Alexander Sosa: aasosa@stanford.edu
Andrew Lee: alee16@stanford.edu
Chris Billovits: cjbillov@stanford.edu
Garrett Gutierrez: garrettg@stanford.edu
Leah Kim: leahkim@stanford.edu

CS 194 - Project Proposal

By: Alexander Sosa, Andrew Lee, Chris Billovits, Garrett Gutierrez, and Leah Kim

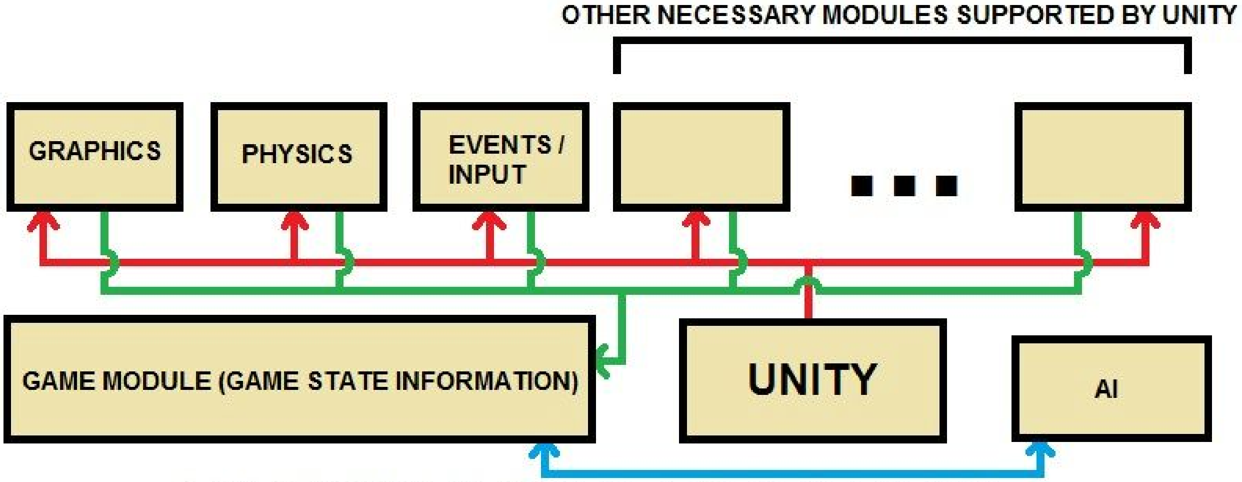
Table Of Contents:

I: Description of the Project:

We will be developing a real-time 3D fighting game in which a character's movement and attacks within a finite arena are controlled by the player's input. The goal of the game will be to defeat an opponent by depleting its health in a similar fashion to existing fighting games like Soul Caliber or Super Smash Brothers.

The primary playing mode will be a single player fighting against a single AI opponent. This AI will be designed in such a way that its skill is balanced against the player's ability so that the experience is both challenging and fun for the player, as well as similar to what a player would expect from fighting a human opponent of the same approximate skill level. The AI will adapt to the player's strategies and learn throughout the duration of a fight which attacks to anticipate and which to use in response.

From the users' perspective, they will be able to start the game up on the computer, navigate a simple and intuitive menu to start their play, and their use familiarity with conventional controls (WASD) or controller input in order to start playing relatively quickly. We strive to create a game that at its core is an enjoyable yet challenging experience. The main allure of the game should be reactiveness of the AI, in addition to the gameplay and mechanics of the game itself.


OTHER NECESSARY MODULES SUPPORTED BY UNITY
GRAPHICS
PHYSICS
EVENTS / INPUT
...
GAME MODULE (GAME STATE INFORMATION)
UNITY
AI

PRELIMINARY MODULARITY DIAGRAM

II: Need for the Product:

Video games have always been the pioneers in advancing computer AI capabilities, and therefore a great opportunity to explore the complexities and challenges that come along with creating a smart AI. Games have inherently also evolved to be more than just entertainment and "time-wasters" — they are now regarded as a complex and creative art form.

Our game will be valuable in two respects. One will be the pure entertainment that comes with playing a game. We hope to offer an immersive and fun experience, but one that does not require a steep learning curve. There will be a clear objective, and it is up to the player's creativity to determine how to fight the computer. The user should be able to spend a few minutes getting used to the gameplay and mechanics but then quickly feel challenged by the opponent in the game — the computer AI.

