Modeling Playability of Arbitrary 2D Games

Aditya Singhal (adis@nyu.edu)
Department of Psychology, Department of Data Science

Buz Galbraith (wbg231@nyu.edu)
Department of Economics, Department of Data Science

Abstract

The automatic generation and validation of games is a useful challenge with implications not just in game-design, but also in cognitive science. We create a new 2D environment which is interactable by humans and programmatic agents. In comparing peoples' perception of the difficulty of given 2D games, we find that our model better aligns with their difficulty scores before playing the game vs after playing it. We discuss areas of future work that become possible through this environment and the Domain Specific Language for Game Creation.

Keywords: procedural-content-generation; game-creation; goal-generation; DSL

Introduction

People show an interest in gameplay from an early age. Some research conducted to study this behavior in children (Chu & Schulz, 2020) suggests that this playtime of creating and solving arbitrary problems is instrumental in the development of human creativity and exploration. We seek to build on previous research conducted by Davidson, Lake and Gureckis in the psychology and computational cognitive science of "playful goal creation" (Davidson, Gureckis, & Lake, 2022) by making a new 2D environment to study arbitrary games.

Motivation

Automated Game Creation Procedural content generation (PCG), a term coined in (Shaker et al., 2018), is the automatic generation of game content and is distinct from many other game-based applications of ML which focus on the training of artificial agents to play games. The most straightforward among the many applications of PCG, is that of level-generation for games. This generation could be "offline" i.e., prior to game release, where PCG is used to automate lower priority tasks, thus allowing smaller teams of developers to focus human efforts and produce games more cost-effectively. Conversely, 'online' PCG could be used to develop new content as players interact with a game in real-time. For instance, RL agents could interact with players over time and learn to produce more personalized content. With enough advances in PCG, game content could be produced quickly enough to make never-ending games that keep evolving, or it could also lead to the creation of completely novel games altogether. The current work is another step in the pursuit to understand how people interact with and enjoy games, so that more such entertaining content and insight into the human psyche can be produced through automated means.

Games as Goal Generation A key influence for this paper was "Creativity Compositionality and Common Sense in Human Goal Generation" by Davidson, Gureckis, and Lake, in which experimenters asked participants to design novel games in a standard 3D environment and describe them in natural language. Experimenters then implemented these games into a structured language called the Domain Specific Language (DSL) for Game Creation and studied the relationship between these games. Formalizing games with a DSL allowed for comparison of the underlying structure of games. It is possible that this DSL would allow for the automatic setup and scoring of any describable game. This research has opened the path for numerous new vectors of research, including melding of DSLs with PCG to allow Reinforcement Learning agents to better generalize across games i.e., allowing for outer transfer, and to procederally generate game content the way humans do. A final implication of this research that underpins much of the current work, is using DSL to develop more robust notions of game difficulty, enjoyability, or playability.

Related Work

Procedural Content Generation

Procedural content generation is the algorithmic creation of game content with limited or indirect user input. This paradigm has been primarily applied to the task of increasing the amount of content in games with pre-existing rule sets. Commercially successful and varied applications of this approach include Dwarf Fortress (*Bay 12 Games: Dwarf Fortress*, n.d.), Diablo (*Blizzard Entertainment*, 2019), Spore (*Spore*™, n.d.), Minecraft (Mojang, 2020), Spelunky (Yu, 2013), and Hades (*Supergiant Games*, n.d.). Another application is the generation of rule sets or games themselves given only game engines and environments, which is difficult because the task of finding "playable" or "fun" rule sets is effectively a very high dimensional search problem with a sparse reward space. Many rule sets could potentially exist, but few are playable and even fewer are fun. Further, this problem requires both a formalized definition of playability and a metric measuring levels of 'fun' to properly evaluate a potential game or rule set. Despite these challenges, interesting progress has been made specifically in the procedural generation of novel board games with projects like Metagame

