KeyStorm

Group – 7 Team

Meenakshi Mazumdar | Product Manager

The KeyStorm C# console app project emphasizes effective project management and teamwork through Azure DevOps.

Katie Freerksen | SWE

Instrumental in organizing the project and ensuring effective communication within the team.

Ryan Moses | SWE

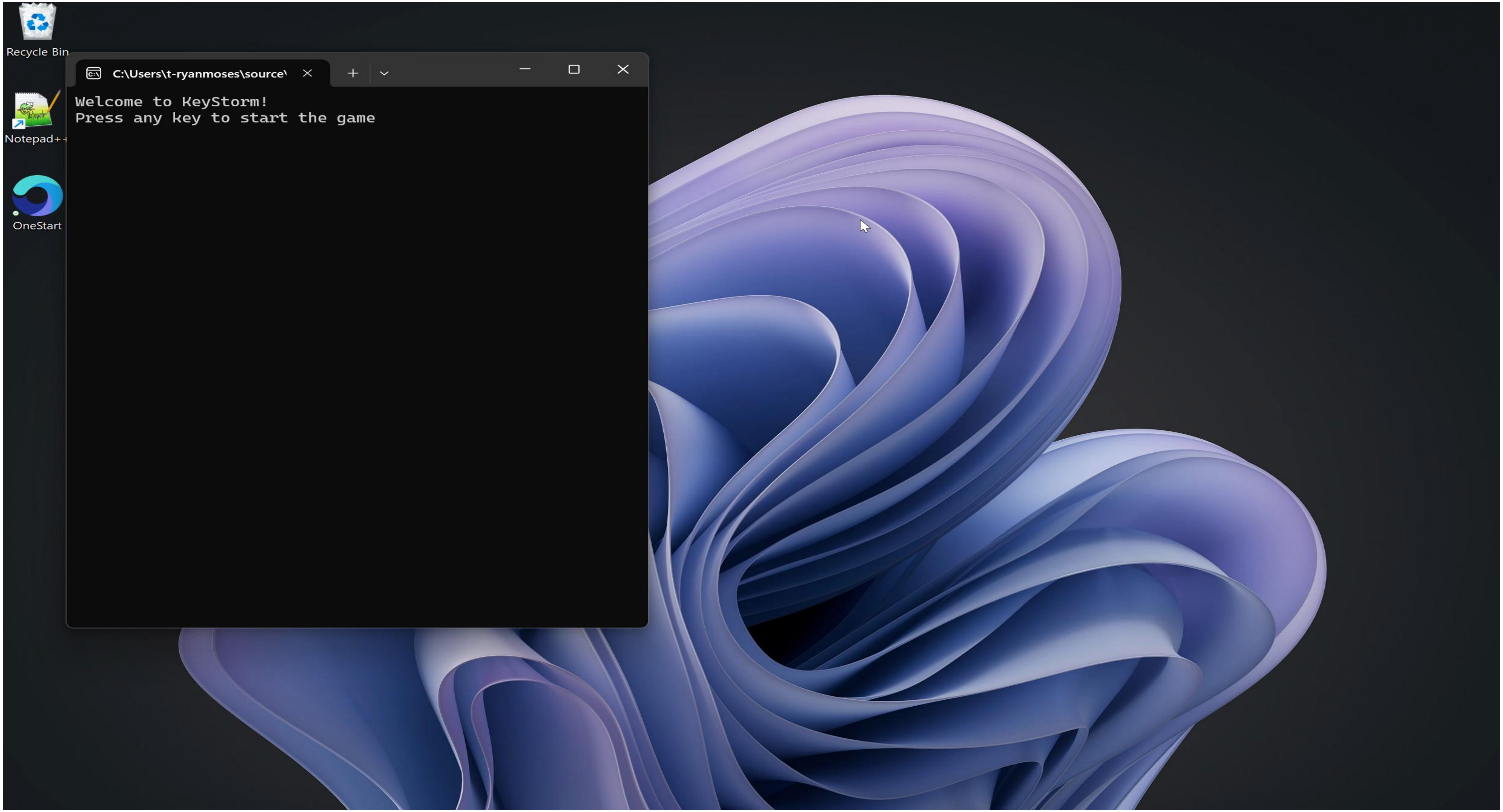
Significantly contributed by helping the team understand the code state, debugging, and integrating different components.

