PreGLAM: A Predictive, Gameplay-based Layered Affect Model

Cale Plut, Philippe Pasquier, Jeff Ens, and Renaud Bougueng

***Abstract*—We present the Predictive Gameplay-based Layered Affect Model (PreGLAM), an affective game spectator model that flexibly integrates into a game design process. PreGLAM combines elements of real-time Player Experience Models, and Affective Non-Player-Character models to output real-time estimated values for a spectator's valence, arousal, and tension during gameplay. Because tension is related to prospective events, PreGLAM attempts to predict future gameplay events. We implement and evaluate PreGLAM in a custom game *Galactic Defense*, which we also describe. PreGLAM significantly outperforms a random walk time series in how accurately it matches ground-truth annotations, and has comparable accuracy to state of the art affect models.**

***Index Terms*—Game Emotion, Affect, Player Modeling**

I. Introduction and Motivation

A. Motivation

Gaming is, among other things, an emotional experience, and there are many potential benefits to modeling emotional responses a game's play. Most game affect models focus on the player's emotions, for example Experience-driven Procedural Content Generation (EDPCG) often creates game content based on a Player Experience Model (PEM) [1]. Generative music systems often control musical emotion to match a PEM [2], [3]. Serious games occasionally use PEMs to enhance the learning process and performance [4]. Outside of modeling players, Non-Player Characters (NPC)s occasionally use emotion models to inform behaviour [5].

In addition to active game agents, games can be watched by passive agents, such as spectators. Spectating is a popular way to consume game content — In 2021, global viewers watched a total of 8.8 billion hours of video game live streams [6]. In 2020, the "League of Legends" Championship finals were watched by a peak of 4 million concurrent viewers, and a total of 45 million viewers [7].

Spectating views an interactive game in a linear fashion, and may have similar experiential qualities to watching a film. Film music's capability to evoke emotions in its viewing audience is well-established [8], and watching a video of gameplay may have a similar effect. Previous research indicates that game players perceive and appreciate when music adapts to match a game's emotion [9].

Affective spectator modeling may also provide insights into PEMs. Holm et al. find similar physiological responses between players and spectators of First-person Shooter (FPS) games [10]. Sport spectators show stronger emotional responses that are more similar to outcomes from personal actions when spectating games compared to non-game entertainment media [11]. While the experiences differ in other ways, the emotional responses of spectators and players are similar, especially with spectators who have game knowledge.

In terms of application, spectator emotion models could be used for diagetic audience reactions in games that simulate a real-world or fictional sports, such as *FIFA Madden*, *Rocket League* or *Blitzball*. Spectator modeling in e-sports games may assist camera operators [12], [13], casters, and analysts in automatically recognizing highly emotive moments.

PEMs are ocassionally used to control generative game music systems [3], [14], though modeling a spectator's emotional perception may be better suited to this task. Winifred Phillips describes one primary function of music as acting as "an audience" [15], responding to a perception of the gameplay experience. We extend this metaphor, and use a spectator emotion model to control the adaptivity of a musical score.

B. PreGLAM

We present the *Predictive Gameplay-based Layered Affect Model* (PreGLAM). PreGLAM is an artificial cognitive agent that models the real-time perceived affect of a biased game spectator. PreGLAM's design is based on real-time generic gameplay-based PEMs and affective NPC models, which we discuss in Section II. Gameplay-based PEMs generally output some real-time estimation of player experience, based on gameplay events and variables, often related to affect or emotion [16], [17], [18].

Because we are modeling a spectator, who may have different desires than the player, we take inspiration from NPC models of affect. These models, such as GAMYGDALA [19], generally output some estimation of an NPCs affect, based on the player's dialogue choices. These models accommodate a range of NPC personalities, goals, and desires. These models often extend the OCC model of affect, named for its creators Ortony, Clore, and Collins [20]. The OCC model primarily describes emotions as valenced responses to events that affect the subject, with the response depending on how the event affects the subject. In the OCC model, tension arises from the expectation of prospective, or future events. To model tension, PreGLAM estimates prospective game events.

PreGLAM outputs a real-time estimation of a game spectator's perceived emotion, based on gameplay events and variables. PreGLAM models a spectator who is biased towards a provided outcome — PreGLAM is provided with a desire in the form of a game outcome, but cannot affect the game world. For our purposes, we model PreGLAM with a desire of the player winning, but any quantifiable outcome could be used. Multiple PreGLAM instances could be simultaneously

implemented with multiple desires, such as to model a spectator biased towards each individual team in a competitive multiplayer game.

PreGLAM's knowledge base is provided as a table of **Emotionally Evocative Game Events (EEGEs)**, which are events that impact PreGLAM's desire, and are described in Section III-B. PreGLAM calculates its belief at each time step by aggregating past and predicted EEGEs, as discussed in Section III. PreGLAM outputs real-time unbounded valence, arousal, and tension values, which describe the current estimate of a spectator's perceived emotion from the gameplay, incorporating predicted events.

To implement and evaluate PreGLAM, we create an action-RPG genre game titled "Galactic Defense" (GalDef), which is further described in Section V. We discuss the specific implementation of PreGLAM into GalDef in Section V-B, and other theoretical implementations in Section IV. We empirically evaluate PreGLAM by comparing PreGLAM's output annotations to ground-truth annotations from human spectators using Dynamic Time Warping (DTW) [21], which is discussed in Section VI. PreGLAM outperforms a random walk baseline across all experimental conditions, and in dimensions of arousal and tension, and insignificantly outperforms the random walk in the dimension of valence.

Compared to previous PEMs, PreGLAM presents several advantages. ML-based PEMs are often trained on game data and corresponding user annotations, which can be expensive. While some ML-based PEMs target generic applications, these models generally require individual training and tuning to be implemented into any particular game [18], [17]. While we do not implement ML prediction here, an ML extension of PreGLAM may be trained on game data alone, without the need for annotations. Additionally, PreGLAM models more emotional dimensions than most previous PEMs, providing a more detailed description of affect. Finally, PreGLAM is flexible across a wide range of game genres and mechanics.

II. Background

A. Player experience models

There are many approaches to modeling game experience. Experience-Driven Procedural Content Generation (EDPCG) use Player Experience Models (PEMs), often to evaluate and generate game content to evoke a particular player experience [22]. PEMs may be used in game testing, to evaluate specific game features [23], [24]. Dynamic Difficulty Adjustment (DDA) uses a game experience model to adjust a game's difficulty during gameplay [25], [26]. PEMs are also often used in live service games to increase player retention [27], [28], [29].

PEMs that model based on real-time game data generally follow a reactive, feature-based approach. Camilleri, Yannakakis, and Liapis present a general model of player affect that models affect based on game-specific *goal-oriented* and *goal-opposed* events, alongside a set of generic features common to their included games [17]. Melhart et al. present the "Arousal Video Game AnnotatIoN (AGAIN) Dataset", which contains game emotion annotations and feature data across nine custom games from three genres [18]. Feature data is separated into 23–47 specific features per game, and 14 general gameplay features that are shared between the included games.

Gameplay-based PEMs are often used to influence the real-time generation of music. Plans and Morelli target using a PEM to influence the emotions of a player towards maximizing "fun" [30]. Separate systems by Prechtl and Plut use PEMs to estimate a tension value to adjust adaptive music [3], [9]. Systems by both Scirea and Williams et al. implement a PEM to control musical generation with emotional dimensions of valence and arousal [2], [31]. The "Adaptive Music System" (AMS) [32] uses a spreading activation model to influence the affective expression of a generative score. In our implementation, we use PreGLAM's output to influence the adaptivity of a generative score, following Winifred Phillips' metaphor of music acting as an "audience" [15]. The use of PreGLAM to control the musical adaptivity is described in a separate paper [33].

B. Affective Non-Player Character (NPC) models

An affect model may allow NPCs to react to the world and gameplay in emotionally informed ways. Cognitive appraisal models of emotion, such as the OCC model [20], are common in modeling NPCs [34], and describe emotions as the arising from an evaluation of how emotionally evocative conditions affect the subject.

