Acceptance Test Report

Project Code Defenders - Robo Tournament
Team Codebenders

Agenda

- Our Team
- Project Vision and Requirements
- Performing Acceptance Tests
- Acceptance Tests Results

Our Team

Frontend

Product Owner

Fanny Delnondedieu

Scrum Master

Dominik Brdar

Testers

Hrvoje Rom

Fabio Patella

Backend

Simone Mezzaro

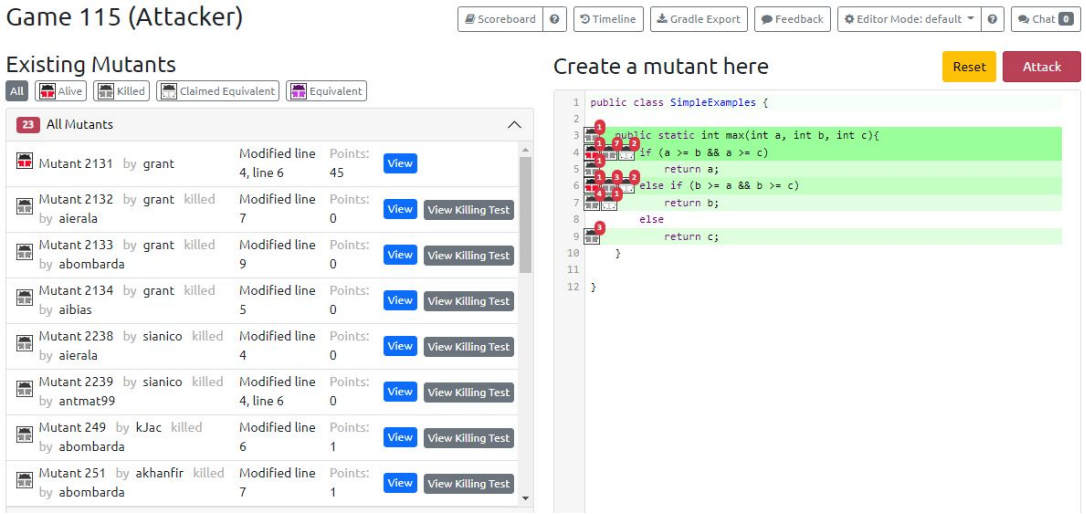
Riccardo Nava

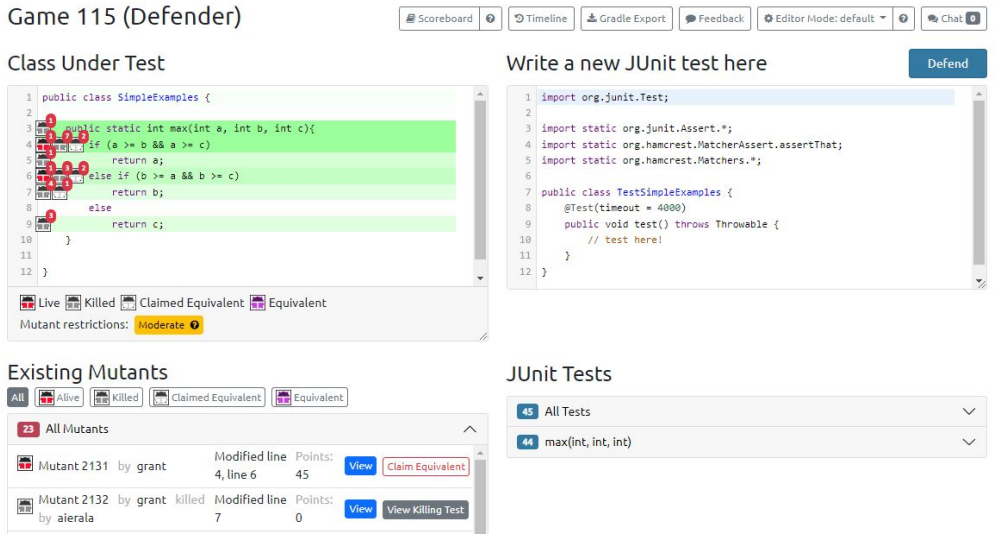
Andrea Restelli

Project vision

Code Defenders

- Software quality and testing are at the heart of software engineering, but they may not always get enough attention from software engineering education.
- CodeDefenders (web game) proposes the use of **gamification** to teach **mutation testing** and to strengthen code writing and testing skills.
- The game supports **team play and competition** by having Attackers - Defenders teams whose goal is to inject errors into code or write unit tests to catch them.
- The **“CodeDefenders: RoboTournament”** project aims at enriching the game by adding support for students tournaments and games against bots.