Skip Thompson | SWE

Proactive in addressing blockers and ensuring the team's progress with significant technical contributions.

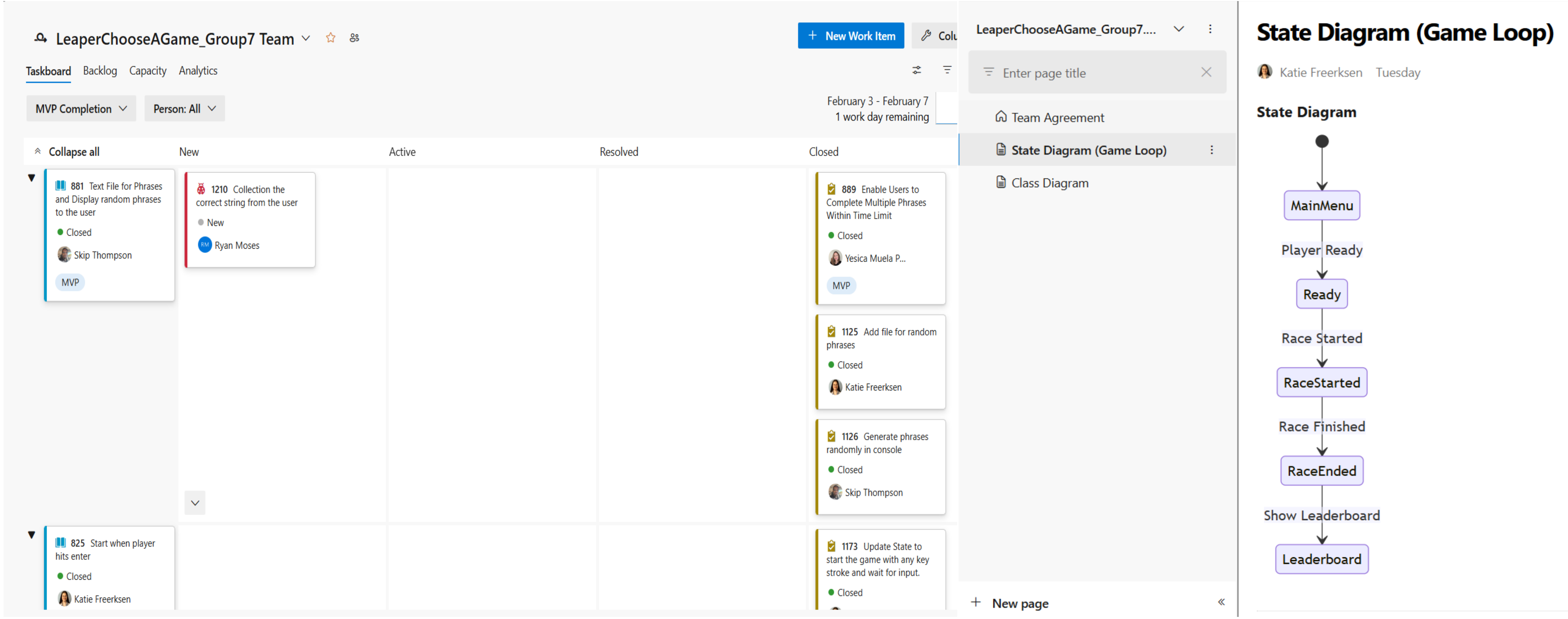
Yesica Muela Pando | SWE

Key player in implementing core features with a strong focus on attention to detail and problem-solving.



```
Welcome to KeyStorm!
Press any key to start the game
```