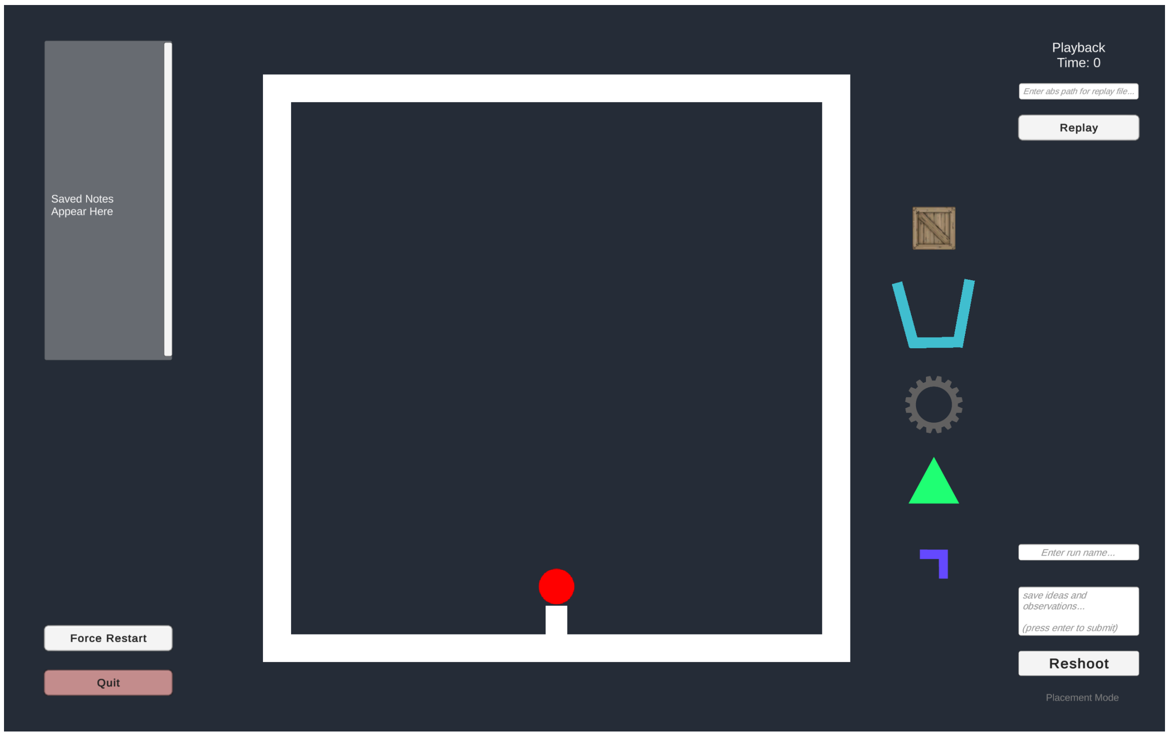

Figure 1: Interface of the Unity Environment

(*the*, n.d.) and Angelina (Mike, 2018). Further research is being conducted to create standard languages to model games including the Video Game Description Language (VGDL) (Summerville, Snodgrass, Mateas, & Ontañón, 2016) and Domain Specific Language (DSL) for Game Creation as mentioned before. These attempts have the potential to vastly improve comparability of more distinct types of games. Despite the notable challenges in procedural generation, the exercise of formalizing games may yield new insights into the psychology underpinning existing game design and game interaction.

Playability

A challenge in evaluating procedurally generated rule sets and games, is codifying a workable notion of the "playability" of a game. In the space of PCG, playability is defined quantitatively as if the game can be completed. That is a program can check if an agent playing this game would be able to reach the terminal state. There, however, also exist other useful notions of playability that are qualitative. Among the qualitative definitions, playability can be thought of as the game-equivalent of usability in interactive systems, which is the measure of a product's use in varying degrees of effectiveness, efficiency and satisfaction (ISO Usability: Definition and Concepts). While this definition of playability is useful in its ability to generalize to interactive systems outside of video games, there exists another more domain-specific definition of playability by González-Sánchez, Padilla-Zea, and Vela (2009) which calls it, "a set of properties that describe the Player Experience using a specific game system, whose main objective is to provide enjoyment and entertainment, by being credible and satisfying, when the player plays alone or in company [. . .]" While there is currently no agreed-upon formalization of playability, keeping these three definitions in mind, we can understand the concept with varying levels of generalization, both quantitatively and qualitatively. For the purposes of the current work, we define playability as simply the likelihood of receiving any reward given a predetermined set of different actions.