We are aware of three systems for NPC design that implement the OCC model, appraising emotion based on the events that occur during play. *ALMA: A Layered model of affect*, attempts to provide emotional reactions in conversation [35]. *GAMYGDALA* focuses on providing a flexible emotion model across a variety of NPC interactions [19]. *EMoBeT* builds on the architecture of *ALMA* by using the emotional model to control behaviour [36].

Across these implementations, game designers provide specific details on NPC goals or wants, and on the possible game events that can impact those goals/wants. ALMA appraises "relevant input" for all characters, using provided appraisal rules for each character. To use GAMYGDALA, designers provide explicit sets of goals and events that are relevant to those goals, and GAMYGDALA appraises game data based on these goals and events. For EMoBeT, developers provide a set of possible emotional responses and events that trigger these responses, as well as behaviour trees that dictate the NPC behaviour based on modeled emotions.

In addition to applying the OCC model, both ALMA and EMoBeT use a layered representation of affect, providing a mood based on designed personality traits and values. The results of the appraisal process then layer an emotion value on top of the mood value, to reflect the longer-term affective states that can influence emotion.

C. Affect representation

1) Mood and Emotions: Affective experiences of mood and emotions are primarily differentiated by their duration and whether they are a reaction to a particular stimulus [37]. Emotions are triggered by an internal or external event [38],

and are short-lived, generally lasting seconds to minutes [39]. Emotions are generally perceived as changing in intensity over time. Frijda describes the chart in Figure 1 as demonstrating the most common perception of how an individual emotion will rise and fall linearly over time [40]. In contrast, moods are described as diffuse and global, generally last longer, and are not elicited by any particular event.

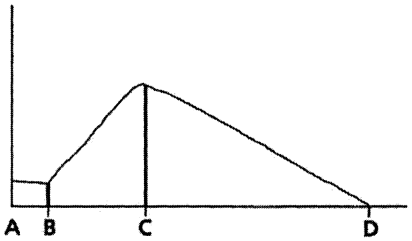

Figure 1. Frijda's graph of the strength of perceived emotions rising and falling over time [40]

In addition to fading in time, emotions are contextual. Emotion classification on facial expressions is more accurate in video form than image form [40], and there is an increase in use of contextual data for automatic emotion recognition in both visual and audio formats [41], [42]. This indicates that emotions are often perceived as changes, rather than an absolute value.

2) VAT Model: We use a 3-dimensional "VAT" model of affect shown in Figure 2, with dimensions of Valence, Arousal, and Tension. Figure 2 also places a subset of common categorical emotions in their approximate relative VAT location, as examples. Our VAT model is based Schimmack and Grob's dimensional model of Valence, Tense Arousal, and Energy Arousal [43], modified for simplicity, parity, and to bring the language into line with common game music terminology. Previous comparisons of affect models for multimedia content analysis demonstrate support for a three-dimensional VAT model [44]. As mentioned in Section I, while other applications of the OCC model generally use a 3-dimensional model of affect using Pleasure-Arousal-Dominance, we believe that the dimension of "dominance" is not applicable when modeling an NPC that does not have any agency to affect the game.

3) Valence: Valence is often paraphrased as "pleasantness", and describes whether an affect is "positive" or "negative". The OCC model primarily details the valenced reactions to emotionally evocative stimulus [20]. Positively valenced events are events that are desirable for the subject. In games, participants identify positive valence events and actions in a racing game as passing another car, improving their position in the race, or making another car crash, while negative valence events were being passed, being hit by another car, or driving off the course [45]. We note that unpleasant emotions experienced or felt during gameplay may be later appraised as positive experiences [46].

4) Arousal: Arousal, sometimes paraphrased as "activity" or "energy", is the dimension of activation. Examples of high-arousal emotions are excitement and fear, while examples of low-arousal emotions are calmness and sadness. Emotions

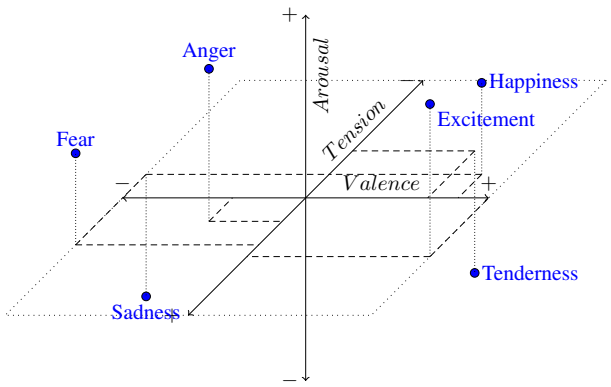

Figure 2. 3-Dimensional VAT Model of Affect, with example emotion categories placed for reference

with high arousal are most associated with physiological changes such as increased heart rate, blood pressure, and increased electrical activity in the brain [47].

5) Tension: Tension is unique in that it necessarily involves the prospect of a future event. Tension is closely related to valence, but distinct — valence is often associated with the desirability of an associated event, while tension is associated with the prospect, or prediction of the event. Tension may be high when anticipating good news, resulting in positively valenced tension, or bad news, resulting in negatively valenced tension.

The OCC model describes tension as arising from prospective events — a subject believes that an event is likely to happen, and has an emotional response depending on the desirability of the prospective event for the subject. As an example of an in-game prospective event that an audience may react to, consider attack telegraphs in MMO games. Figure 3 shows a set of attack telegraphs from the MMO *Wildstar*. Highlighted areas on the ground indicate that an enemy attack will be incoming in soon, and will hit any players who are standing in the highlighted areas. This effectively communicates a prospective event of incoming attack, and a related increase in tension as to whether the player will avoid the incoming attack.

Figure 3. Attack telegraphs from *Wildstar* [48]. The red areas indicate that an attack is coming soon

III. PreGLAM Framework

As mentioned, PreGLAM models the real-time perceived affect of a game spectator using a 3-dimensional VAT model, based on the actions and events of gameplay. PreGLAM blends approaches from real-time generic models of player affect, such as Camilleri, Yannakakis, and Liapis' model [17], with the flexible perspective of affective NPC models such as *ALMA* [35], GAMYGDALA [19], and *EMoBeT* [36]. To account for musical expectancy, PreGLAM also models the emotional response to prospective events.

PreGLAM is provided a gameplay desire, which is any quantifiable gameplay outcome. PreGLAM is given a set of events, which are annotated based on how they affect the given desire, how strong their effect on the desire is, and the values of variables that provide gameplay context. PreGLAM continues the layered approach seen in ALMA and EMoBet, and models emotion in relation to a longer-lasting mood value.

PreGLAM layers its event-focused appraisal model of emotion onto an environmental mood value, to produce a single output affect value per dimension every 250 ms. PreGLAM calculates an emotion value from both events that have occurred, and events that are predicted to occur. PreGLAM scales emotion values through time, to represent the rise and fall of emotions over time. Figure 4 shows how PreGLAM mimics the emotional appraisal process of a spectator, but does not directly influence the gameplay loop.