Along with a direct consumer desire for a challenging opponent in a video game, our game will address greater societal and intellectual needs through our research into constructing and developing a strong AI. One one hand, because games in general require strong AI for greater offline enjoyment, developing a "smart" enough AI will be a critical part in enhancing the product and increasing the attractiveness and playability of the game. We seek to define the optimal AI by addressing the need for research into preventing cheating AI, overly predictable (but difficult) AI, and arbitrary / weak AI, and present algorithmic solutions to implement in the next section:

AI Design Desiderata

1. **Cheating.** Cheating AI will adjust its attributes or perform actions beyond the capacity of a player in the user player's game logic. Cheating is bad because it breaks realism and the reasonability premise of a game. This also applies to AI difficulty that is coarsely exposed through broad attribute changes. In racing games, AI racers' tires often become magically stickier and the cars accelerate faster when they are far behind. They then can catch up quickly and stay close to your vehicle all race, an action that has earned the moniker "rubber banding." In first person shooter games, violating AI may be aware of your position through walls, or lock on to the player's head unnaturally fast and precisely, and mitigate recoil perfectly.

   A good AI does not cheat even when the player may not be able to infer cheating. Thus, as a guiding principle, our AI will not use or modify internal game states that cannot be exposed to or inferred by the player.

2. **Predictability.** Some AI are omniscient and destructive, yet have one well-defined weakness that must be exploited to subvert them. In fighting

games, cheating AI will be able to kill the player in one or two hits, except for some sequence of tells in which a certain sequence of actions will leave the player unscathed, as if the AI were incapable of moving the extra inch. On the most extreme end, old video game bosses behave more like a set of obstacles than a perceptive enemy. Predictability kills player creativity and forces rote repetition to defeat an enemy rather than promoting reasoning about the virtual world.

A good AI avoids predictability by being truly adversarial, and considering the user's action in a more holistic manner. In addition to reflex-based triggers, which drive most commercial game AIs, we will incorporate minimax and knowledge-based inference to learn truths about the video game world, and piece together higher-end strategies and principles. A global knowledge base computes transitive closures, enabling an AI to take a set of circumstances and operate off a larger set of facts.

3. **Arbitrariness.** Arbitrary AI is different from simply being unpredictable; cheating AIs don't provide a challenge and lead to player boredom. Usually arbitrariness arises from too much weight placed on random actions, or suboptimal actions. For agents whose actions depend on reinforcement learning, an overly high exploration vs. exploitation ratio value makes this glaringly obvious. Alternatively, agents following a game-theoretic strategy (i.e. minimax), when faced with a suboptimal player decisions, will be overly evasive and avoid exploiting the player.

   Our agent will remedy these concerns by pre-training agents against each other to reduce the default exploration to exploitation ratio, so that game policies are mostly set. AI can identify abstract matchups and player strategies using latent variable machine learning models early on in the game. This will allow for real-time adaptation to the player's strategy without making oddly specific (poor) moves.

III: Potential Audience:

Video games have become ubiquitous in the 21st century. They serve as ice-breakers and communal activities at parties and large gatherings. They serve as individual outlets for use of strategy and skill, as well as feelings of accomplishment and entertainment. And they also serve as artistic and immersive experiences into the reality of the world created by the authors. Our audience will be in the second category since we want to provide users with a challenging but fun experience that allows them to explore their creativity in strategy-development, the skill-building of getting used to and attuned to mechanical input movements in the game, and the feelings of entertainment and accomplishment as they battle in the game and eventually find ways to best the computer AI.

By the nature of the product we are making, we serve a fairly large audience. They will range from hardcore gamers who want a challenging experience that looks and feels real and casual game enthusiasts who simply want an enjoyable experience to people who don't usually play video games but may give our game a chance due to its accessibility and shallow learning curve. Finally, there may be an audience of AI-enthusiasts who simply want to see how the AI observes and reacts to their gameplay.

Clearly, the level of technical sophistication of our audience will vary a lot— we require no prior knowledge of video games (beyond perhaps standard controls), of AIs and how they behave, or even of fighting-style video games in general. We want to create a product that is available to a wide range of players, from novices to experts and developers interested in the internals of the game.

IV: Discussion of Competing Products:

Super Smash Brothers has set a precedent for a competitive, multiplayer video fighting game that is available for a wide range of ages for people to enjoy. Soul Caliber, Mortal Kombat, and Street Fighter, as well, are known as mainstream fighting games between two opponents.

Super Smash Brothers and Soul Caliber both focus on multiplayer play, with the main mechanic being that you are allowed to fight against your friends. Super Smash Brothers has released a number of campaign single-player modes with each release. However, the AIs themselves in the gameplay of a 1-1 fight are not very strong. As the player advances in skill, the campaign relies on other characteristics of enemies that break the rules of engagement: they are larger, have an expanded moveset, or have special abilities.

Our product strives to be different in the approachability of the game — we want it to be intuitive and quick to pick up. Furthermore we will focus on developing an AI that lets the player reason logically about the game world under our optimal design principles.