The Environment

The Figure 1 shows the interface of our Unity Environment [1] (Technologies, 2019). The main components are the pedestal, the ball, the 5 objects (Crate, Bucket, Gear, Triangle, and Corner), a reshoot button, a note-taking system, and a replay system. The environment complies with Unity's 2D physics. The ball and pedestal must be placed somewhere on the x-axis before being shot, and cannot be interacted with again until the agent presses the "reshoot" button. Shooting follows catapult-like mechanics, and a power bar appears for the agents when they drag their mouse down from the ball, giving them a sense of the power and direction of their shot. The objects are freely movable and have colliders to interact with the ball but not with each other. This makes overlapping placements of objects possible. The choice of the 5 objects was based on simple shapes that may play a role in many different

[1]https://github.com/adityaarunsinghal/temporal-goals-in-games

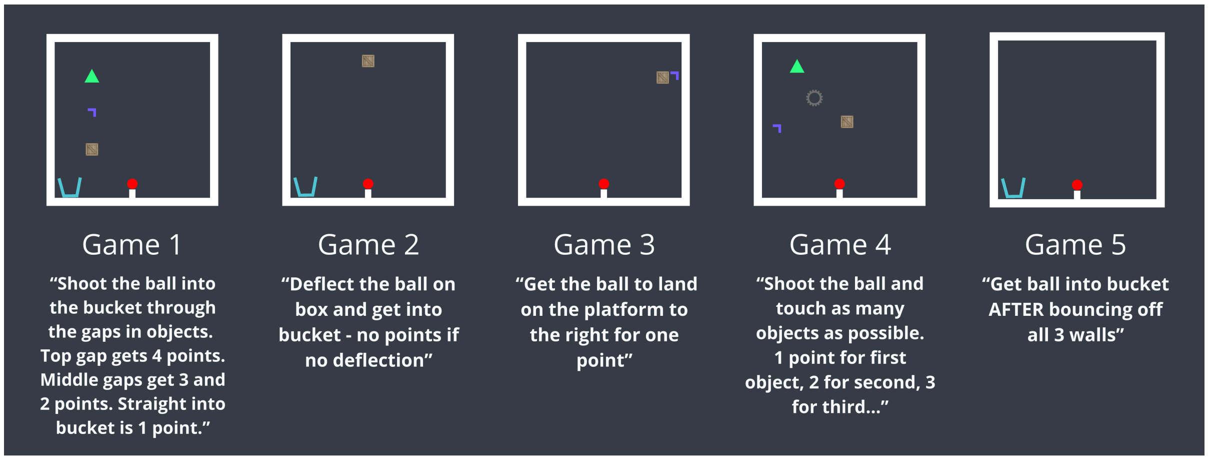
Game 1
"Shoot the ball into the bucket through the gaps in objects. Top gap gets 4 points. Middle gaps get 3 and 2 points. Straight into bucket is 1 point."
Game 2
"Deflect the ball on box and get into bucket - no points if no deflection"
Game 3
"Get the ball to land on the platform to the right for one point"
Game 4
"Shoot the ball and touch as many objects as possible. 1 point for first object, 2 for second, 3 for third..."
Game 5
"Get ball into bucket AFTER bouncing off all 3 walls"

Figure 2: Example Game Descriptions and Setups

games but was not based on previous research. The agent is not given a choice to duplicate, delete or rotate objects so as to maintain the low dimensionality of the action space.

