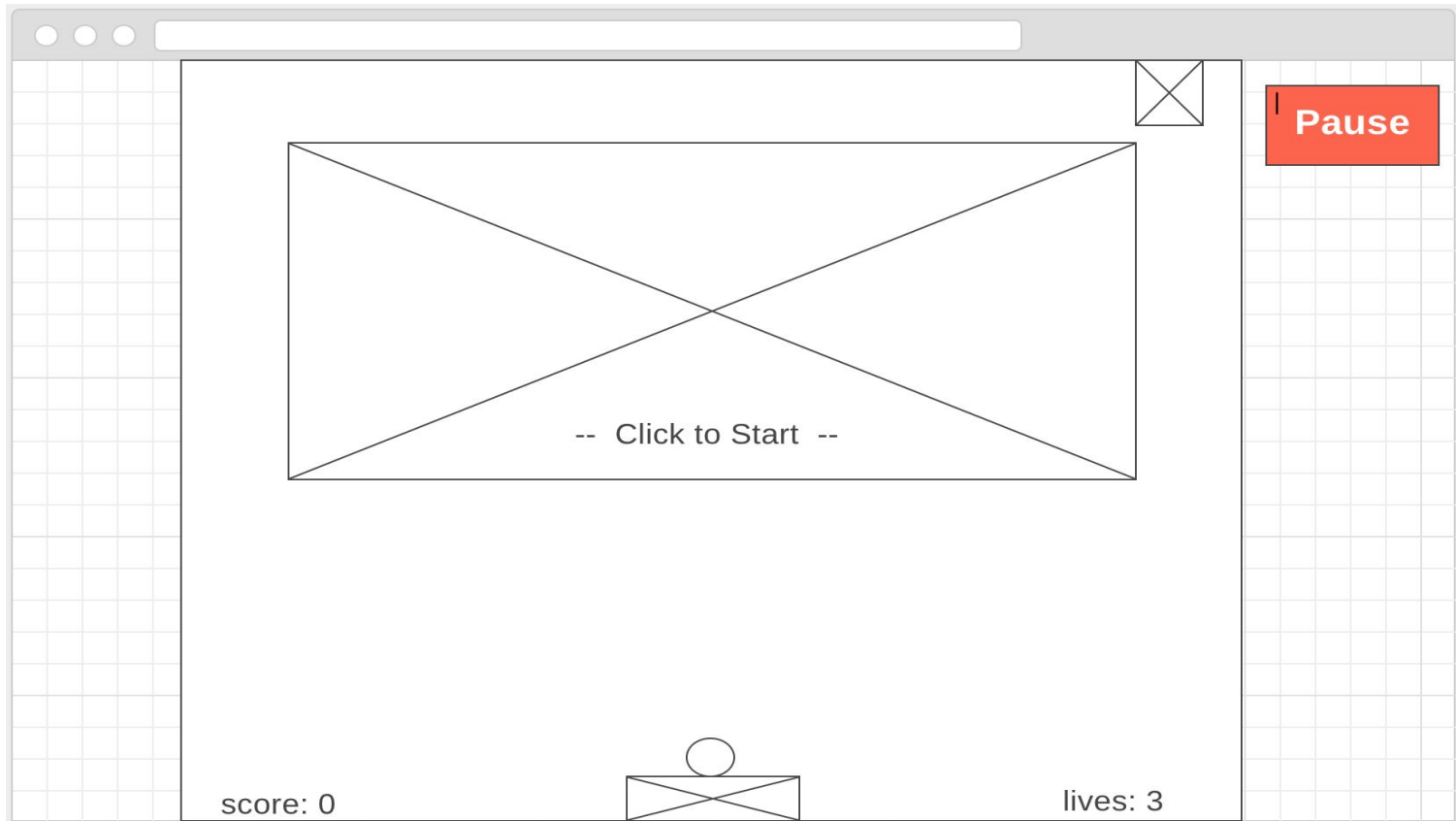
Team Members

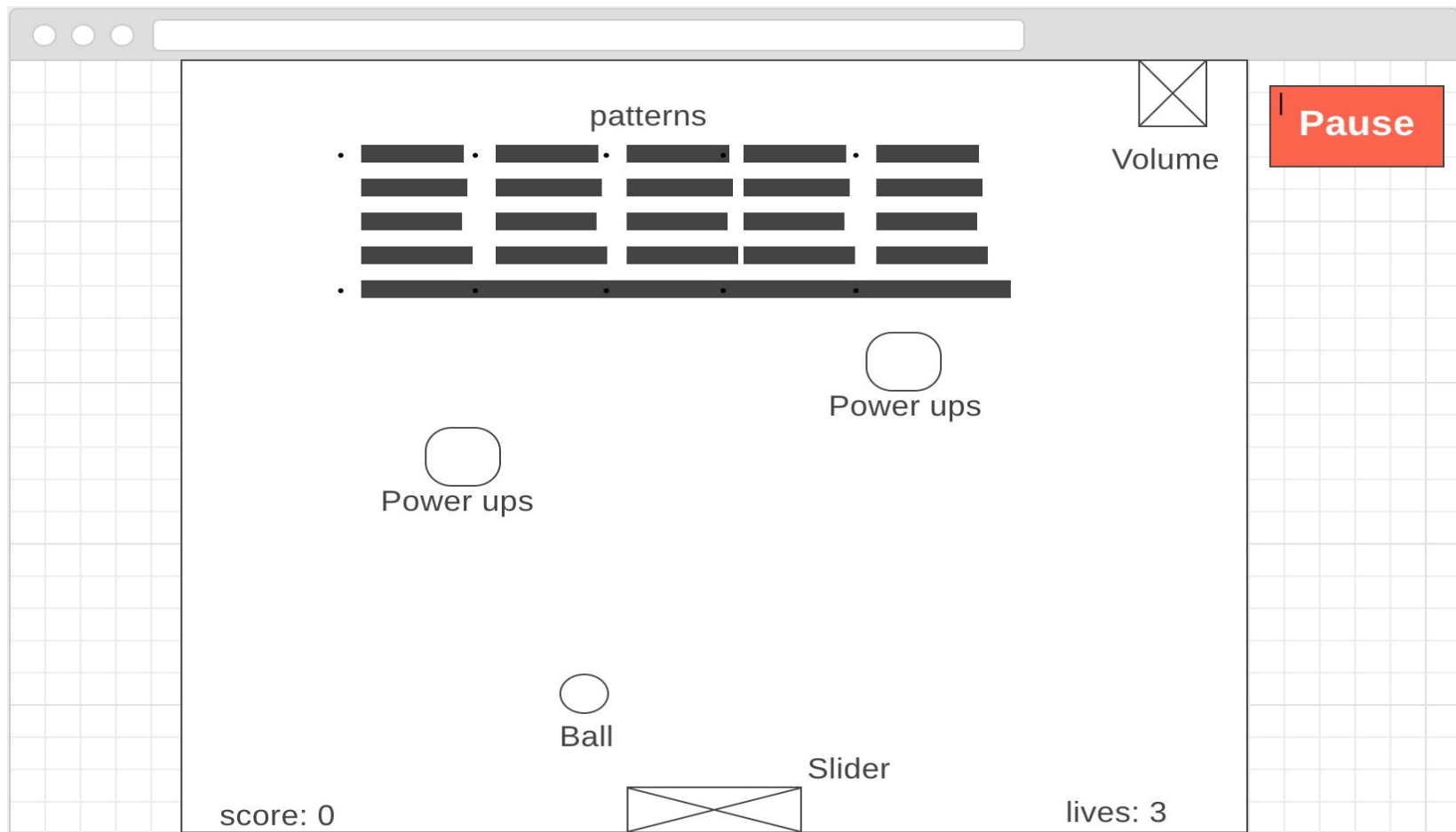
- Aditya Parmar - 011819964
- Afreen Patel - 011811397
- Neil Thaker - 011538215
- Purvesh Kothari - 011548615
- Rahil Modi - 011813789

