Adaptive Arena

Sprint 0

Meet The Team

Inna Ivashchenko
Project Manager / Developer

Johnhenry Brayman
Lead Game Developer

Shariah Allen
Game Developer / AI Engineer

Anthony Piombino
QA Engineer

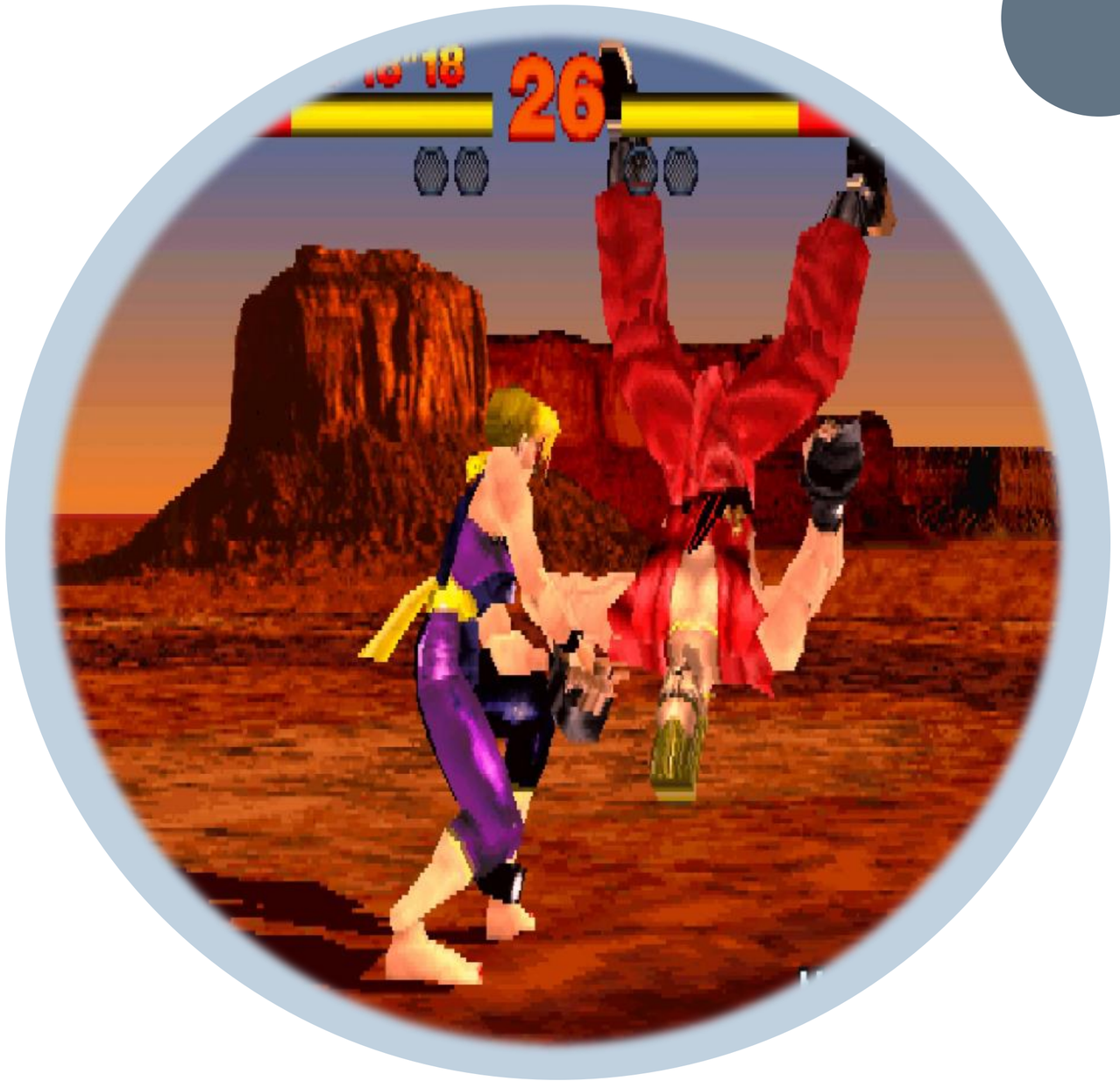
Gabrielle Schwan
UI/UX Developer

Agenda

- Problem Statement
- User Personas
- Project Description
- Project Timeline
- Technologies
- Algorithms
- Minimum Viable Product
- Working Agreement
- Sprint 0 Retrospective

Problem Statement

- Competitive fighting games are typically daunting for newer players since most skills are transferable across the different titles.
- At the same time, more experienced players lack a consistent opponent to learn from and advance their skills.
- How can we create a compelling game that scales to our player's skill level and creates enjoyment for all users while encouraging high replay value?