The environment can be interacted with using a mouse-click system (for human agents) and also through action vectors (for programmatic agents). Actions include mouse positions for placement and shooting of balls, x-y positions for objects, choice of object, and a boolean value for resetting the ball onto the pedestal. Each run of the game produces a comprehensive state trace that tracks the ball position, ball velocity, all object positions, collisions, and resets for every time step. These same values are also available as an observation vector for the RL agent after each step.

Gym The ability to input actions and output observations from the environment as vectors, enables the formulation of the unity environment as an OpenAI Gym (OpenAI, 2019), which in turn unlocks various RL frameworks like Stable Baselines (*https://github.com/DLR-RM/stable-baselines3*, 2021)). A "step" in this environment is comprised of three required actions: a reset, a placement on the x-axis, and a shoot of the ball. The step keeps retrieving observations from the environment until the ball stops moving (velocity under a specified threshold). These stacked observations of arbitrary lengths from each timestep in the ball's trajectory enable us to formulate temporal goals, similar to the ones our human participants described. The reward function need not observe just the end state of the ball ("ending in a bucket") but can also capture its interactions with other objects/walls on the way ("bouncing off the crate before ending in the bucket"). This makes it possible to emulate more real-world goals in games.

Pilot Games

In an initial pilot of the environment, 5 participants were asked to generate 3 games each within 15 minutes of interacting with the environment. The criteria for the games were simply that each game must be playable by a single player, scorable and "fun to play." This study resulted in 5 main motifs for games:

1) End-state dependent (Bucket) – The ball is meant to do something before ending in the bucket

2) End-state dependent (non-Bucket) – The ball is meant to do something before ending in non-bucket end state

3) Non-end state dependent – The ball is meant to do something during its trajectory, but the end-state is irrelevant

4) Ball bouncing challenges – The objective is to bounce the ball off something while it is in motion

5) Object Interaction – The objects are meant to be interacted with while the ball is still in motion

Some example games that were described by participants are shown in Figure 2. These are the 5 games that we used for our playability experiments described in the following sections. Game 1, 2 and 5 fall under the 'Bucket End-State' category, although games 2 and 5 also have a motif for 'Ball Bouncing Challenge.' Game 3 is an example of games which are exclusively in the 'Non-Bucket End-State,' while Game 4 is exclusively a Ball Bouncing game where the end-state is irrelevant. None of the games under the 'Object Interaction' type were included for the purposes of this study, because that would greatly increase the action space, thus making a grid search over different shots less straightforward.

DSL Finally, the 15 games described in the pilot were also converted to the Domain Specific Language for Game Creation introduced in the paper by Davidson et al. (2022). This formalization of game description could be useful to automate setting up and scoring of games, and even PCG through RL. Although the current pilot study does not implement DSL, all games described by participants were successfully codified as DSL snippets. This allows for more detailed study of the structure of games and validates the DSL as a promising formal language for all sorts of games. Some work is being conducted by the original authors to make a bridge between NLP and DSL such that human descriptions of games can be accurately converted to the formal description, mak-