Azure DevOps



LeaperChooseAGame_Group7 Team

February 3 - February 7
1 work day remaining

New	Active	Resolved	Closed
881 Text File for Phrases and Display random phrases to the user ● Closed Skip Thompson MVP			
1210 Collection the correct string from the user ● New Ryan Moses			889 Enable Users to Complete Multiple Phrases Within Time Limit ● Closed Yesica Muela P... MVP 1125 Add file for random phrases ● Closed Katie Freerksen 1126 Generate phrases randomly in console ● Closed Skip Thompson
825 Start when player hits enter ● Closed Katie Freerksen			1173 Update State to start the game with any key stroke and wait for input. ● Closed

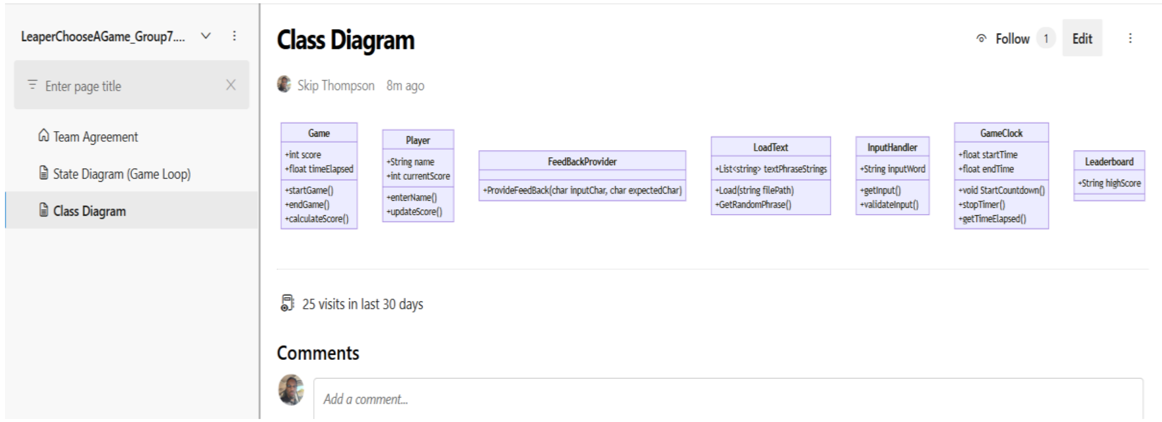
LeaperChooseAGame_Group7....

State Diagram (Game Loop)