Like any other video game, ours will also differ from competing products simply because the style and design of our gameplay that we choose will be different from any existing game. The artistic aspects will be unique, as will be any story and graphics that we choose to bind to it. Naturally we hope to create a gameplay experience that feels novel as well; familiar enough to existing fighting games that the object is clear and the controls are intuitive, but with interesting constraints and mechanisms so as to provide a new challenging experience that appeals to many people who like the challenge and thrill of trying to beat a new game.

V: Major Technologies Used:

Our project will be built in Unity, which will serve as the engine for the game. For certain algorithms in the game such as those related to AI, Java or C++ plugins will be used assuming that doing so is feasible and results in appropriate performance. Otherwise the game will be scripted in C# or JavaScript. Git will be used for version control and to distribute changes that we have individually contributed. As we are aiming to have players be able to play locally on their machines, we will not need extensive external resources such as databases or web services.

VI: Potential Approaches:

Our end goal is to create a game that is fun to play, provides an engaging experience, and maintains the player's interest. To do this, we would like to target primarily the creation of a challenging AI. However, there are several other approaches we could take to make the game appealing:

- **Polished and well-developed graphics.** Since the game will consist of an almost completely-visual medium, along with game sounds and effects, the visual appeal of the game will be an important factor in determining how engaged a player is with the game experience. Satisfying animations and fluid movements can contribute a lot to the value of the experience.
- **Complex game mechanics.** The game, at its most core functionality, must still be playable and offer some sort of novel challenge. This approach would focus on implementing a very complex set of game mechanics and strategies for the player to master. Akin to competitive Super Smash Brothers Melee, we could focus exclusively on precise timings and inputs that the player would learn to utilize, thus providing an ever increasing challenge for serious gamers to pick up and compete with.
- **Multiplayer competition**. Another approach to making a game appealing is to introduce the notion of collaborative play. Games that are localized to single players on their personal machines are suddenly much more interesting when a large number of people can play at once or interact with each other in some way.

While there are clearly different approaches to making a good game, we have decided to focus on developing an engaging AI as the primary way to engage players, maintain interest, and provide a satisfying experience. One reason is that three out of the five members of our team have expertise or interest in AI development. Another equally important reason is our goal to make the game accessible for a wide range of audiences. Having a steep learning curve would detract many people from simply picking up the game and playing. Likewise, only a small subset of people would enjoy having a game with beautiful graphics but unsatisfying gameplay.

The next level of potential approaches will be in determining what sort of algorithms we will use, and ultimately how we construct the model for a satisfactorily-challenging AI. As discussed in our ”Need for the Product” section, there are also several approaches to developing a better AI. We will experiment with different methods until we find the one that best synergizes with our existing gameplay and rules, and one that best adheres to our goal of creating a successful AI: one that is adaptive, one that can closely mimic the rationality and strategic reasoning that humans deploy, and especially one that does not need to resort to cheating in order to be challenging. Our justification for pursuing AI development as core to our approach is simple: we wish to create the best playing experience for our users. Not too many games are able to provide the player with AIs that are smart enough or challenging enough; we want our users to believe they are playing not against a simple or predictable computer but a smart agent that lives up to its name.

VII: Assessment of Risks:

Graphics and Storyline: Getting graphics to look exactly as desired can be a very challenging task; if we were to focus our time on fixing every small graphics issue that presented itself, we might not be able to fully flesh out more important structural features of the game. The same can be said for creating a storyline for the game: creating an interesting and creative story behind the game could take a long time. Because the focus of our project is on developing a game with strong, realistic computer opponents by utilizing artificial intelligence techniques and not on developing the most beautiful or intellectually stimulating game possible, it is thus important to think of graphics and the storyline as secondary objectives, ones to be perfected only after our primary goal has been met. If we finish the AI and general game logic early, we can put our remaining time into improving the look of the game.