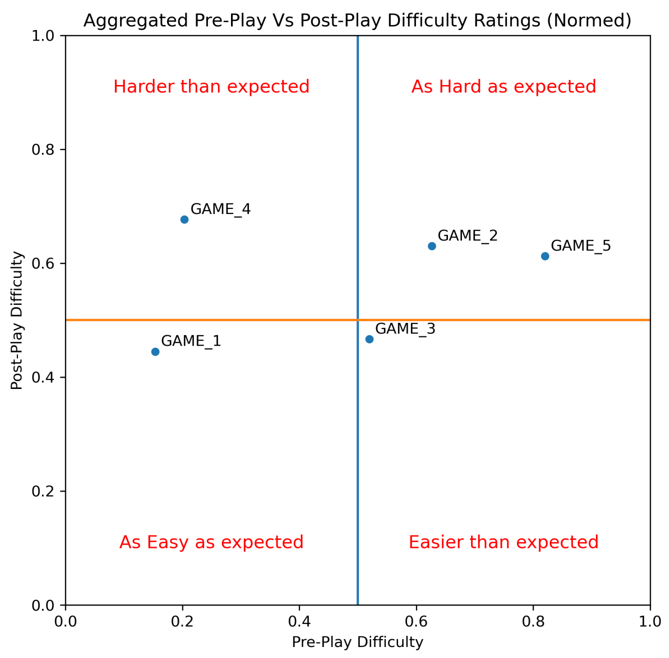

Figure 3: Pre-Play Vs Post-Play Ratings

ing the task of game-validation much more efficient. Future work with a larger corpus of participant data will focus more on codifying games as DSL.

Methods

Human Data

The 5 games for the experiment were shown to 12 different participants who were not involved in the earlier game-creation process. The descriptions by the original game creators, as shown in the Figure 2 were shown verbatim to participants along with an image of the initial setup. After a brief explanation of the environment's mechanics, they were asked to rate the intuitive difficulty of each game on a 10-point scale without playing it first. Then they were given "playtime" with the environment to setup, play and calculate rewards for each game themselves and score the difficulty on a 10-point scale again. We call these the Pre-play and Post-play difficulty scores respectively. These scores were scaled within each participant using Min-Max Scaling and then averaged to get the final, normalized scores for difficulty.

Automated Playability Scores

The reward functions for the 5 games described above were coded into the environment manually. An action space was defined which eliminated some spaces for ball placement according to the game logic – for example the bottom left corner in games where the bucket is kept on the floor was not included in the grid-search. Then a search over all mouse_x and mouse_y positions was conducted such that different shots, with different powers from different ball placements would be made and observed until the ball stops moving. The step size for this grid search was set at 0.05 so that when searching over possible values in the range (-1, 1) for each variable, there would be up to 64000 different shots made for each playability trial. The rewards received in Game 1 and Game 4 varied across shots (non-boolean), and so any $reward > 0$ could be counted as a playable shot. While this binarization worked for Game 1, we saw that Game 4 had more than 56% of shots being called playable because they ended up touching at least one object. For this reason, we chose to call all shots with $num\_obj\_touches >= 2$ to be "winning shots," which also aligns more closely with what we observed from the human interactions with Game 4. This brought playability of Game 4 down from 0.56 to 0.19.

Results/Analysis

Pre- and Post-Play Difficulties

In the Figure 3, we plot the aggregated difficulty scores from participants for the 5 games on a unit scale and divide the space into quadrants representing the expectations of difficulty before play vs the updates scores post-play. This gives us insight into which games people were good at predicting difficulties for, and which ones they were surprised by.

We find that all bouncing games (Game 2, 4, and 5) were perceived as difficult post-play, as compared to Games 1 and 3 which were deemed easier after playtime. Of these, only game 4, the one without any required end-state, was perceived as 'unexpectedly' difficult, as inferred from its position on the graph. Games where bouncing off other things in the environment wasn't explicitly part of the task (Game 1 and 3) were found to be either expectedly or unexpectedly easy. This may be the result of these games being more direct or *aim-focused*