User Personas: The Competitive Gamer

Name: Anna Grace
Age: 28
Gender: Female
Occupation: Game Developer & Twitch Streamer
Anna is a season game developer with a solid fan base, that enjoys playing games in her free time, specifically fighting, first-person shooters (FPS), and racing games. She's often searching for new and exciting games that have varying difficult levels and unique features that keeps players engaged.

Challenges:
- Tired of playing the same popular titles, with reoccurring characters and stories
- Wants to find new games developed by small teams to stream for her community

Goals:
- She enjoys playing games and learning different mechanics
- Wants to explore new games with AI elements
- Seeks a game that can be replayed several times

Quote:

" I just want to entertain my community and learn new skills; I love a good challenge!"

User Personas: The Button Smasher

Name: Jordan Carter
Age: 19
Gender: Male
Occupation: Undergraduate Student and Student Athlete

Jordan is a busy student athlete studying animation, that loves to play games on his free time on the weekends or while hanging out with friends. He loves playing games with lots of action, unique characters, and prefers to not memorize combos but just plays for fun, not to win.

Challenges:
- Complex inputs and combinations makes gameplay frustrating
- Long or vague tutorials

Goals:
- Have fun solo or with friends
- Win occasionally, or access to varying difficulties

Quote:

" I skip tutorials, I just want to have mindless fun!"

User Personas: The Lore Explorer

Name: Dylan Taylor
Age: 12
Gender: Male
Occupation: Middle School Student

Dylan enjoys playing games after school with friends, family, or by himself, and specifically enjoys games that are action-packed. He seeks out games that provide a good challenge, has lots of enemies, and visual stimulation. He aspires to be a game developer and enjoys learning how to make 2D games during his video game club meetings.

Challenges:
- Hard to find action games that are age appropriate
- Wants games with unique enemies
- Prefers 2D games

Goals:
- Wants to explore new action games
- Age-appropriate game with visual stimulation

Quote:

“ Games are fun when they have a lot of action!”

Project Description

- 2D-fighting game focusing primarily on AI opponents and improving your general fighting game skills.
- AI will scale appropriately based on selected skill level
  - Easy : Mostly random or suboptimal moves
  - Medium : Basic strategy with a little randomness
  - Hard : Uses machine learning to adapt and respond
- Provide user-friendly interface for user entry
- Tally up and output results of game

Project Timeline

Sprint 0

- Determine project topic
- Determine job roles
- Create and update wiki
- Conduct video presentation and retrospective

Sprint 1

- Solidify project details and draft artifacts
- Demonstrate working interactive product
- Update wiki
- Conduct video presentation and retrospective

Sprint 2

- Draft tech paper and artifacts
- Demonstrate interactive product with improvements
- Update wiki
- Conduct video presentation and retrospective

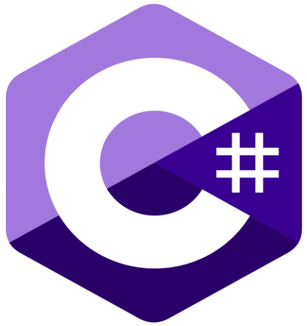
Sprint 3

- Demonstration of MVP
- Submission of final technical paper, deployment, and user manual
- Update wiki
- Conduct video presentation and retrospective