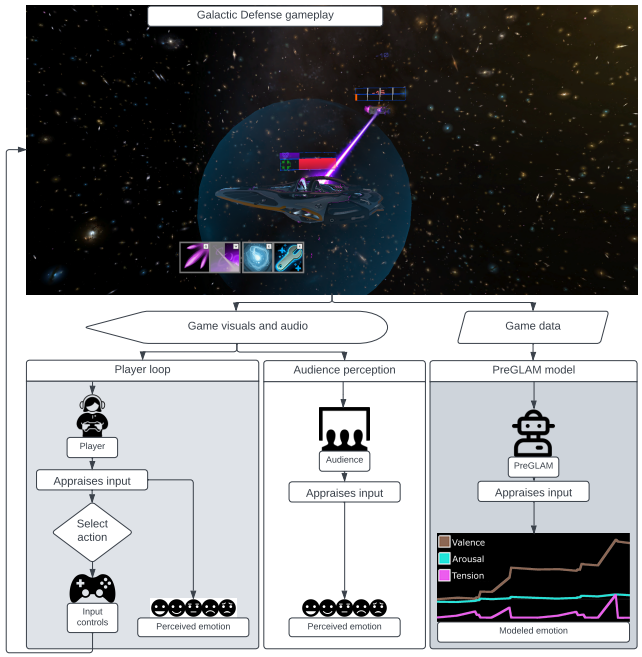

Figure 4. Diagram of PreGLAM's relation to player, audience, and game. Note PreGLAM does not directly interact with the player or game

We implement PreGLAM into our game *Galactic Defense*, as described in Section V-B. For our implementation, we provide a baseline value of 1 as the minimum value for emotional impact, and scale all other values based on the baseline value. We implement intensity modifiers that scale these values between 100–200%, depending on the values of the intensity modifiers. For example, in our game, both the player and their opponents are protected by a recharging shield, which takes damage instead of health. One intensity modifier for the basic attack EEGE is the opposing shield's percentage — as the shield approaches 0%, the intensity modifier approaches 200%.

A. Mood

As mentioned in Section II, moods are longer-lasting and more general affective experiences than emotions, and are not connected to any particular source. Mood is therefore related to longer-lasting, environmental, macro-levels of gameplay. At any time, PreGLAM can be given a mood value for valence, arousal, and tension. These values are arbitrarily defined by the game designer — to maximize PreGLAM's flexibility, we avoid making assumptions about any game mechanics or relationships. The mood value represents the designer's intended mood for a particular section of gameplay, or area. The only programmatic restriction for mood values is that PreGLAM only outputs values every 250 ms, and will not reflect changes to mood values that are faster than this time step. Given the longer-term nature of mood, we recommend that mood values remain relatively stable during gameplay.

One possible source for mood values may be environment. For example, depending on a game's design, a "snow"-themed area may have a lowered arousal than a "fire"-themed area. In this case, PreGLAM values could be set when an area is loaded. Another source may be particular enemy types — bosses may have elevated tension, and therefore the mood value could be set when boss combat begins. Another source may be player resources — the valence value may change when the player uses a potion, with a value depending on the number that remain. In our implementation, the mood directly is set by the expected difficulty of each gameplay stage, determined by the player's and their opponent's attributes. The mood value provides a base perceived level of affect for PreGLAM, and if no EEGEs are being modeled, PreGLAM will output the mood value.

B. Emotionally Evocative Game Events

PreGLAM's appraisal of EEGEs drives the moment-to-moment reactions to gameplay. As mentioned in Section I-B, PreGLAM models two types of EEGEs: Past and Prospective. EEGEs that have already occurred in the game are past EEGEs, while EEGEs that have not occurred yet are prospective EEGEs.

All EEGEs are provided a set of variables as shown in Table I, which also shows the variables that PreGLAM calculates for each EEGE. EEGEs are provided a name, base emotion value v, a set of context variables $\{c_1, ..., c_n\}$, and a time ramp value r. In terms of design, EEGEs extend Camilleri, Yannakakis, and Liapis' *goal-oriented* and *goal opposed* game features [17].

Because PreGLAM models a spectator, who may have desires that differ from the player's goals, v is based on how each EEGE affects PreGLAM's given desire. This is inspired by NPC models that respond to how events affect each specific

Table I
EEGE VARIABLES

Provided variables		
Variables	Symbol	Relevant Section
Name	N/A	N/A
Base emotion value	v	Section III-B1
Context variables	$\{c_1, ..., c_n\}$	Section III-B2
Time ramp value	r	Section III-B3
Calculated variables		
Variables	Symbol	Equation
Dimensional emotion value	v_{dim}	Section III-B1
Intensity modifier	mod	Equation 1
Time scalar	s	Equation 2a
Emotion value	e	Equation 3

NPC's goals [19], [36], rather than only modeling player goals. If an EEGE positively effects the chosen desire, v will be positive value. The specific value of v is determined by how strongly the EEGE will affect the desire, and is used to calculate v_{dim}, a base emotion value for each VAT dimension.

PreGLAM derives a single intensity modifier mod for each EEGE from the provided set of context variables $\{c_1, ..., c_n\}$ using Equation 1. At each time step t, PreGLAM calculates a time scalar for each EEGE s_t based on the time passed since the EEGEs creation p_t and the provided time ramp value r, using Equations 2a or 2b, depending on whether the event is past or prospective. PreGLAM calculates an emotion value e_{dim_t} for each dimension, for each EEGE, using Equation 3.

$$mod = 1.0 + \frac{1}{n}\sum_{i=1}^{n} c_i \tag{1}$$

$$s_t = 1.0 - (p_t/r) \tag{2a}$$

$$s_t = p_t/r \tag{2b}$$

$$e_{dim_t} = v_{dim} * mod * s_t \tag{3}$$

1) Base emotion value: All EEGEs are provided a base emotional value v, and PreGLAM calculates v_{dim} for dimensions dim of valence, arousal, and tension. We base all values on an arbitrarily defined unit of 1, though any consistent scale may be used. The base value represents the emotional perception of the event when it occurs, if no intensity modifiers are increasing the emotional intensity. We directly assign emotional values, but emotional values could also theoretically be derived from ground-truth annotations.

The value of v is derived from how strongly the EEGE affects the desire. If the EEGE positively effects the given desire, v will be positive. The base value for valence $v_{valence}$ is equal to the base provided v. In our implementation, $v_{arousal}$ is a constant 1 for all EEGEs. During development, we informally experimented with assigning varying levels of base arousal to events, but found that a single, static arousal level more closely reflected our emotional perception. This represents arousal as a measure of overall game activity, as $v_{arousal}$ directly scales with the number of EEEGs that are occurring. Tension is only calculated for future events, and is positive regardless of whether the outcome positively or negatively affects the events. For past events, $v_{tension} = 0$, and for future events, $v_{tension} = |v|$.

2) Context variables and intensity modifier: EEGEs are provided a possibly empty set of context variables $\{c_1, ..., c_n\}$, which are given as a percentage of a given game variable. These variables describe the context of the EEGE, and modify the intensity of the emotion. For example, a player attack in the context of a full-health opponent may result in a less-intense emotional perception than a player attack in the context of a nearly-defeated opponent — lightly hitting an opponent is assumed to be less intense of an emotional experience than knocking them out. PreGLAM calculates a single intensity modifier value mod for each EEGE using Equation 1.

3) Time ramp and scalar: As discussed in Section II-C, emotions rise and fall in intensity over time in a generally linear fashion. EEGEs are assigned an initial time ramp value r when created, which represents the duration over which the EEGE's emotion value will smoothly ramp through time.

At each time step t, PreGLAM calculates time scalar s_t for each EEGE using Equation 2a for past events, and Equation 2b for prospective events, where p_t is the time that has passed since the event's creation. In other words, as an event approaches the present, it's time scalar approaches 1.

C. Output

At each time step t, PreGLAM calculates a single emotion value per dimension e_{dim_t} for each EEGE using Equation 3. PreGLAM then calculates an output affective value for each dimension a_{dim_t}, based on the current provided mood value m_{dim_t} and set of emotional values from all EEGEs $\{e_{1_{dim_t}}, ..., e_{n_{dim_t}}\}$, using Equation 4. We note that while PreGLAM's output is technically unbounded, values always trend towards the provided mood value over time.

$$a_{dim_t} = m_{dim_t} + \sum_{i=1}^{n} e_{i_{dim_t}} \tag{4}$$

PreGLAM uses the "Grapher" plugin for Unity [49] to create a graphical output and save all affect values to .csv format. While PreGLAM performs calculations on every frame, we sample the output every 250 ms. This sample rate is chosen due to Unity's inconsistent timing and to synchronize with annotation software for empirical evaluation, discussed in Section VI.