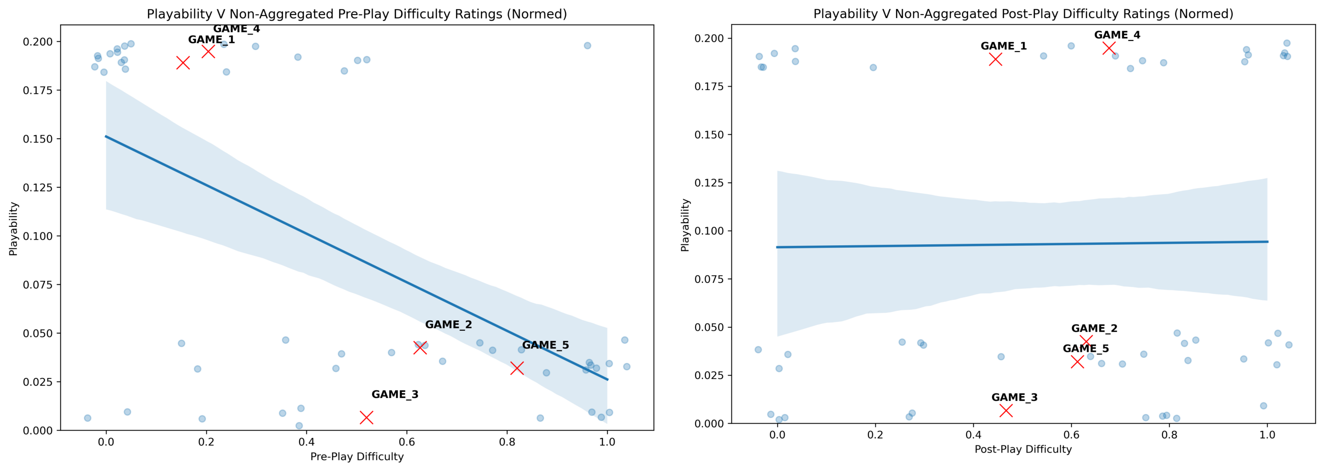

Figure 4: Playability Interactions

as opposed to redirection or *collision-focused*. Another interpretation from the figure is that both games with multiple scoring levels (Game 1 and 4) were perceived as easy to play before actual playtime. These are interesting patterns, and future work will test them with a higher-powered study.

Correlation with Playability

As seen in Figure 4, we find that pre-play difficulty scores are significantly negatively correlated with playability. This means that the easier the participants thought a game would be after only seeing the setup and reading the game description, the more "playable" the games actually ended up being. Interestingly, post-play difficulty scores had a close to 0 correlation with our implementation of playability. This hints towards the idea that people's score after experience with the games cannot be captured by our current definitions of playability where a *grid-search* over different actions is conducted. Perhaps these ratings would be better aligned with a completely different concept like learnability, which would need to be modelled differently.

Limitations/Discussion

Definition of Playability

The implementation of playability-checking as a grid-search may suffer from biases against certain games where scoring points is easy, but "solving" the game is harder. An example of this is Game 4, where the majority of shots actually get a $score > 0$, but people shift their goals to be more challenging without being explicitly asked to do so. Most participants made comments about wanting to touch three or more objects in the same shot even though the game description did not mention any threshold for what scores counted as "wins." Every participant was able to at least one object in the very first try, which made a score of 1 too trivial. In future work, we will score the participants via their state traces during interaction to see how their difficulty scores correlate with their true rewards from the games. Another important concept may be the "finickiness" of a game-setup, which may calculate how likely a shot is to succeed if some noise is added to it. We may capture this more and more with a smaller step-size in our grid search, but a more intelligent, gradient-informed grid-search may be a better implementation. We also do not incorporate knowledge of the actual rewards distribution from either the grid-search or the human play-time, and default simply to a Boolean quantization of playability. A new playability function that accounts for this variety in rewards may align closer to what people naturally intuit.

Amount of Data

Since this is a pilot study, we acknowledge the small sample size, both in number of games and number of participants. In future work, we will host the game on a web client and recruit participants more formally. The current results may be skewed and low on power due to outliers.

Temporal Window of Observations

Due to differences in implementations of the Unity environment state-traces and the OpenAI gym observation vector, there is mismatch in the agent's temporal perceptive field. The gym implementation currently outputs observations every n milliseconds whereas the unity environment can use its internal update cycle to output observations after every update of the physics system. Because of this, certain quick events (like bouncing off a wall) may not be captured in the final, stacked observation vector, thus making the end-rewards inaccurate in some special cases. For the purposes of the current work, we believe that this disadvantage is equal across games, making the relative results interpretable, but future work will eliminate this bias through some restructuring of the environment.