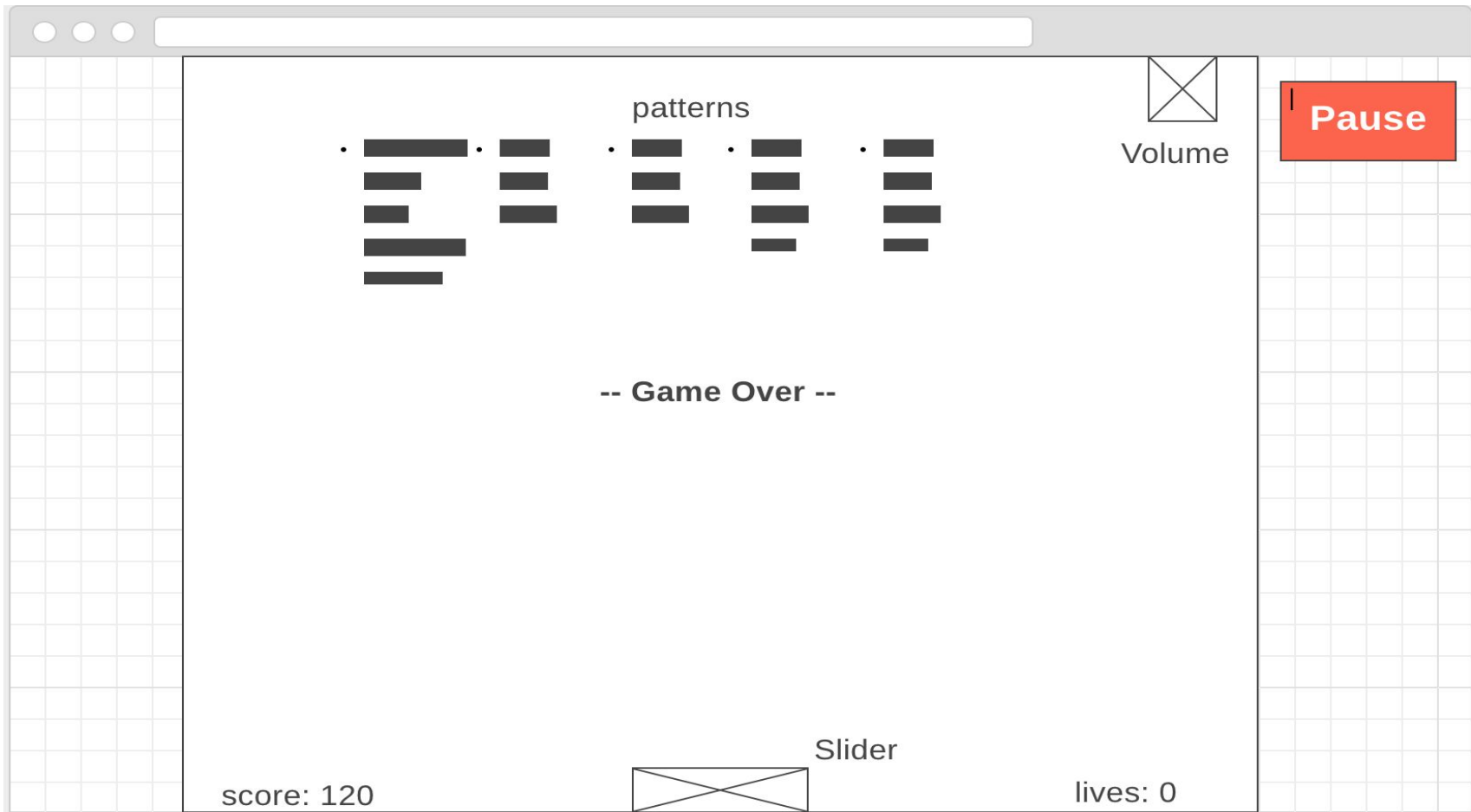
UML Class Diagram



Wireframe

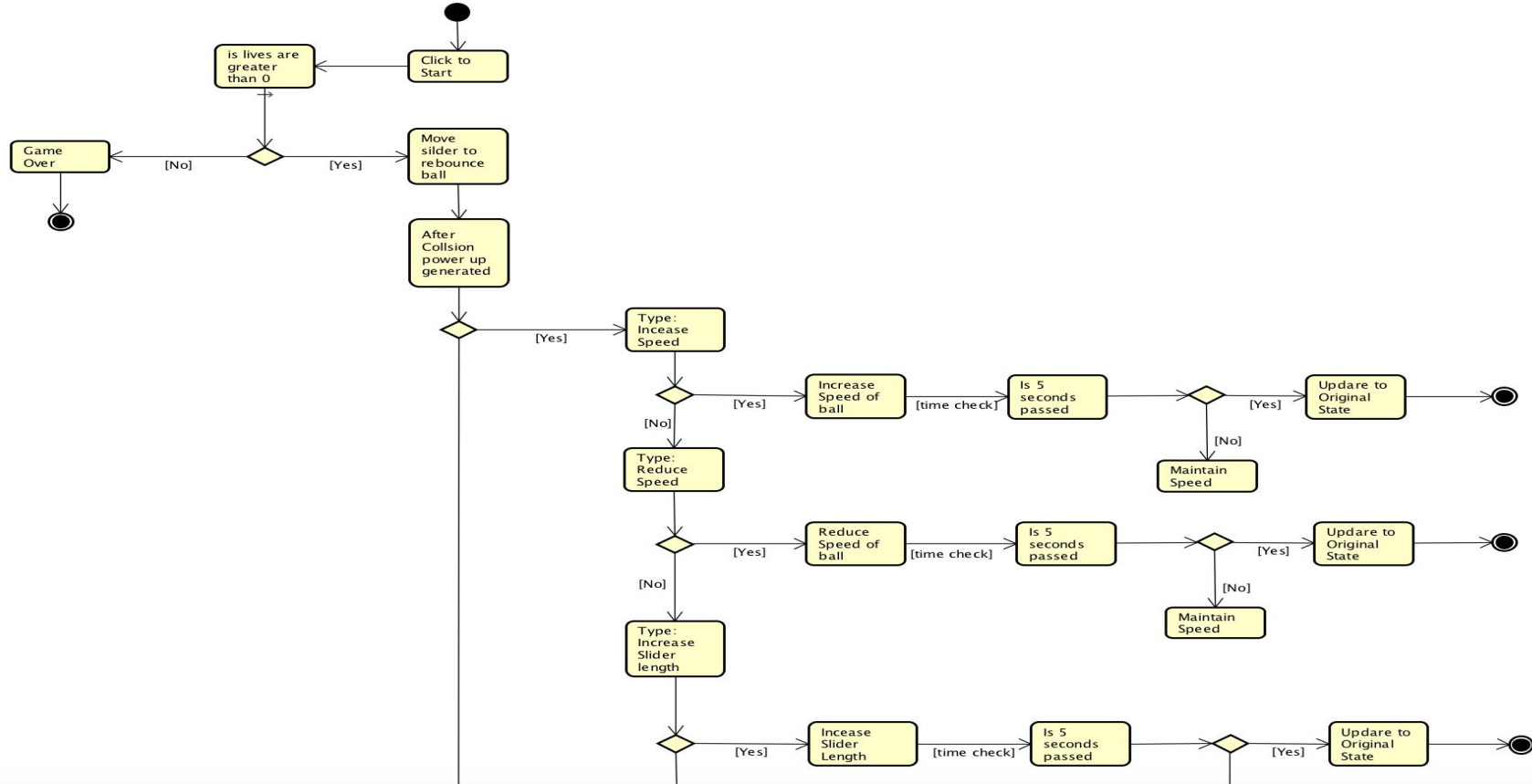


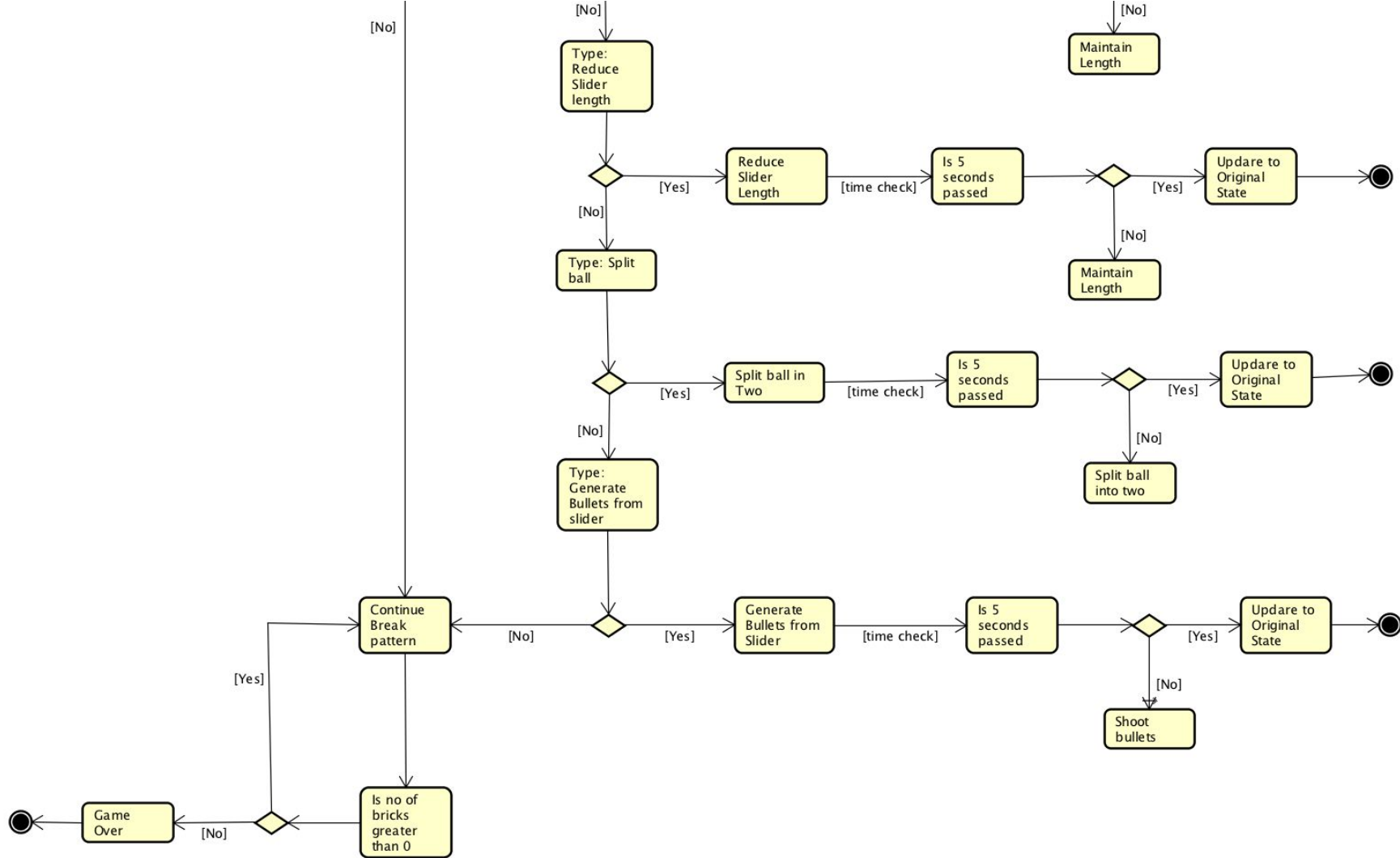




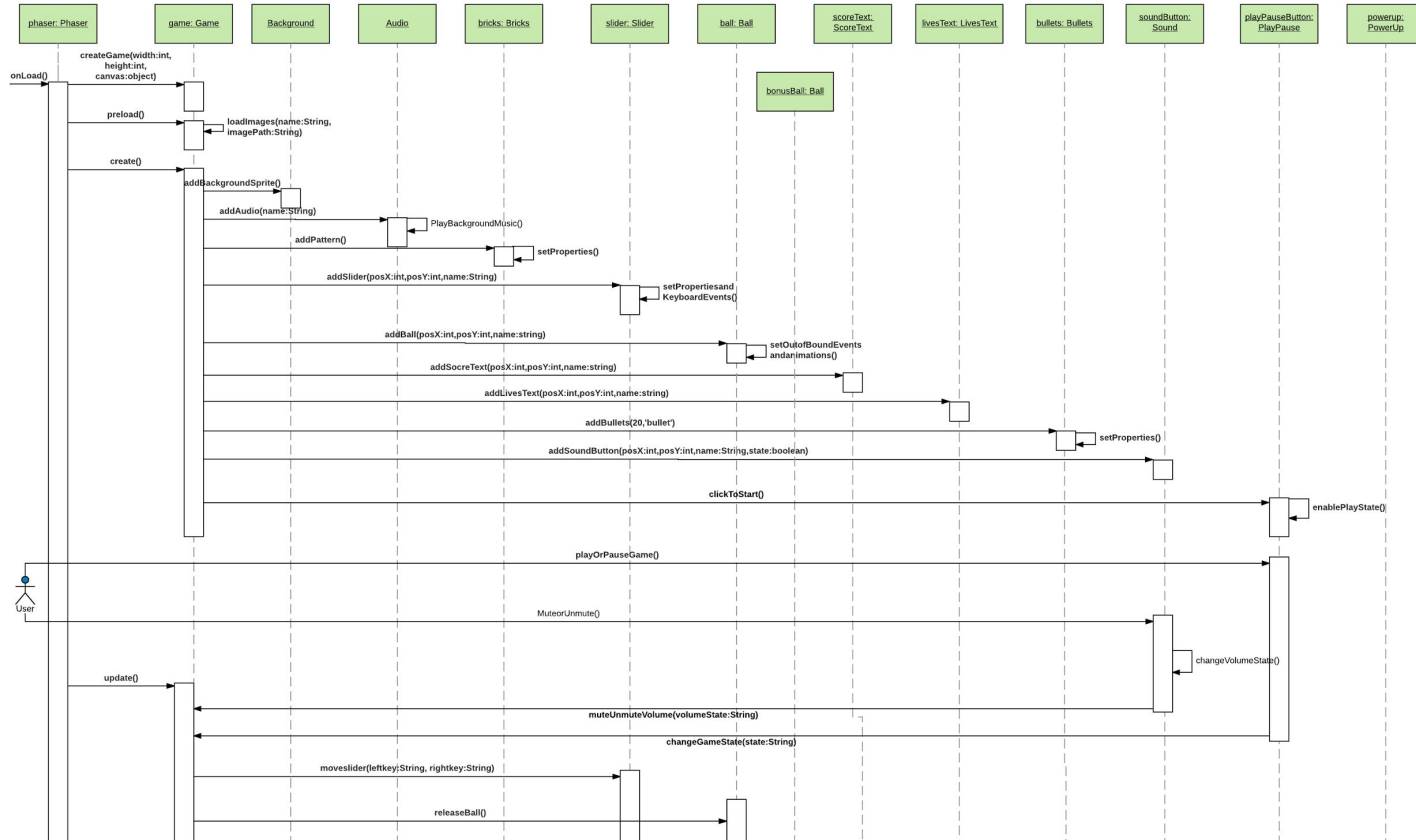
Activity Diagram

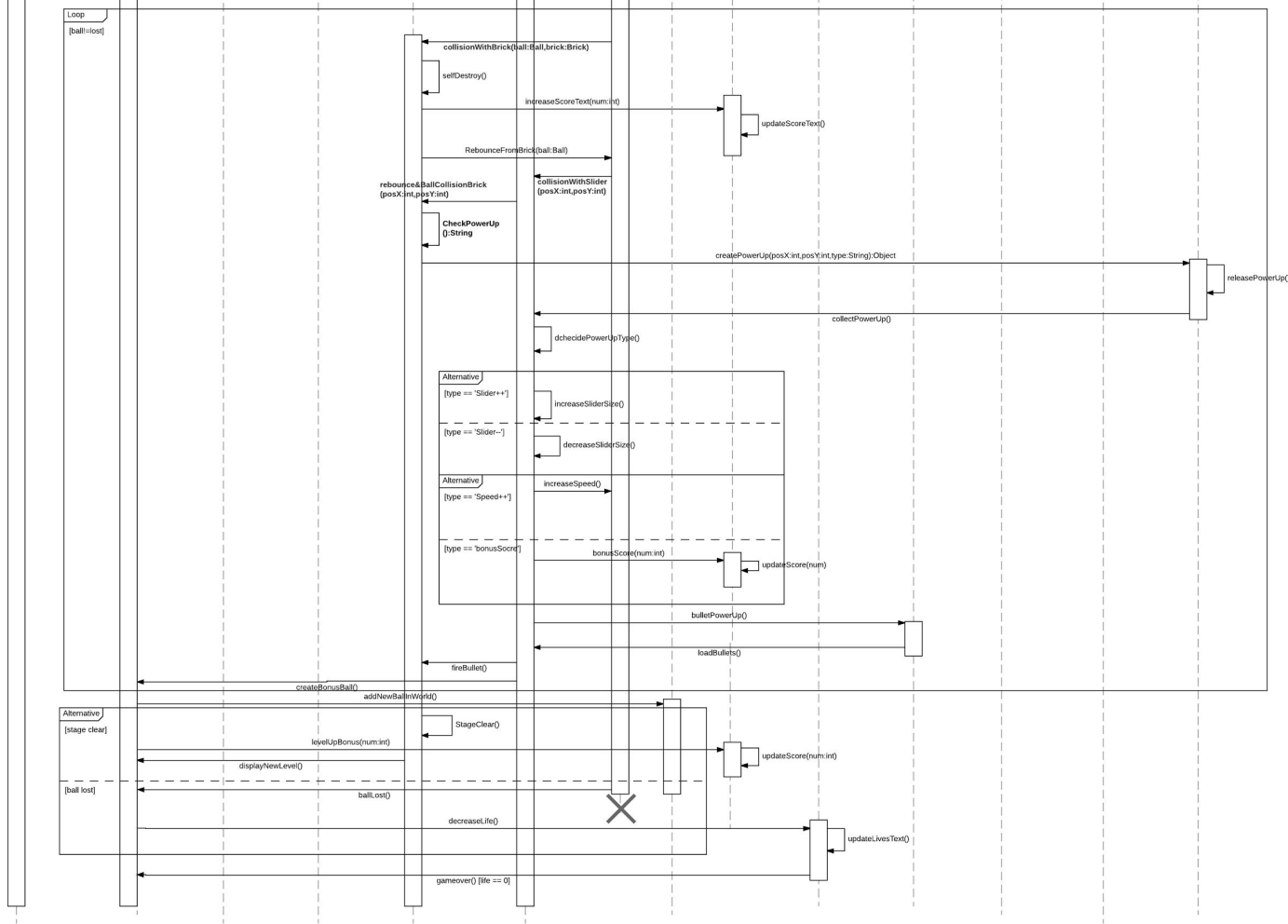
act Activity Diagram0





Sequence Diagram





Use case spec

Use case name: Play a Breakout Game

Brief Description: Player plays a Breakout Game

Actors: Slider, Ball

Overall Flow of events:

1. Slider moves left and right using keyboard.
2. Ball continuously moves in the game world and collides with brick pattern.
3. Sometime power up is generated for some action in the game.
4. Score and life gets updated as per hits of ball with bricks pattern.

Various Flows:

1. Slider movement in both left and right directions, default lives are 3 and points are 0.
2. Ball bounce back when hits the slider.
3. Ball bounce back when hits game world boundary.
4. Ball hits bricks pattern and pattern object destroyed. Score gets updated and randomly powerup may fall down.
5. If the powerup is collected by the slider then consequent action takes place depending on the type of power up.
6. If the powerup is of slider width change then it will update slider width. Power up can be of increasing ball speed or increase 50 points.
7. Bullets powerup can hit some bullets on the bricks and bricks will be broken.
8. Once all pattern bricks are destroyed a stage is cleared and new stage will be started.
9. If all three lives are lost and bricks pattern is not totally broken then game will be over.

Preconditions:

- Player opens the browser and game page gets loaded.
- Player clicks in the Game World to start a game.

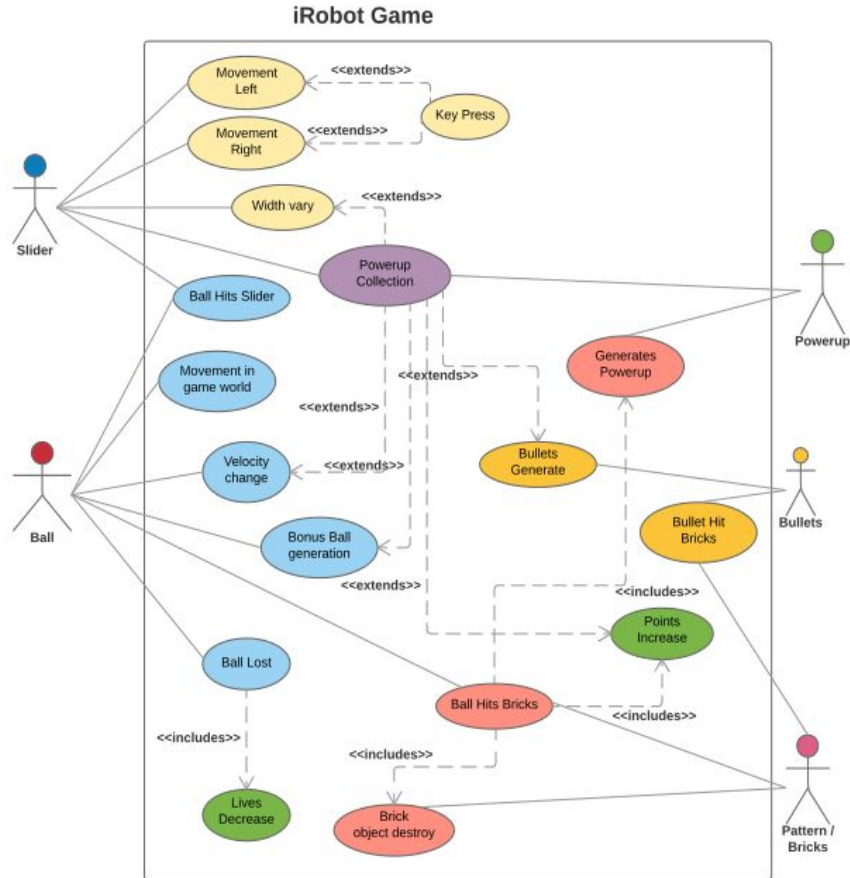
Success Guarantee:

- Player has cleared a stage and whenever player lost 3 lives then game will be over.