IV. USE-CASE EXAMPLES

To demonstrate the generalized applicability of the PreGLAM framework, we describe theoretical implementation in two highly dissimilar games: *Dark Souls*, a dark fantasy, mostly single-player action-RPG known for a high degree of difficulty, and *The Sims*, a casual life simulation game. These games are dissimilar enough that any universal assumptions about specific emotional implications of any game mechanics are unlikely to apply in both cases. We demonstrate that PreGLAM provides more flexibility in modeling diverse mechanics than current state of the art models [18].

As mentioned in Section II-A, previous generic models generally include some set of "generic" features. For example, the AGAIN database includes "score" [18], and Camilleri, Yannakakis, and Liapis include a generic value for "number of enemies engaged with the player" [17]. We note that neither *The Sims* nor *Dark Souls* measures a score, and that there are no enemies to engage with in *The Sims*. Another example of this possible mismatch is that we argue that the number of enemies engaged with in *Dark Souls* is a poor predictor of player experience, given how often the player is in one-on-one scenarios.

A. Dark Souls

Figure 5. Screenshot from fight with Taurus Demon from Dark Souls [50], used for example in Section IV-A

Dark Souls is a fantasy action-RPG game developed by From Software. The player plays as an undead human, tasked with the eventual goal of defeating a series of bosses and lighting a series of bonfires. Bonfires also act as checkpoints, and the player will revive at the most recently visited bonfire when they die. As the player progresses, they receive resources of "souls" and "humanity", which are used to increase their power. When the player dies (an extremely common occurrence in Dark Souls), they revive at the bonfire where they most recently rested, and the souls and humanity that the player has in their inventory when they die remain at the location of their death. If the player touches that location, they can re-gain their lost resources. If the player dies before re-gaining their lost resources, those resources are lost forever. For more information on the gameplay of Dark Souls, see the game mechanics guide page at IGN [51]. A screenshot of *Dark Souls* is shown in Figure 5.

Most of the gameplay in *Dark Souls* is in combat, and the game and series are known for having a deliberate pace, and a high degree of combat difficulty. Attacks in Dark Souls have long animations, and the effect of an attack or ability may not occur until seconds after the player has pressed the corresponding button. This is true of enemy attacks as well, and thus enemy attacks in Dark Souls are often highly telegraphed. To succeed, the player must learn the animations and attack patterns, to respond appropriately and at the right time. These responses include blocking, dodging, moving away from the attack, or parrying. To parry, the player has a small timing window to attempt the parry — too soon or late compared to the attack and the player will receive full damage. Blocking and dodging both use a "stamina" resource, which is also used when attacking.

An implementation of PreGLAM into *Dark Souls* that has a desire of the player winning would likely resemble most generic gameplay-based PEMs. The player successfully attacking opponents, avoiding or parrying attacks, healing themselves, or applying positive status effects are goal-oriented events, with positive valence in PreGLAM. The player being hit, receiving damage, or receiving negative status effects are goal-opposed events, with negative valence.

In addition to the events of the gameplay, resource management is also key in *Dark Souls* combat. Stamina is a resource used to take both aggressive and defensive actions, and some stamina or energy bar is common across action games. Similarly, the player has limited uses of refillable consumable objects, which restore health, cast spells, or have other gameplay effects. These variables provide context for the events of *Dark Souls* gameplay. We note that while these variables represent mechanics that are common in action games, the specific representations, names, and values vary from game to game.

For an example of mood values and prospective events in PreGLAM, consider the situation shown in Figure 5, as appraised by PreGLAM with a desire of the player defeating the Taurus demon. The Taurus Demon is often the first boss that the player encounters after completing the tutorial area, and therefore we assign a mood value of somewhat elevated arousal and tension compared to the previous area.

The Taurus demon has 5 moves: a jumping attack used at range, a vertical heavy attack with a long windup, a short-range horizontal swing with a short windup, a very short-range attack with a short windup, and a large jump used if the player stands in a particular spot. We do not know when the AI in *Dark Souls* determines which attack to use internally. However, in Figure 5, we can reasonably predict that the Taurus demon will use its jump attack. Note that this estimation requires specific game knowledge, but is trivial with such knowledge.

B. The Sims

The Sims series is a life simulation series developed by Maxis. The player controls one or more "sims", or virtual people, as the sims simulate an artificial life. Sims have wants and traits that control their behaviours, and also have a set of physical and emotional needs to attend to. The physical needs of sims are represented by a set of bars that fill or empty as the need is addressed or ignored. Some examples of physical needs are hunger, sleep, and bathroom.

Sims may get jobs to make in-game currency, which can be used to customize the furniture and architecture of the house that the player's sims live in. Additionally, in-game currency is used to pay for food and to pay regular bills that arrive for the player. Sims additionally have skills, which can be improved by interacting with certain types of furniture.

The player may choose to control one or more sims, and may have one or more goals for each sim. Various sim

Figure 6. Gameplay of the Sims 1 [52]

goals may be complementary, depending on the specifics of each goal. Starting with *The Sims 2*, released in 2004, and continuing through 2019's *The Sims 4*, sims have explicit sets of "wants" and "fears" [53]. These wants and fears are semi-randomly selected based on each sim's personality attributes and traits.

Gameplay in *The Sims*, as with *Dark Souls*, involves the management of resources, and the completion of specific tasks. However, what resources are managed, the ways that resources are drained and refilled, the set of tasks to be completed, the way that players discover tasks, and the actions to complete each task all differ greatly between these two examples.

Certain generic features used in previous models are likely to be poor predictors of player affect. In *The Sims*, the player does not have an avatar, and does not travel any distance, nor engage with any enemies, for example. While some generic features could be reinterpreted to fit within *The Sims*, such as tracking the distance each player-controlled sim travels, sims generally move around a single space, rather than progressing through a game space as in games like *Dark Souls*. Because of the very different meanings of these features between these games, we question the generic applicability of such features.

Comparatively, the gameplay of *The Sims* is not only easily modeled in PreGLAM, but the gameplay already contains a set of emotionally relevant events for player-controlled and NPC agents. Each event is associated with measurable mechanical changes that could be used to predict the event occurring. Events in the Sims are already annotated based on whether they positively or negatively affect the sim's desires, being wants or fears.

Previous generic models generally require some manual interpretation of game mechanics to derive game-specific features [18], [17], and often include mechanical assumptions that limit their flexibility. PreGLAM requires an equal amount of manual interpretation to these models, but as demonstrated, can more flexibly be incorporated into a wider variety of game genres and mechanics.

V. Use-case application: Galactic Defense

A. Game description

To evaluate PreGLAM and provide an applied use-case, we integrate the model into a video game that we develop called *Galactic Defense* (GalDef). Figure 7 shows an annotated screenshot from GalDef. GalDef is an action-RPG game originally designed for research in game music and emotion. *GalDef* is open-source [1]. In *GalDef*, the player controls a single spaceship, called the "Vatic Savate", and must win a series of 1-on-1 battles with opposing AI-controlled spaceships.

Each ship in *GalDef* has two resource bars: Health and Shield. Shields have low hit points, but constantly regenerate. If shields are completely depleted, they will not return until they have fully regenerated. Shields are temporarily lowered while a ship is taking action. Health does not regenerate, but can be partially restored with an ability. When a ship takes damage, it will take shield damage if the shields are up, and will otherwise take damage to health.

Each ship in *GalDef* has 3 abilities, and the player has an additional bonus ability. Figure 7 includes a description of each ability. Shields are removed while using an ability until a short time after the ability finishes or is canceled. The "Heavy Laser" and "Repair" abilities can be interrupted — there is a delay before they activate, and if any damage is taken during this time, the ability is canceled. When an ability finishes activating or is canceled, the acting ship will raise its shields after a short delay, if the shield has hit points remaining.

In *GalDef*, the player progresses through three stages of increasing difficulty. The player fights two opponents in the first stage, and three opponents for stages two and three. After defeating a stage, the player's ship is healed to full, and the player enters a non-combat section of the game. During this transitional segment, the player selects from a set of upgrades.