Technologies

Unity 6.3 LTS

Git LFS Extension



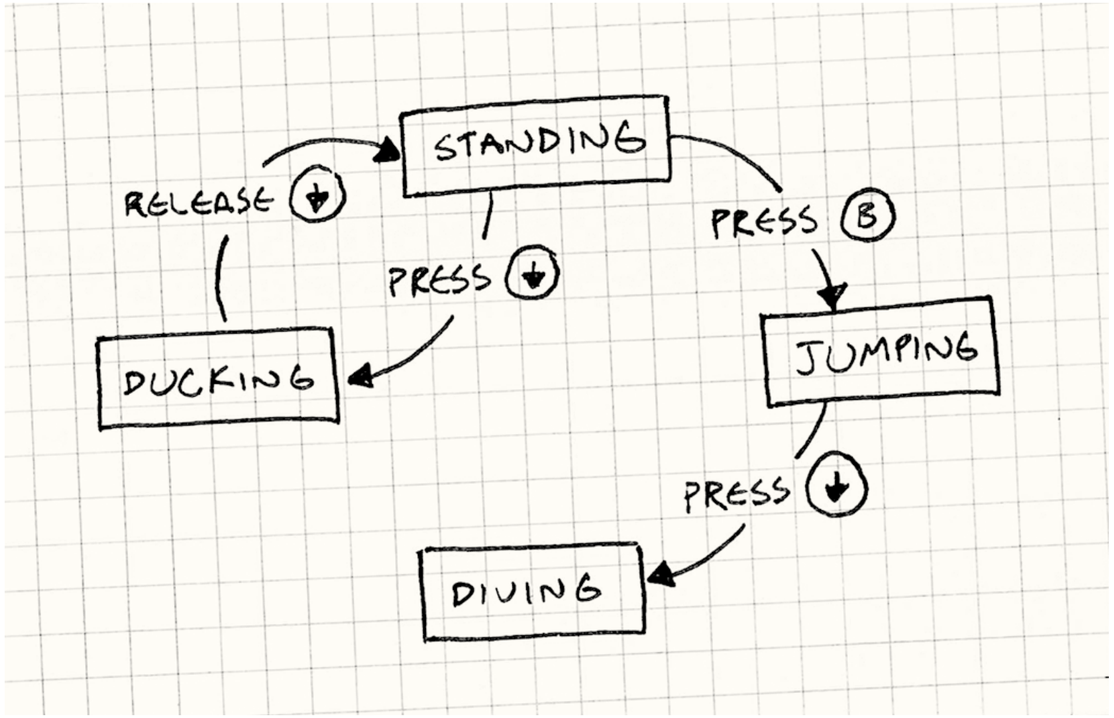
C# Programming

Python (for Machine Learning)

Algorithms

Easy Difficulty : Randomized Finite State Machine

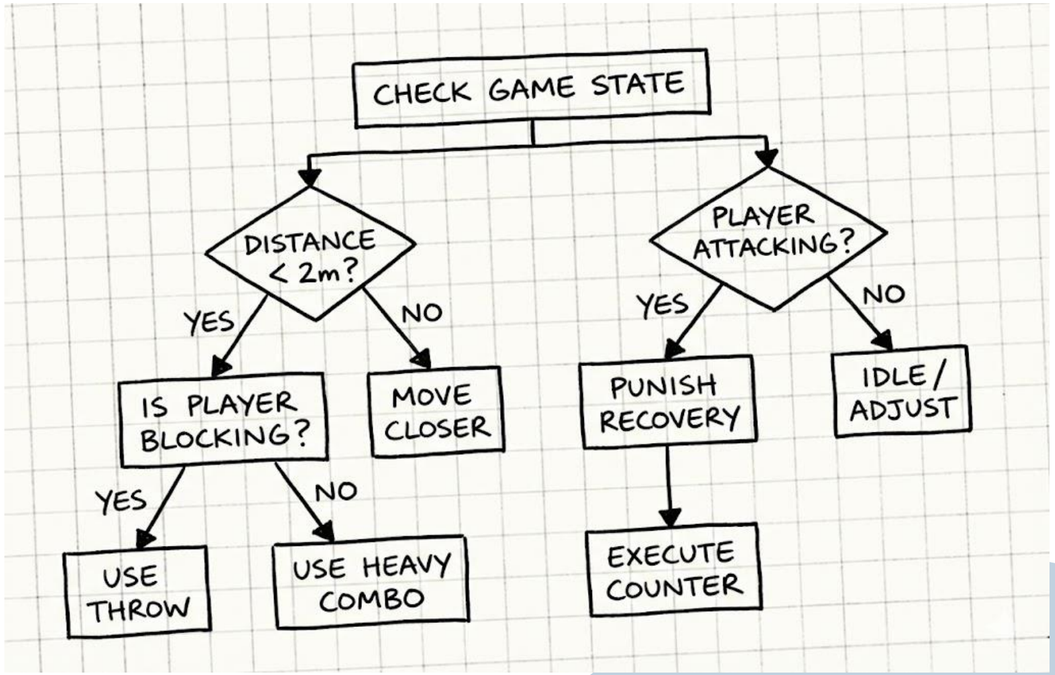
- The enemy AI is presented with multiple pre-defined states (Idle, Attack, Advance, Block, Duck).
- Instead of using strategy, it picks which state to move to next randomly.
- Attacks and movements still make sense but are suboptimal.