Katie Freerksen Tuesday

State Diagram

Azure DevOps

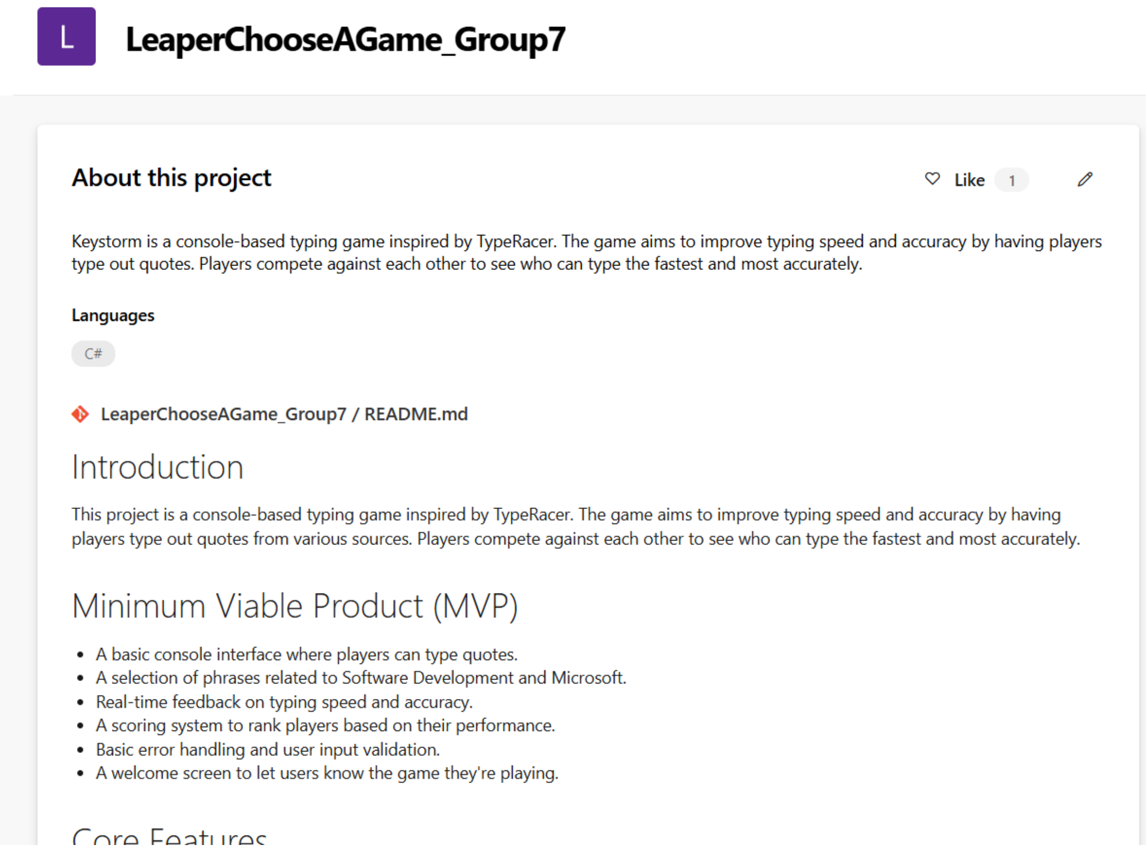


Class Diagram

Skip Thompson 8m ago

25 visits in last 30 days

Comments



LeaperChooseAGame_Group7

About this project

Keystorm is a console-based typing game inspired by TypeRacer. The game aims to improve typing speed and accuracy by having players type out quotes. Players compete against each other to see who can type the fastest and most accurately.

Languages

C#

LeaperChooseAGame_Group7 / README.md

Introduction

This project is a console-based typing game inspired by TypeRacer. The game aims to improve typing speed and accuracy by having players type out quotes from various sources. Players compete against each other to see who can type the fastest and most accurately.

Minimum Viable Product (MVP)

- A basic console interface where players can type quotes.
- A selection of phrases related to Software Development and Microsoft.
- Real-time feedback on typing speed and accuracy.
- A scoring system to rank players based on their performance.
- Basic error handling and user input validation.
- A welcome screen to let users know the game they're playing.