The full set of upgrades is shown in Table II. From the 10 possible upgrades, 3 are randomly selected. The player selects 2 of these selected upgrades to apply to their ship. This design is inspired by games in the *roguelike* genre, in which player's must adapt their build based on available upgrades. This upgrade design provides additional strategic depth, as the player evaluates upgrades and may change their play to accommodate the stronger abilities. These upgrades are designed to be roughly equivalent in overall power, providing variety without sacrificing emotional consistency.

Table II
Full list of possible upgrades in *Galactic Defense*

Upgrade title	Effect
Laser Supercharger	Reduces charge time of laser attack by 50%
Repair Supercharger	Reduces charge time of self-repair by 50%
Cannon Supercharger	Reduces # of shots taken until both cannons fire by 50%
Laser focusing crystal	Laser attack deals 2x damage
Cannon focusing crystal	Light attacks deal 2x damage
Repair nanite swarm	Increases self-repair from 45→90% of missing HP
Absorbitive capacitor	Increases parry window by 2x
Ion supercharge	Increases riposte disable duration
Shield capacitor	Doubles shield points
Reinforced Hull	Doubles health points

[1]GalDef is available on GitHub at https://github.com/CalePlut/Galactic_Defense

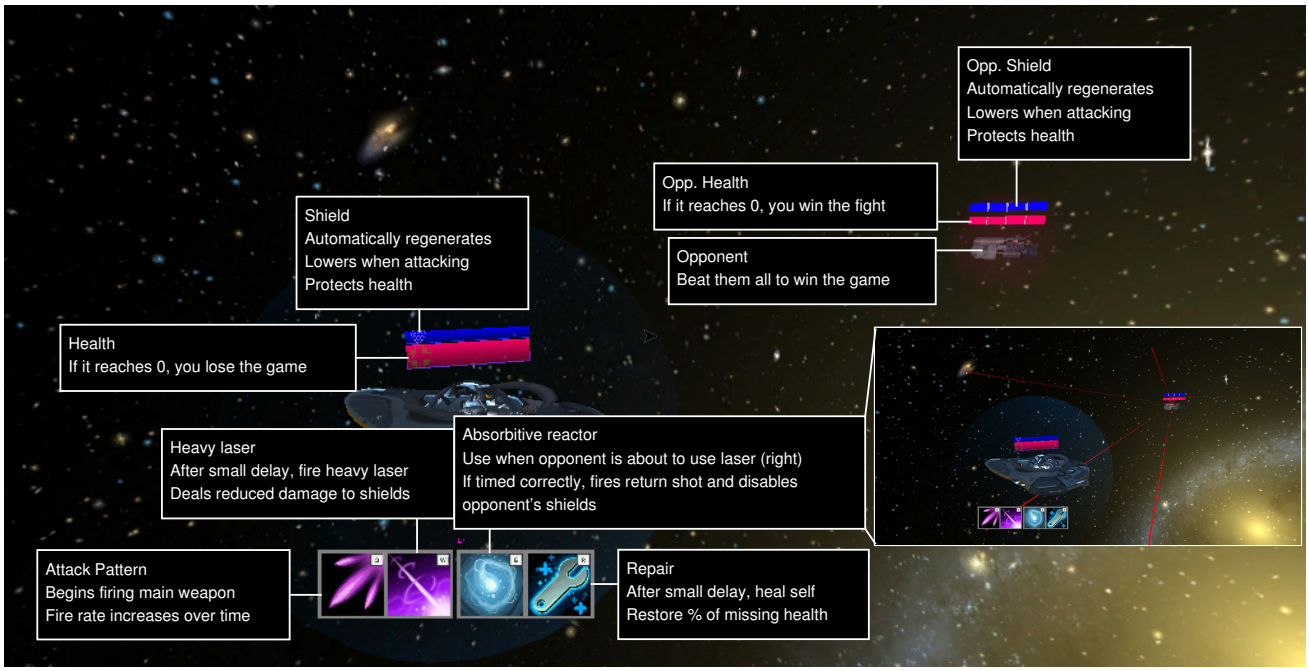

Figure 7. Visual tutorial for Galactic Defense

We use the output of PreGLAM to control the musical adaptivity of a score that is composed to adapt independently with 5 levels for valence, arousal, and tension. We assign thresholds for levels of low, low-medium, medium, high-medium, and high for PreGLAMs output, centered on a middle value of 0. The musical score is further discussed in its own paper [33].

B. PreGLAM Integration

1) Mood: The relative power between the player and their opponent in *Galactic Defense* is designed to create sections of alternating ease and challenge. The player begins the game with roughly equivalent attributes to their opponent, and is expected to win the first battle with moderate difficulty. After the first battle, the player upgrades an aspect of their ship, increasing their power. In the next battle, the player first encounters two opponents of equal strength to the opponent that they defeated in the previous stage, and then encounters an opponent of greater power, roughly equal to the player's upgraded power. This cycle then repeats for the final upgrade and stage, with the final boss tuned to be difficult for the player.

We provide mood values based on this power curve and general progress through the game. Figure 8 shows the relative power levels of the player and enemy as the player progresses through the game, and the corresponding mood levels. These mood values are derived from the design of the game — they serve to describe the general state of the game environment. We increase the mood value for valence as the player customizes and empowers the Savate with upgrades. We increase arousal's value as the player and opponent's powers increase — attacks are faster, and deal more damage. We increase the mood value for tension as the opponent's power increases, and as the player achieves more progress through the game.

This designed power curve informs the mood values for PreGLAM. Overall, valence, arousal, and tension rise as the player encounters more powerful opponents. Essentially, if PreGLAM understands that the un-upgraded player is expected to easily defeat a weak, early opponent, the mood level of valence, arousal, and tension is lower than if the upgraded player is fighting a difficult final boss.

In our implementation, mood values range from -6 to 0. As mentioned in Section V-A, we use PreGLAM to control the adaptivity of an accompanying musical score. Mood values are arbitrary, and set by a game's designer. The specific range of our mood value was designed based on the values that our adaptive score takes as input.

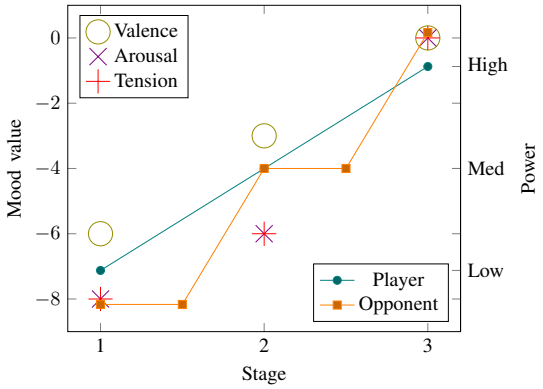

Figure 8. Mood values by level in Galactic Defense. As the player and opponent become more powerful, the mood values become increasingly active and excited

2) Emotion: We manually determine a set of EEGEs, with a desire of the player winning the game, shown in Table III.

As before, we notate the player as "P", and their opponent as "O". The selection and emotional value ranges of events are determined by experiential evaluation during playtesting of GalDef.

Prospective events are generally modeled based on privileged and/or advanced information. As an example, in our implementation, opponents have a 50% chance to finish a series of attacks with a heavy attack. This is determined behind the scenes, before the series of attacks begins, and PreGLAM begins to model a heavy attack well before any visual indication or change in player status. Similarly, the player's attack combo takes several seconds to complete — PreGLAM models the emotional perception moving towards the end of the combo, rather than responding to each event. Other prospective events are derived from modeling the designed optimal player strategy.

In our implementation, we assign a static ramp of 90 seconds for our time ramp r, for all past EEGEs. While the duration of emotions ranges from seconds to hours [38], our 90 second value is based on iterative playtesting.