Algorithms

Medium Difficulty : Decision Trees

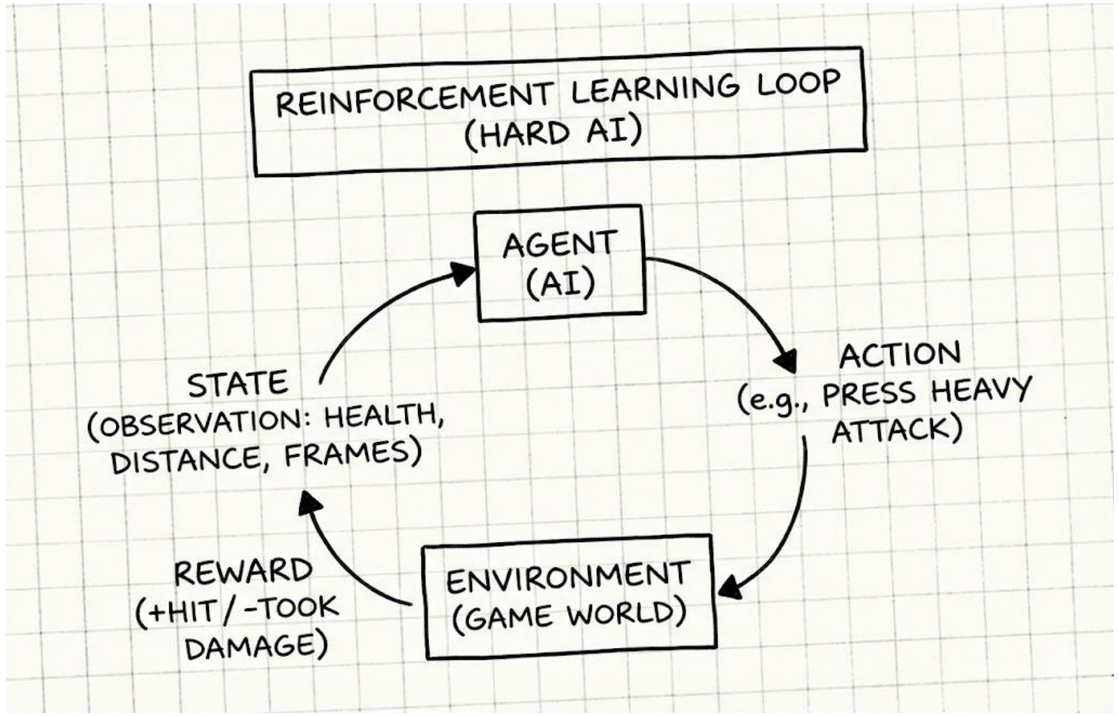
- The enemy AI in this difficulty will start to react differently depending on certain criteria.
- Before making an action, it will check a specific condition and act accordingly.
- Can still make random decisions but in a more educated way.
    - Ex. Check to see if the player is attacking. If yes, randomly choose between block or counter.



Algorithms

Hard Difficulty : Reinforcement Learning

- The enemy AI will fight through many simulations either against itself, or one of the before mentioned difficulties.
- It will receive a reward (+1) for landing a hit on the player, and a punishment (-1) for being struck by the player.
- The AI will learn which actions lead to rewards and which actions lead to punishments.



Minimum Viable Product (MVP)

The Environment:

- A simple flat ground with invisible walls on each side to prevent falling off.
- A basic camera with a script that keeps both players in-frame while zooming in as close as possible.

The AI Difficulties:

- For the rules-based difficulties, the State Machine and Decision Tree logic must work, even if they are not complex.
- For the hard difficulty, the Unity Agent must be able to read observations and output actions from the Neural Net.

The Combat System:

- The bare minimum controls will be Locomotion, an Attack action, and a Block action.
- Possibility to include combos, ducking, countering, and throwing later.

The Game Flow:

- The player must be able to enter the game from a main menu and select an AI difficulty.
- In-game must show health bars depleting, and a return to the main menu when one character reaches zero.

Team Working Agreement

Communication And Responsibilities:

- Communication through WhatsApp.
- All team members must respect each other and have the right to contribute their ideas to the project.
- All ideas, vision and changes discussed together, and final decisions are up to most of the team, except urgent situations.
- Each person works within the boundaries established before the start of work.
- Everyone is responsible for their own part.
- All conflicts should first be resolved between the people involved. If that is not possible the issue should be brought to the Team Lead.
- Each member must inform the team about any changes or improvements they make.

Meetings:

- Meeting happens through ZOOM or WhatsApp
- The team must have Scrum meetings two times in a week and check WhatsApp daily to avoid missing any project information.
- Schedule of a Scrum meeting : Monday at 10am and Saturday at 10am
- Each team member should participate in at least one meeting per week unless they have an important reason.

Retrospective

Team Strengths

- Open to each other’s ideas
- Team was flexible and responded quickly
- Prioritize tasks and follow the plan
- Everyone achieved their responsibilities
- Quickly adapt to new requirements

Team Weaknesses

- Communication in the team chat was not updated by everyone
- We must work on time management
- Learn new skills to improve the project
- Work structure was not created, and communication was a bit chaotic

Improvements

- Implement a framework for meetings (e.g., Trello)
- Improve team communication
- Participate in weekly Scrum meetings
- Use innovative tools and receive new knowledge
- Update task status at least twice a week

Wiki Page Link

https://github.com/htmw/2026SA-QuestKeeper/wiki

Thank You