- For more details refer to Project Plan document

Project requirements

- Implement a **tournament application**. This application must use CodeDefenders as a remote service (through APIs) and must include at least two tournaments modalities.
- Design and implement a set of **OpenAPIs for CodeDefenders** which can be used from the tournament application to manage games and players.
- Implement a **load balancing** mechanism which allows the tournament application to communicate with **multiple CodeDefenders servers** and to always create games on the less loaded server.
- Implement a **streaming** component which allows users to follow in progress games live. This component can optionally include an “overall tournament view” showing schedule, standings and other information for each tournament.
- Design and implement a set of **APIs** which allows users to train **bots** over past games data and to let those bots play CodeDefenders.
- Please refer to document *Requirements Definition* for more details

Performing Acceptance Tests

- We listed all the actions that can be performed in our application
  - Each action maps one or more requirements by the customer and is covered by one specific acceptance test
  - Each test in this document has ID, name and description (what action is it testing), link to User Story containing mapped project requirement
  - Refer to Acceptance Test Plan document for the complete description of the tests
- As planned, we went through all the tests together with our customer to verify that they are successful and that the behavior of the application is the one required by him.
- Finally, the customer expressed whether the product we implemented meets all his needs or not.
- Acceptance tests coverage of the user stories is shown in the next slides

Performing Acceptance Tests

- As planned, we had a meeting with the customer on Saturday 07/01/2023 at 16:30 to show him the final version of the product and go through acceptance tests together.
- The Product Owner was present, together with five other members of the team.
- Tests from 1 to 22.3 were performed by Fabio Patella.
- Tests 23 and 24, about bots APIs, were performed by Riccardo Nava.
- For each test, the customer expressed whether or not it was passed. For some tests he also added some comments, that are reported in the next slides.

Tests mapped onto User Stories

CDF-32 Login/Register

Test-1 Registration
Test-2.1 Successful Login
Test-2.2 Rejected Login

CDF-41 Display Tournaments Info

Test-3 Display Tournaments Information

CDF-35 Team Creation

Test-4 Create a Team

CDF-54 Team Management

Test-5 Leave the Team
Test-6 Kick Members Out of the Team
Test-7 Promote Team Member as Leader
Test-8 Invite Players to the Team

CDF-37 Join Team

Test-9 Join an Open Team
Test-10.1 Accept Received Invitations
Test-10.2 Decline Received Invitations

CDF-33 Create Tournament

Test-11 Create Tournament
Test-20 Upload a Class
Test-21 Choose a Class

CDF-34 Join Tournament

Test-12 Join Tournament

Tests mapped onto User Stories

CDF-36 Starting Games

Test-13 Tournament is Started
Test-14 Games are Split in Phases
Test-15 Users can Play Games

CDF-38 Leave Game and Game End

Test-16 Return to Tournament App

CDF-39 View Game Stream

Test-17 Join a Game Streaming

CDF-40 Notifications of Game Stream Updates

Test-18 Receive Game Streaming Updates

CDF-31 Load Balancing

Test-19 Load Balancing
Test-22.1 Register a CD Server Instance
Test-22.2 Update a CD Server Instance
Test-22.3 Remove a CD Server Instance

CDF-43 Bots can Play

Test-23 Bots can Play a Game

CDF-44 Bots can be Trained

Test-24 Bots can be Trained

Summary of the acceptance tests

User Story	Test	Status	Comment
CDF-32 Login/Register	Test-1 Registration	PASS	
	Test-2.1 Successful Login	PASS	
	Test-2.2 Rejected Login	PASS	
CDF-41 Display Tournaments Info	Test-3 Display Tournaments Information	PASS	
CDF-35 Team Creation	Test-4 Create a Team	PASS	
CDF-54 Team Management	Test-5 Leave the Team	PASS	
	Test-6 Kick Members Out of the Team	PASS	
	Test-7 Promote Team Member as Leader	PASS	
	Test-8 Invite Players to the Team	PASS	

User Story	Test	Status	Comment
CDF-37 Join Team	Test-9 Join an Open Team	PASS	
	Test-10.1 Accept Received Invitations	PASS	
	Test-10.2 Decline Received Invitations	PASS	
CDF-33 Create Tournament	Test-11 Create Tournament	PASS	
	Test-20 Upload a Class	PASS	
	Test-21 Choose a Class	PASS	
CDF-34 Join Tournament	Test-12 Join Tournament	PASS	*Teams with too many players can not join, as a future improvement we could allow the team leader to select who can participate*
CDF-36 Starting Games	Test-13 Tournament is Started	PASS	
	Test-14 Games are Split in Phases	PASS	
	Test-15 Users can Play Games	PASS	
CDF-38 Leave Game and Game End	Test-16 Return to Tournament App	PASS	