Table III
EEGEs USED IN *GalDef*

Event name	Base emotion value	Context variables
P. complete atk combo	1	Missing O. shield
P. heavy atk	1	Missing O. health
O. atk combo	1	Missing P. shield
O. heavy atk	-2	Missing P. health, Parry active
P. shields down	-2	Missing P. health
O. shields down	2	Missing O. health
P. exploit O. disable	3	Missing O. health
P. death	-3	P. shield recharge time
O. death	3	O. shield recharge time
P. heal	2	Missing P. health
O. heal	-2	Missing O. health

VI. Empirical Evaluation

A. Empirical Methodology

To evaluate PreGLAMs accuracy in modeling an audiences perceived emotional response, we collect real-time user annotations from 48 participants. Prior to conducting this evaluation, we obtain approval from our research ethics board (REB). Prior to participating, participants indicate informed consent via our study website. Before annotating, each participant plays GalDef to familiarize themselves with the mechanics. Each participant annotates a single affective dimension, and watches a total of 4 videos of GalDef gameplay, recording one real-time annotation curve per video. The 4 videos differ only in the musical accompaniment to the gameplay. We evaluate how closely these participant ground-truth annotations match PreGLAM's annotations.

While we originally intended to use the *RankTrace* [54] function of *PAGAN* [29], due to cross-platform media issues, we implement a custom annotation software shown in Figure 9, based on *RankTrace* [2]. We attempt to exactly replicate the functionality of RankTrace.

Prior to annotating video, participants freely play *GalDef* to familiarize themselves with the gameplay. Training is provided in an interactive gameplay tutorial, a graphical format as shown in Figure 7, and a video format is available for them to watch. Participants are given 25 minutes to download the game and complete as many tutorials as they desire, and then to freely play *GalDef*. After at least 25 minutes, participants may begin the annotation tasks.

Figure 9. Screenshot of participant annotation interface. Note that chart resizes automatically to match unbounded participant range

While watching a video of gameplay, user can press the up or down arrows to indicate an emotional change. As with PAGAN/RankTrace, and in order to maintain consistency with PreGLAM's sample rate, button presses are collected every 250 ms, and the user is provided a visual graph of their annotation so far. Held buttons will continue to increase or decrease the annotation level.

We create 20 videos, $\approx 3-4$ minutes long, of *Galactic Defense* gameplay. We select videos that have strong changes to their emotional expression, based both on PreGLAMs output during the video and our informal evaluation. Each video has an accompanying annotation file for each dimension, generated by the output of PreGLAM.

Each participant completes one unbounded annotation curve per video, annotating along a single dimension. At each time step, the annotation software records perceived changes in emotion, indicated by participants pressing the up or down arrow keys. As with RankTrace, participant annotations are unbounded, and participants are shown a graph with the history of their annotations.

B. Results

48 participants take part in our study. Of these, 24 use he/him pronouns, and 26 use she/her. 55% of participants report playing between 0-4 hours of games per week, and the average age of participants is 23.60 years old. 39 participants are recruited from undergraduate students at the School of Interactive Arts and Technology at Simon Fraser University, 4 participants are recruited via email and message boards, and 5 participants are recruited using Amazon's Mechanical Turk platform. For all participants, the study is identical.

We analyze our results using Dynamic Time Warping (DTW), with the dtw-python library [21], using z-score scaling. DTW is a measurement of dissimilarity of the contour

[2]Our software is available on Github at https://github.com/CalePlut/GalDef_Annotation

of two time series that may be at different speeds. For each participant, we calculate the distance between PreGLAM's output annotation and ground truth participant annotations. To serve as a baseline, we also compare the ground truth participant annotation with a random walk time series of equal length, generating a new random walk series for each participant.

A dtw distance of 0 indicates that two time series are nearly identical. We normalize the dtw data to a 0-100 range. An accuracy of 100 indicates a perfect match with the data, and an accuracy of 0 indicates the maximum distance observed in the data.

The overall accuracy rating for the random walk is 56.35, with standard error mean (SEM) of 1.56. PreGLAM's overall accuracy rating is 70.80, SEM=1.04. When separated by dimension, PreGLAM has an accuracy rating of 63.95 SEM=1.89, 78.35, SEM=1.92, 72.17 SEM=1.14 for Valence, Arousal, and Tension, respectively, compared to baseline accuracy of 56.11, 53.51, and 58.77. Figure 10 shows the mean and 95% confidence interval for these values. Our transformations result in a slightly higher baseline accuracy compared to previous models such as the classifier trained on the AGAIN dataset [18], and Camilleri et al's model [17].

When evaluating the AGAIN Dataset, Melhard et al. use a Random Forest Classifier to evaluate various subsets of the datasets, compared to a set baseline accuracy of 50%. This classifier's accuracy ranges from 58.06% to 82.50%, depending on the game and the subset of features used. Camilleri et al.'s general model was evaluated in three different games, achieving average accuracy of 86.84%, 69.08%, and 75.16%, compared to baseline accuracy of 53.95%, 51.97%, and 51.59%, respectively. Table IV demonstrates an approximate comparison between these studies. Table IV provides the average model accuracy, average baseline accuracy when evaluating the model, and a derived average improvement over baseline. Note that for the AGAIN Dataset, we do no have access to the average accuracy, but instead use the midrange.

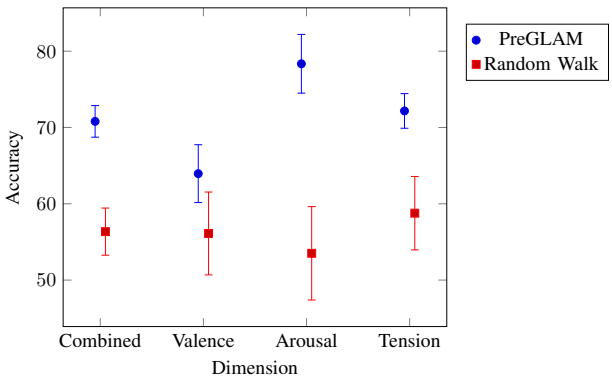

Figure 10. Mean and 95% confidence interval for normalized accuracy rating of PreGLAM and Random Walk and standard error, separated by dimension.

We test the distribution, and find normal distribution in all measures. A t-test finds significant difference between PreGLAM and random walk compared to user annotations, $p < 0.01$ for both metrics. We perform post-hoc two-way t-tests separated by dimension. Results of these t-tests are shown in Table V.

Table IV
PROVIDED ACCURACY MEASURES FROM REFERENCED MODELS

Source	Avg. accuracy	Avg baseline acc.	Difference
PreGLAM - All dimensions	70.80	56.34	14.46
PreGLAM - Arousal only	78.35	53.51	24.84
Camilleri et al.	77.03	52.50	24.53
AGAIN Dataset	70.28*	50	20.28

* Midrange, as average is not provided

Table V
RESULTS OF T-TEST COMPARING ACCURACY RATINGS BETWEEN PREGLAM AND BASELINE

Measure	Overall	Valence	Arousal	Tension
Dtw-Distance	$p < 0.01$	$p = 0.02$	$p < 0.01$	$p < 0.01$

PreGLAM significantly outperforms baseline in terms of accuracy rating. When separated by dimension, PreGLAM significantly outperforms baseline in valence, arousal, and tension. We perform an ANOVA on accuracy ratings across all dimensions, and find that the three dimensions are significantly differentiated from another. Post-hoc Tukey tests show that all pairwise comparisons of dimensions are also significantly different — modeled arousal is significantly more accurate than modeled tension, which is significantly more accurate than modeled valence.

C. Discussion

PreGLAM is generally successful at modeling an audience member's perceived emotion. Annotations from PreGLAM are significantly more accurate than baseline, and comparable in accuracy to state of the art models in estimating perceived arousal. PreGLAM has several strengths compared to SoA models. PreGLAM's framework can be more flexibly applied than compared models, and PreGLAM models more affective dimensions than most previous models. One major advantage to PreGLAM's architecture is that a ML-based implementation could be trained on game data alone, without the need for external annotations.