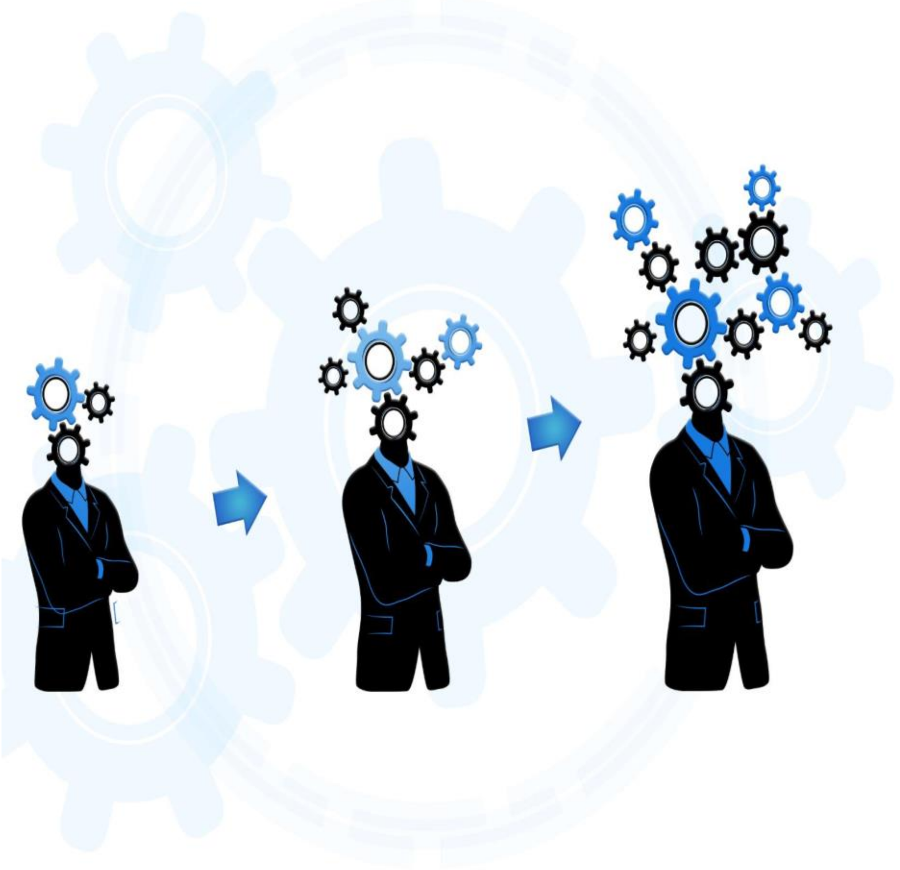
Core Features

Team Agreement

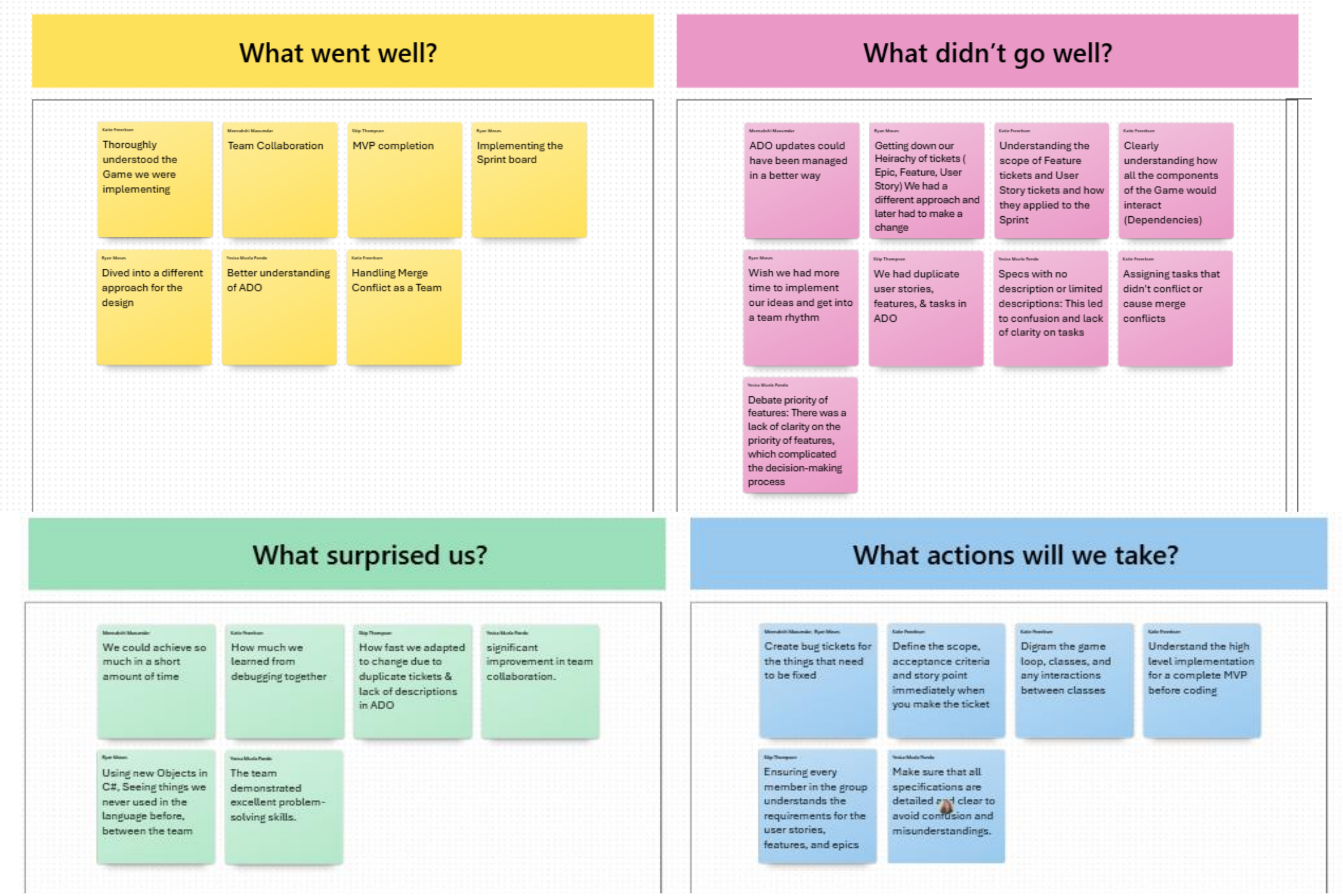
- **Team Values:** Trust, transparency, and alignment on common goals.
- **Knowledge & Engagement:** Attend meeting, respect others' ideas, and ask for help when needed.
- **Build the Right Thing:** Meet business criteria and understand customer needs.
- **Engineering Excellence:** Follow best practices, produce high-quality code, and continuously improve processes.
- **Communication:** Timely, respectful, and inclusive communication
- **Working Hours:** Core hours 8am – 4pm PT; no responses expected during holidays, vacations, or after 6PM PT.