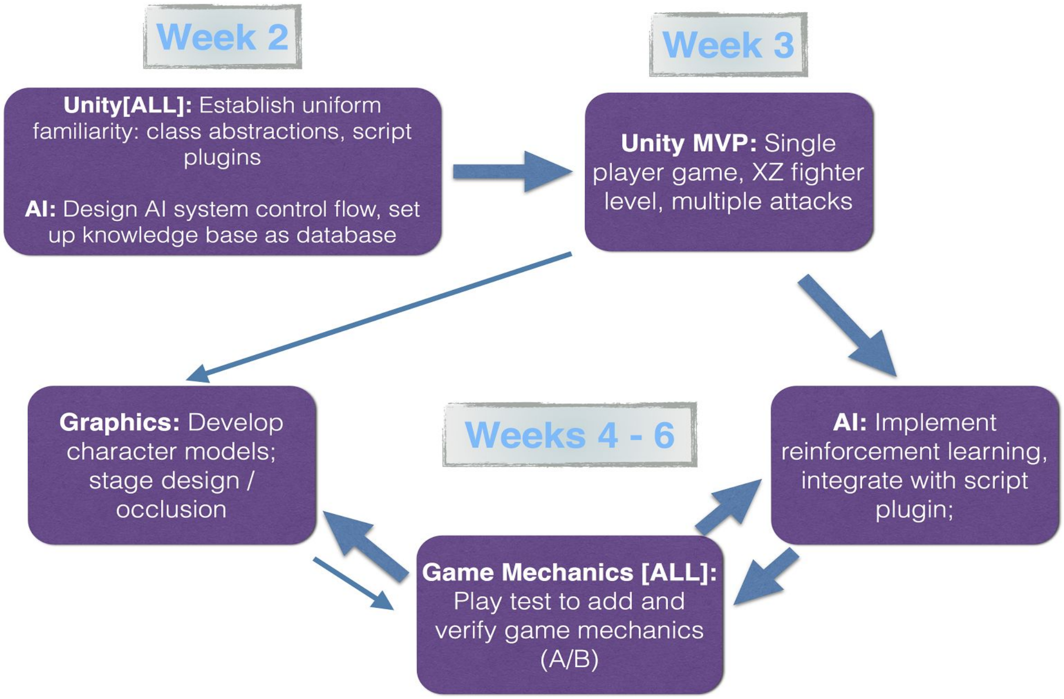
Feature Creep: It can often be enticing to continually add new features to a game without having finished previous add-ons or even the core of the game. Diverting time to extraneous features which would add to a fully developed game runs the risk of not allowing enough time for the development of the core game logic and our artificially intelligent opponents. It is thus necessary to focus first on the basic structure that allows a rudimentary version of the game to be developed before increasingly adding features.

Parallelization: Because graphics and game logic are seemingly so disjoint, it can be appealing to work in two separate groups and try to reconcile code at the end. However, this presents a major risk. Fitting the two completed pieces together may not be simple. There may be features in the game which cannot easily be translated into the graphics engine, and certain gameplay features might not immediately sync up well with the fully developed graphics. It is thus necessary to continually work to integrate gameplay with the graphics framework together throughout the development of the game, not allowing the development to reach a point where the two are irreconcilable and a large amount of work must be redone.

Modeling Software Choice: Depending on which software we use to create our game's graphics, we might be left with too little freedom on the gameplay development side or have to spend time learning a new engine. Unity might force us to use C# to write all of our game and artificial intelligence code, which could be a hassle to learn if we cannot easily avoid using the language. However, learning another graphics software that offers more flexibility might take a long time. This tradeoff between the ease of integrating the graphics engine with our code for the AI game logic and having to learn a new system for the graphics development is a tricky one.

Large Group: Having more group members increases the potential for disagreements between group members on how best to solve a certain problem. This can be alleviated by instituting a voting system to reach decisions. Also, having such a large group can lead to procrastination by members of the group who believe that the other members can pick up their slack. However, this issue can be fixed by setting clear goals for each member of the team to achieve by certain deadlines. With such deadlines, it is hard for a member to slack off without the group noticing and intervening.

VIII: Next Steps:


Week 2
Unity[ALL]: Establish uniform familiarity: class abstractions, script plugins
AI: Design AI system control flow, set up knowledge base as database
Week 3
Unity MVP: Single player game, XZ fighter level, multiple attacks
Weeks 4 - 6
Graphics: Develop character models; stage design / occlusion
AI: Implement reinforcement learning, integrate with script plugin;
Game Mechanics [ALL]: Play test to add and verify game mechanics (A/B)

Three of our team members have strong expertise in AI-related fields and two in 2D games, but only one has had experience developing a 3D game within Unity. Our chief objective is to get working with Unity and have a hands-on feel for the extent, abstractions and modularity of each component, so that we can all understand how to integrate our code.

By the end of Week 3, we want to have the minimally viable product in the Unity engine. It will have the minimum baseline feature set to develop nontrivial AI based on machine learning. It should permit one character type with 3 fighting attributes in an XZ stage against an identical opponent.

From there, in weeks 4 - 6, we will jointly iterate on game mechanics, which then direct our progress in graphics and AI. We will then split up according to experience and interest — potentially having three members working on developing the algorithms for the AI and having two team members taking the lead of creating graphics and story.

Features will be discussed and only pursued if we believe them to be central to our core gameplay and playing experience. Beyond the 6-week mark, we plan to incorporate feedback and assess our current progress to refine our goals for the rest of the quarter.