Most game affect models model a single affective dimension, such as Arousal [18], [17], Tension [54], [3], [55], or a 2-dimensional Valence-Arousal or Arousal-Tension model [31], [2]. PreGLAM uses a 3-dimensional Valence-Arousal-Tension model. PreGLAM significantly outperforms baseline in all three dimensions, though it demonstrates the least accuracy in Valence. Higher degrees of variance when measuring valence compared to other affective dimensions is consistent with previous research in affect. When composing music to express particular emotion, perceptions of valence demonstrate more noise than other dimensions [56], and when automatically detecting emotions, predictions of valence are generally less accurate [57].

PreGLAMs design is relatively robust to inaccuracy. PreGLAM sums together the collected emotion values from all local EEGEs to build an aggregate emotion value, and any

individual event has a limited ability to affect this aggregated value. Additionally, because PreGLAM models emotional responses within a short local time window, any single incorrect prediction or response will only affect the accuracy of the model for a short time. Finally, the OCC model that PreGLAM extends includes the possibility of predictions being incorrect. Human spectators, when estimating prospective events, may perceive an emotional change based on a predicted future event that is later disconfirmed.

Another advantage to PreGLAMs context-based time locality is that it can describe momentum where global models may struggle. For example, a player using a "control" deck in the game *Magic: The Gathering* often loses most of their health during the game, and appear to be losing, if only the current cards and player health are observable. However, once the control player draws cards with specific mechanical implications, the momentum suddenly shifts. In terms of the global game state, the control deck player is simply evening the game, but in the local view of each turn, the control deck player is dominating the game [58].

PreGLAM has one major drawback, which is that it must be integrated into a game at the design phase, and requires privileged access to real-time game telemetry. This is a disadvantage compared to post-hoc models that can be used with any game that has training data available. Because of this drawback, there are significant barriers to testing PreGLAM in various circumstances.

VII. Future work

Overall, while PreGLAM has many theoretical advantages, we implement and evaluate only a single, specific use-case. While PreGLAM is genre-agnostic, we study PreGLAM only within the action-RPG genre, in a single-player game. While EEGEs predictions may be extended with ML classification, we manually implement the predictive features. While PreGLAM's output could be used to influence many game features, we only evaluate its use to control a musical score.

We believe that there are several avenues for future work with PreGLAM. One future step is to directly compare PreGLAM's model to other techniques for modeling game emotion. While we establish a baseline efficacy for PreGLAM, it is difficult to evaluate exactly how effective PreGLAM is compared to the state of the art approaches, especially with the noise inherent in real-time affect data.

Another immediately available step in improving and evaluating PreGLAM is the use of a classifier model in EEGE prediction. Because the values of EEGEs are externally assigned, a classifier model may be trained using only gameplay data — PreGLAM does not need to be trained on ground-truth emotional annotations. Rather than ramping EEGE predictions in time, based on specific game data, a prospective EEGE's strength could scale with the classifier's odds, for example.

Because we use PreGLAM to control an adaptive score and to allow remote study participation, PreGLAM had additional requirements to run in real-time, within a Unity executable. While ML prediction would likely greatly improve PreGLAM's predictive capabilities, the implementation is outside of the scope of this paper.

More elaborate future work involves expanding the breadth of PreGLAM's capabilities, by evaluating alternative implementations with differing game mechanics, genres, desires, and timings — Currently PreGLAM models a small time window in combat, but the same framework could be applied to model longer-term mechanics as well, such as player progression or narrative elements. This avenue of future work on PreGLAM generally involves further expanding and evaluating the range of its theoretical capabilities.

VIII. Conclusion

PreGLAM presents a novel approach to modeling the perceived emotion of a passive spectator. PreGLAM uses a flexible framework that describes the actions and events of gameplay as they occur in time. We demonstrate the broad theoretical applicability of PreGLAM, and present an actual implementation in a video game. We implement PreGLAM into our game *GalDef*, manually assigning EEGEs and associated values

PreGLAM models the perceived game emotion of a passive spectator. PreGLAM is flexible, and can be applied to a wide variety of game mechanics, with a wide variety of desires, limited only by the possibilities of a game's design. PreGLAM is also lightweight, integrating into game design processes and frameworks without needing large datasets or training.

We evaluate our implementation empirically, comparing the annotations from PreGLAM with ground-truth annotations provided by gameplay spectators. PreGLAM significantly outperforms baseline, and has comparable accuracy to state of the art models. Overall, PreGLAM presents a new design-focused framework for modeling the perceived emotion of a passive gameplay spectator.

References

[1] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Transactions on Affective Computing*, vol. 2, no. 3, pp. 147–161, 2011.

[2] M. Scirea, J. Togelius, P. Eklund, and S. Risi, "Affective evolutionary music composition with MetaCompose," *Genetic Programming and Evolvable Machines*, vol. 18, no. 4, pp. 433–465, 2017.

[3] A. Prechtl, "Adaptive music generation for computer games," Ph.D. dissertation, 2016.

[4] B. Mostefai, A. Balla, and P. Trigano, "A generic and efficient emotion-driven approach toward personalized assessment and adaptation in serious games," *Cognitive Systems Research*, vol. 56, pp. 82–106, 2019.

[5] P. Gebhard, M. Kipp, M. Klesen, and T. Rist, "Adding the emotional dimension to scripting character dialogues," in *International Workshop on Intelligent Virtual Agents*. Springer, 2003, pp. 48–56.

[6] J. Clement, "Number of hours of video games streamed online 2021," May 2021. [Online]. Available: https://www.statista.com/statistics/1125469/video-game-stream-hours-watched/

[7] C. Gough, "League of legends championships viewers 2020," Mar 2021. [Online]. Available: https://www.statista.com/statistics/518126/league-of-legends-championship-viewers/

[8] L. Fernández-Aguilar, B. Navarro-Bravo, J. Ricarte, L. Ros, and J. M. Latorre, "How effective are films in inducing positive and negative emotional states? A meta-analysis," *PloS one*, vol. 14, no. 11, p. e0225040, 2019.

[9] C. Plut and P. Pasquier, "Music matters: An empirical study on the effects of adaptive music on experienced and perceived player affect," in *2019 IEEE Conference on Games (CoG)*, 2019, pp. 1–8.

[10] S. K. Holm, J. K. Kaakinen, S. Forsström, and V. Surakka, "Self-reported playing preferences resonate with emotion-related physiological reactions during playing and watching of first-person shooter videogames," *International Journal of Human-Computer Studies*, p. 102690, 2021.