Recent Modifications:

- Emphasized best engineering practices and continuous improvement.
- Added a point on understanding project needs.
- Clarified working hours expectations.



Sprint Retrospective



What went well?

- Thoroughly understood the Game we were implementing
- Team Collaboration
- MVP completion
- Implementing the Sprint board
- Dived into a different approach for the design
- Better understanding of ADO
- Handling Merge Conflict as a Team

What didn't go well?

- ADO updates could have been managed in a better way
- Getting down our Heirachy of tickets (Epic, Feature, User Story) We had a different approach and later had to make a change
- Understanding the scope of Feature tickets and User Story tickets and how they applied to the Sprint
- Clearly understanding how all the components of the Game would interact (Dependencies)
- Wish we had more time to implement our ideas and get into a team rhythm
- We had duplicate user stories, features, & tasks in ADO
- Specs with no description or limited descriptions: This led to confusion and lack of clarity on tasks
- Assigning tasks that didn't conflict or cause merge conflicts
- Debate priority of features: There was a lack of clarity on the priority of features, which complicated the decision-making process

What surprised us?

- We could achieve so much in a short amount of time
- How much we learned from debugging together
- How fast we adapted to change due to duplicate tickets & lack of descriptions in ADO
- significant improvement in team collaboration.
- Using new Objects in C#, Seeing things we never used in the language before, between the team
- The team demonstrated excellent problem-solving skills.

What actions will we take?

- Create bug tickets for the things that need to be fixed
- Define the scope, acceptance criteria and story point immediately when you make the ticket
- Digram the game loop, classes, and any interactions between classes
- Understand the high level implementation for a complete MVP before coding
- Ensuring every member in the group understands the requirements for the user stories, features, and epics
- Make sure that all specifications are detailed and clear to avoid confusion and misunderstandings.

[redacted]

Thank You