Not using State Traces

The previously mentioned limitation could be circumvented by analyzing state traces, but they can get quite large for long grid searches. An effort to make them sparser will be made in the next iteration. Similarly, the available state traces from human participants is in no way used even though they contain valuable information about the heuristics people intuitively use to solve these games.

Conclusion

We find that people's perception of difficulty of a 2D physics game setup before attempting it is highly correlated with the actual percentage of shots that score points in a comprehensive grid search. However, there is a concept drift in what "difficulty" means to participants after gameplay, and it cannot be captured by the same metric of playability. We see that games with multiple levels of scoring make them seem less daunting pre-play, perhaps because at least one of the scores seem "achievable" to begin with. Games where redirection of the ball was key were scored high on difficulty post-play as compared to games involving direct-shots/aiming. Finally, we add our environment to the pool of 2D physics RL gyms as a means to study human gameplay in greater detail. We are excited to keep updating the environment and see what the community is able to discover through it.

Future Work

In future work, we hope to model difficulty of games using different definitions of playability. There is much to be studied about human goal generation, including learnability of games, object affordances, intuitions of what is "fun" and "creative" etc. We also hope to study what it means for something to be a 'game' by coming up with different "null" games which are either machine-generated or collected from participants. This could lead to important insight into what goals people deem worth pursuing and which ones are thought to be too mundane.

Acknowledgements

The authors would like to vehemently thank Guy Davidson, Brenden Lake and Todd Gureckis for their support and guidance throughout this project. Previous work by Julian Togelius was also a great inspiration. We are grateful for Prof. Radulescu and Johnny Ma for taking the time to evaluate the project. Finally, we thank the many participants who gave us their time and interest for this study.

References

(n.d.).

Bay 12 games: Dwarf fortress. (n.d.). Retrieved from `http://www.bay12games.com/dwarves/`

Blizzard entertainment. (2019). Retrieved from `https://www.blizzard.com/en-us/`

Chu, J., & Schulz, L. E. (2020, 10). Play, curiosity, and cognition. *Annual Review of Developmental Psychology*, *2*. doi: 10.1146/annurev-devpsych-070120-014806

Davidson, G., Gureckis, T. M., & Lake, B. M. (2022, 03). Creativity, compositionality, and common sense in human goal generation.
doi: 10.31234/osf.io/byzs5

González-Sánchez, J., Padilla-Zea, N., & Vela, F. L. (2009, 08). *Playability: How to identify the player experience in a video game.* doi: 10.1007/978-3-642-03655-2_39

(2021, May). Retrieved from `https://github.com/DLR-RM/stable-baselines3`

Mike. (2018, Sep). *Games by angelina.* Retrieved from `http://www.gamesbyangelina.org/`

Mojang. (2020, 04). *Minecraft official site.* Retrieved from `https://www.minecraft.net/en-us`

OpenAI. (2019). *Gym: A toolkit for developing and comparing reinforcement learning algorithms.* Retrieved from `https://gym.openai.com/`

Shaker, N., Togelius, J., Nelson, M. J., & Ag, S. I. P. (2018). *Procedural content generation in games.* Cham Springer International Publishing Springer. Retrieved from `https://www.academia.edu/1578545/Procedural_content_generation_in_games`

Spore™. (n.d.). Retrieved from `https://www.spore.com/`

Summerville, A. J., Snodgrass, S., Mateas, M., & Ontañón, S. (2016, 07). The vglc: The video game level corpus. *arXiv:1606.07487 [cs].* Retrieved 2021-11-24, from `https://arxiv.org/abs/1606.07487`

Supergiant games. (n.d.). Retrieved from `https://www.supergiantgames.com/games/hades/`

Technologies, U. (2019). *Unity - unity.* Retrieved from `https://unity.com/`

Yu, D. (2013). *Spelunky world.* Retrieved from `https://spelunkyworld.com/`