[11] D. H. Kwak, Y. K. Kim, and E. R. Hirt, "Exploring the role of emotions on sport consumers' behavioral and cognitive responses to marketing stimuli," *European Sport Management Quarterly*, vol. 11, no. 3, pp. 225–250, 2011.
[12] D. W.-S. Phang, *Intelligent camera control in game replays*. Lehigh University, 2014.
[13] H. Lie, D. Lukas, J. Liebig, and R. Nayak, "A novel learning-to-rank method for automated camera movement control in e-sports spectating," in *Australasian Conference on Data Mining*. Springer, 2018, pp. 149–160.
[14] M. Scirea, "Affective music generation and its effect on player experience," Ph.D. dissertation, 2017.
[15] W. Phillips, *A Composer's Guide to Game Music*. Cambridge, MA: The MIT Press, 2014.
[16] N. Shaker, M. Shaker, and M. Abou-Zleikha, "Towards generic models of player experience," in *Eleventh Artificial Intelligence and Interactive Digital Entertainment Conference*, 2015.
[17] E. Camilleri, G. N. Yannakakis, and A. Liapis, "Towards general models of player affect," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2017, pp. 333–339.
[18] D. Melhart, A. Liapis, and G. N. Yannakakis, "The arousal video game annotation (again) dataset," *IEEE Transactions on Affective Computing*, vol. 13, no. 4, pp. 2171–2184, 2022.
[19] A. Popescu, J. Broekens, and M. Van Someren, "Gamygdala: An emotion engine for games," *IEEE Transactions on Affective Computing*, vol. 5, no. 1, pp. 32–44, 2013.
[20] A. Ortony, G. L. Clore, and A. Collins, *The cognitive structure of emotions*. Cambridge university press, 1990.
[21] T. Giorgino, "Computing and visualizing dynamic time warping alignments in r: The dtw package," *Journal of Statistical Software, Articles*, vol. 31, no. 7, pp. 1–24, 2009. [Online]. Available: https://www.jstatsoft.org/v031/i07
[22] G. N. Yannakakis, P. Spronck, D. Loiacono, and E. André, "Player modeling," 2013.
[23] S. C. Bakkes, P. H. Spronck, and G. van Lankveld, "Player behavioural modelling for video games," *Entertainment Computing*, vol. 3, no. 3, pp. 71–79, 2012.
[24] T. Mahlmann, A. Drachen, J. Togelius, A. Canossa, and G. N. Yannakakis, "Predicting player behavior in tomb raider: Underworld," in *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*. IEEE, 2010, pp. 178–185.
[25] V. Software, "Left 4 dead," Mar 2009. [Online]. Available: https://web.archive.org/web/20090327034239/http://www.l4d.com/info.html
[26] J. Pfau, J. D. Smeddinck, and R. Malaka, "Deep player behavior models: Evaluating a novel take on dynamic difficulty adjustment," in *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019, pp. 1–6.
[27] P. Bertens, A. Guitart, and Á. Periáñez, "Games and big data: A scalable multi-dimensional churn prediction model," in *2017 IEEE conference on computational intelligence and games (CIG)*. IEEE, 2017, pp. 33–36.
[28] M. Viljanen, A. Airola, A.-M. Majanoja, J. Heikkonen, and T. Pahikkala, "Measuring player retention and monetization using the mean cumulative function," *IEEE Transactions on Games*, vol. 12, no. 1, pp. 101–114, 2020.
[29] D. Melhart, A. Liapis, and G. N. Yannakakis, "Pagan: Video affect annotation made easy," in *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2019, pp. 130–136.
[30] D. Plans and D. Morelli, "Experience-driven procedural music generation for games," *Computational Intelligence and AI in Games, IEEE Transactions on*, vol. 4, no. 3, pp. 192–198, 2012.
[31] D. Williams, J. Mears, A. Kirke, E. Miranda, I. Daly, A. Malik, J. Weaver, F. Hwang, and S. Nasuto, "A perceptual and affective evaluation of an affectively driven engine for video game soundtracking," *ACM Computers in Entertainment*, vol. 14, no. 3, 2017.
[32] P. Hutchings and J. McCormack, "Adaptive music composition for games," *IEEE Transactions on Games*, 2019.
[33] C. Plut, P. Pasquier, J. Ens, and R. Bougueng, "Preglam-mmm application and evaluation of affective adaptive generative music in video games," in *The 17th International Conference on the Foundations of Digital Games (FDG) 2022*, 2022.
[34] K. Karpouzis and G. N. Yannakakis Editors, "Socio-Affective Computing 4 Emotion in Games Theory and Praxis."
[35] P. Gebhard, "ALMA: a layered model of affect," in *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005, pp. 29–36.
[36] S. Belle, C. Gittens, and T. N. Graham, "Programming with affect: How behaviour trees and a lightweight cognitive architecture enable the development of non-player characters with emotions," in *2019 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 2019, pp. 1–8.
[37] P. Ekkekakis, "Affect, mood, and emotion," *Measurement in sport and exercise psychology*, vol. 321, 2012.
[38] P. Verduyn, P. Delaveau, J.-Y. Rotgé, P. Fossati, and I. Van Mechelen, "Determinants of emotion duration and underlying psychological and neural mechanisms," *Emotion Review*, vol. 7, no. 4, pp. 330–335, 2015.
[39] ——, "Determinants of emotion duration and underlying psychological and neural mechanisms," *Emotion Review*, vol. 7, no. 4, pp. 330–335, 2015.
[40] N. H. Frijda, *The laws of emotion / Nico H. Frijda.* Mahwah, N.J.: Lawrence Erlbaum Associates, 2007.
[41] Z. Callejas and R. Lopez-Cozar, "Influence of contextual information in emotion annotation for spoken dialogue systems," *Speech Communication*, vol. 50, no. 5, pp. 416–433, 2008.
[42] P. Barros, N. Churamani, E. Lakomkin, H. Siqueira, A. Sutherland, and S. Wermter, "The OMG-Emotion behavior dataset," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–7.
[43] U. Schimmack and A. Grob, "Dimensional models of core affect: a quantitative comparison by means of structural equation modeling," *European Journal of Personality*, vol. 14, no. 4, pp. 325–345, 2000.
[44] N. Malandrakis, A. Potamianos, G. Evangelopoulos, and A. Zlatintsi, "A supervised approach to movie emotion tracking," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 2376–2379.
[45] R. L. Hazlett, "Measuring emotional valence during interactive experiences: boys at video game play," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, 2006, pp. 1023–1026.
[46] J. A. Bopp, E. D. Mekler, and K. Opwis, *Negative Emotion, Positive Experience? Emotionally Moving Moments in Digital Games*. New York, NY, USA: Association for Computing Machinery, 2016, p. 2996–3006. [Online]. Available: https://doi.org/10.1145/2858036.2858227
[47] K. Niven, *Affect*. New York, NY: Springer New York, 2013, pp. 49–52. [Online]. Available: https://doi.org/10.1007/978-1-4419-1005-9_1088
[48] U. Nixius, "Wildstar - telegraph," 2014. [Online]. Available: https://wildstar.fandom.com/wiki/Telegraph
[49] N. Coding, "Grapher - graph, replay, log: Utilities tools: Unity asset store," Dec 2017. [Online]. Available: https://assetstore.unity.com/packages/tools/utilities/grapher-graph-replay-log-84823
[50] N. Life, "Dark souls: Remastered screenshots," Jul 2020. [Online]. Available: https://www.nintendolife.com/games/nintendo-switch/dark_souls_remastered/screenshots
[51] u. FrostedSloth, K. (username), and S. Saris, "Game mechanics - dark souls wiki guide," Feb 2013. [Online]. Available: https://www.ign.com/wikis/dark-souls/Game_Mechanics
[52] U. thedarkcave, Feb 2020. [Online]. Available: https://thefinkedfilms.com/2020/02/04/the-sims-20th-anniversary/
[53] R. Jones, "The sims 4 wants and fears explained," Jul 2022. [Online]. Available: https://www.rockpapershotgun.com/the-sims-4-wants-and-fears
[54] P. Lopes, G. N. Yannakakis, and A. Liapis, "Ranktrace: Relative and unbounded affect annotation," in *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE, 2017, pp. 158–163.
[55] C. Plut, "Music Matters Musical Examples," March 2019. [Online]. Available: https://soundcloud.com/khavall/sets/music-matters-musical-examples/s-5fBfr
[56] S. Vieillard, I. Peretz, N. Gosselin, S. Khalfa, L. Gagnon, and B. Bouchard, "Happy, sad, scary and peaceful musical excerpts for research on emotions," *Cognition and Emotion*, vol. 22, no. 4, pp. 720–752, 2008. [Online]. Available: https://www-tandfonline-com.proxy.lib.sfu.ca/doi/pdf/10.1080/02699930701503567?needAccess=true
[57] A. Haag, S. Goronzy, P. Schaich, and J. Williams, "Emotion recognition using bio-sensors: First steps towards an automatic system," in *Affective Dialogue Systems*, E. André, L. Dybkjær, W. Minker, and P. Heisterkamp, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 36–48.
[58] P. V. D. da Rosa, "The only point of life that matters is your last," Oct 2017. [Online]. Available: https://strategy.channelfireball.com/all-strategy/mtg/channelmagic-articles/the-only-point-of-life-that-matters-is-your-last/