Minimal Guarantee:

- Player couldn't clear a single stage.

Use case diagram



User Story

User Story: Play the breakout game

In order to play the breakout game

As a player

I want to handle the ball so that it destroys bricks in the breakout pattern.

Scenario#1: User should be able to move the slider in both the direction

Given that user has started the game

When he presses the left arrow key on the keyboard

And he presses the right arrow key on the keyboard

Then the slider moves to the left side on the screen

And slider moves to the right side on the screen

Scenario#2: User destroys the brick in the breakout pattern

Given that user has started the game

And ball is alive in the game world

When ball bounces back from slider

And hits the brick in the breakout pattern

Then a brick is destroyed from the game world

Scenario#3: User gains the points on brick destroy

Given that user has started the game

And ball is alive in the game world

When ball collides with the brick in the breakout pattern

Then current score is increased by 10 points

Scenario#4: User collects the power ups

Given that the ball is alive in the game

And the ball has collided with the break generating the power up

When the power up falling downwards collected by the user using slider

Then the respective effect of power up is applied in the game

Scenario#5: Game is over

Given that user has only one life left in the game

When the ball goes out of the bound from the game world

Then the total available life goes to zero

And the game transits to the game over state

And game automatically restarts after 5 seconds

Scenario#6: User makes progress to the next stage

Given that user has one or more lives available in the game

And a ball is live in the game

When the last available brick in the pattern destroyed

Then a game automatically moves to the next stage

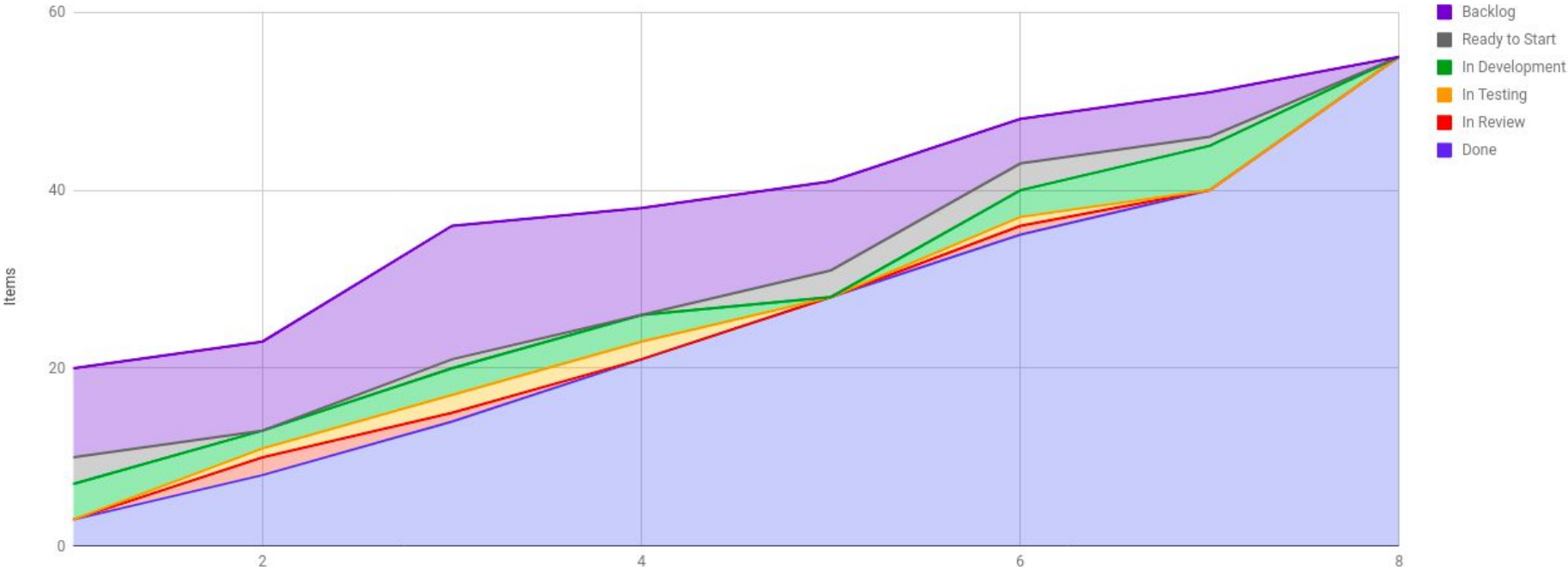
And a stage bonus of 50 points added to the total score