User Story	Test	Status	Comment
CDF-39 View Game Stream	Test-17 Join a Game Streaming	PASS	
CDF-40 Notifications of Game Stream Updates	Test-18 Receive Game Streaming Updates	PASS	
CDF-31 Load Balancing	Test-19 Load Balancing	PASS	
	Test-20 Upload a Class	PASS	
	Test-21 Choose a Class	PASS	
	Test-22.1 Register a CD Server Instance	PASS	*No admin page so we need to manually send API requests. Admin page could be added as a future improvement.*
	Test-22.2 Update a CD Server Instance	PASS	*PUT instead of POST request is more appropriate when updating a resource*
	Test-22.3 Remove a CD Server Instance	PASS	*What happens to the ongoing matches hosted on this instance if it is removed? This operation is allowed only when there are no ongoing tournament matches hosted on the instance to be removed. As a future improvement we can handle also the case when this prerequisite is not satisfied (i.e. fault tolerance).*

User Story	Test	Status	Comment
CDF-43 Bots can Play	Test-23 Bots can Play a Game	PASS	
CDF-44 Bots can be Trained	Test-24 Bots can be Trained	PASS	*Right now the API is working as expected so the test is passed. As a future improvement we could suit the data returned depending on the data needed by the bots for training purposes. For example by adding filters.*

Summary

In conclusion, the customer expressed his satisfaction with the product we implemented. It met all his needs and satisfied all the mandatory requirements.

In the next slide, we are presenting the initial requirements listing and for each of them, its completion status at the end of the project.

What was not done were **optional** requirements.

List of requirements

Legend:
Done
Not done

Functional requirements

Tournament App
- Tournament list
- Multiple active tournaments
- Multiple CodeDefenders servers
- Register to tournament app
- Login to tournament app
- New tournament
- Choose tournament type
- Create teams
- Join teams
- Join tournament (single)
- Join tournament (team)
- Schedule tournament matches
- Notify of upcoming match
- Assign teams/players to matches
- Redirect to CodeDefenders
- Redirect to Tournament app
- Restrict CodeDefenders interaction
- Tournament information overview
- Match results

Streaming App
- Guests can view streams
- Semi Real-time match overview
- Semi Real-time match updates
- Visual effects
- Visual effects toggle

Bots integration
- Bots history
- Bots join matches
- Bots play matches
- Bots get match status

Non Functional requirements

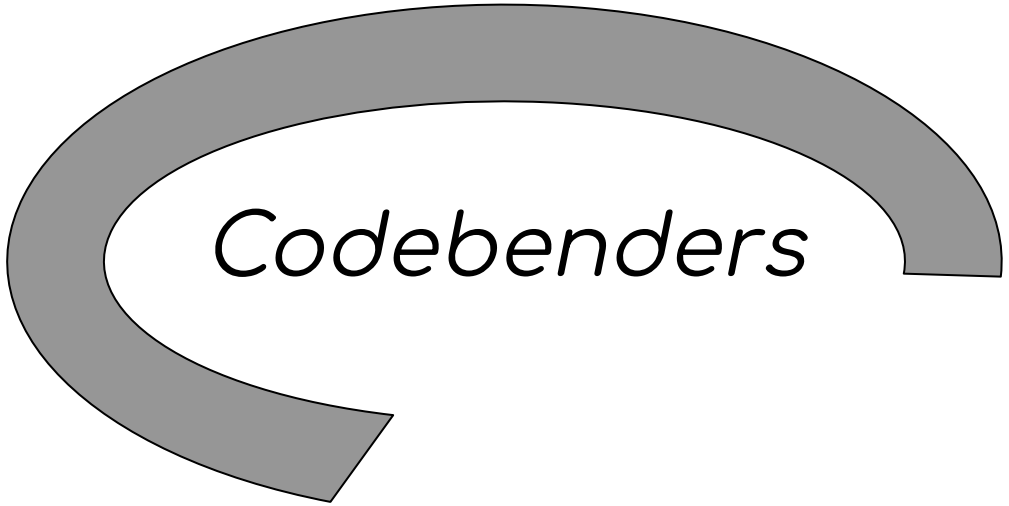
- Load balancing
- Latency of events
- Fault tolerance of CD servers

Statistics

Overall, our acceptance tests had **100% pass rate**. They covered **all the mandatory requirements**, as shown in the previous tables.

Some of the optional requirements weren't implemented and, as such, were not covered by our acceptance tests.

Overall, the acceptance tests covered **27/31** of the requirements initially identified.



contact info:

fanny.delnondedieu@fer.hr
dominik.brdar@fer.hr
hrvoje.rom@fer.hr
simone.mezzaro@mail.polimi.it
fabio.patella@mail.polimi.it
andrea2.restelli@mail.polimi.it