Task Board

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

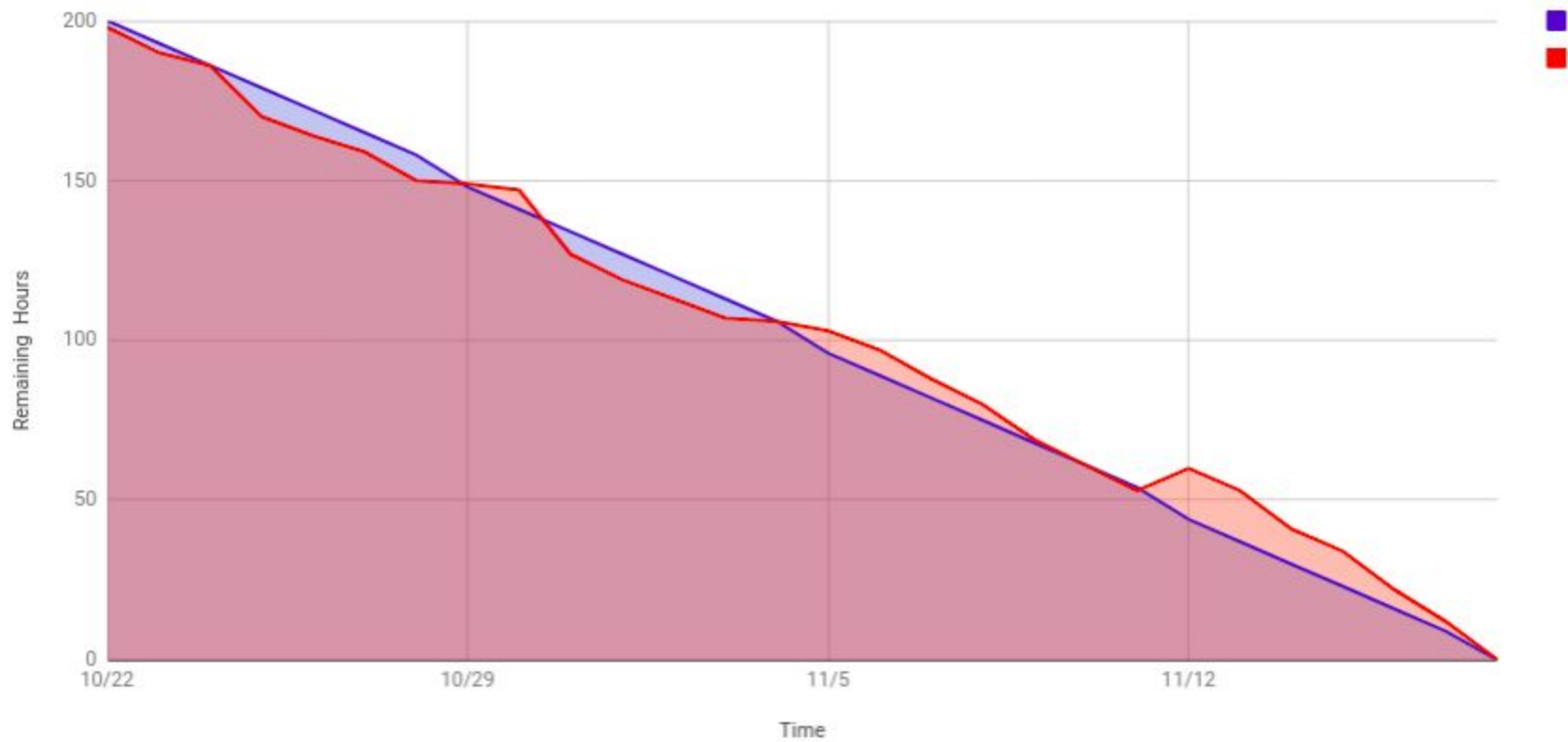
Cumulative Flow Diagram

Cumulative Flow



Scrum Burndown Chart

Burndown (Team iRobot)



Project Retrospective

What went well?

- Sufficient research helped to finalize the technologies for development of the game.
- As per estimation and using various XP core values, Breakout game is developed in Phaser JS Game engine as per requirements.
- Timely and frequent commits helped to keep the state of the game working most of the time.
- Proper escalation of the issues helped to fix the bugs quickly and efficiently.
- Team bonding and collaborative effort during development hurdles.

What didn't go so well? That needs to be improved

- Project task board was not up to date sometimes causing the misunderstanding about the project's progress.
- Less familiarity with the Javascript code testing strategies took more time in testing and bug fixing.
- Manual testing of the code upon new commits led to bugs in the functionalities.
- Acceptance of new stories when a sprint was about to end.
- Some features of the game were changed due to the dependency on the Phaser JS game engine than actually thought of.

What have we learned?

- To work as a team and provide the accurate estimates of the tasks.
- Importance of the different agile strategies like Kanban and Scrum and its related documentation.
- Implementation of the Agile principles to make the team self improving.
- Assignment of the tasks efficiently among team members based on individual's skill sets and expertise.
- Implementation of design patterns in Javascript.



Thank You

