



universität
wien

BACHELORARBEIT / BACHELOR'S THESIS

Titel der Bachelorarbeit / Title of the Bachelor's Thesis

„A Stealth Serious Game About Hiring Bias“

verfasst von / submitted by

Paul Friedrich Pesak

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Bachelor of Science (BSc)

Wien, 2024 / Vienna, 2024

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

UA 033 521

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Informatik - Medieninformatik

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Ing. Dr. Helmut Hlavacs

Abstract

Racism and sexism can be found in many areas of life, including the recruitment process. Although there is growing awareness of these issues, many biased hiring decisions are made unconsciously. This bachelor thesis explores the issue of unconscious bias in recruitment through the use of serious games. Serious games are increasingly recognised as a fun way to educate users, but they also provide a way to observe and later evaluate the decisions of a player. This study provides an overview of serious games that have addressed issues of bias and presents a newly designed game that aims to raise awareness of unconscious bias in hiring decisions. This game was used to assess the hiring decisions of participants given only the “stats” and image of each applicant. The results indicate the presence of significant hiring bias against women among several actors, particularly in the direct choice between male and female candidates and in the assignment of employees to traditionally male-dominated jobs such as construction workers. These findings confirm the continuing influence of gender bias in recruitment scenarios and suggest that unconscious bias can have a significant impact on recruitment outcomes.

Kurzfassung

Rassismus und Sexismus sind Themen, die in vielen Lebensbereichen präsent sind, so auch bei der Einstellung neuer Mitarbeiter. Obwohl das Bewusstsein für diese Probleme zunimmt, werden die meisten vorurteilsbehafteten Entscheidungen bei der Einstellung von Mitarbeitern oft unbewusst getroffen. Diese Bachelorarbeit untersucht das Problem der unbewussten Voreingenommenheit bei der Rekrutierung durch den Einsatz von Serious Games. Serious Games werden zunehmend als eine Möglichkeit anerkannt, Benutzer spielerisch zu bilden, bieten aber auch die Möglichkeit, die Entscheidungen der Spieler zu beobachten und zu bewerten. Diese Arbeit gibt einen Überblick über Serious Games, die sich mit Rassismus und Sexismus beschäftigen haben und stellt ein neu entwickeltes Videospiel vor, dessen Ziel es ist, das Bewusstsein für unbewusste Vorurteile bei Einstellungsentscheidungen zu erhöhen. Das Spiel wurde verwendet, um die Einstellungsentscheidungen von Testpersonen zu bewerten, die nur "Stats" und ein Foto von jedem Bewerber als Auswahlkriterien hatten. Die Ergebnisse zeigen, dass mehrere Testpersonen weibliche Bewerber bei der Einstellung diskriminieren, insbesondere wenn sie zwischen einem männlichen und einem weiblichen Bewerber wählen müssen und beim Zuweisen zu eher traditionell männlich dominierten Berufen wie Baustellenarbeiter. Diese Ergebnisse bestätigen den anhaltenden Einfluss von geschlechtsspezifischen Vorurteilen auf die Rekrutierung und deuten darauf hin, dass unbewusste Entscheidungen einen erheblichen Einfluss auf das Ergebnis von Bewerbungen haben können.

Contents

Abstract	i
Kurzfassung	iii
List of Tables	vii
List of Figures	ix
Listings	xi
1 Motivation	1
2 Related Work	3
2.1 Hiring Bias in Real Life	3
2.2 Serious Games about Gender and Race Bias	4
3 Design	7
3.1 Important Aspects for Stealth Serious Games	7
3.2 Basic Concept	7
3.3 Game Scenario and Job Roles	8
3.4 Employee Attributes	9
3.5 Gameplay	11
3.6 Hiring Employees and Stealth Assessment	14
3.7 Winning and Losing	19
4 Implementation	21
4.1 Generation of Construction Sites and People	21
4.2 Touch Controls	26
4.3 Hiring and Working	28
4.4 Finishing a Construction Site	31
4.5 Statistics Assessment	33
5 Evaluation and Discussion	35
5.1 Setup	35
5.2 Results	36
6 Conclusions and Future Work	41

Contents

Bibliography

43

List of Tables

5.1	Participants hiring probability significantly less than 50% of a group per category.	37
5.2	How many participants distributed significantly less than 33.33% of a demographic group to a job.	38
5.3	How many participants distributed significantly more than 33.33% of a demographic group to a job.	38

List of Figures

3.1	Symbols in the game for different job roles.	9
3.2	Bar showing the experience distribution of a person.	9
3.3	Bar showing the stress level of a worker.	10
3.4	Bar showing the talkative value of a person.	10
3.5	The map of the game with construction sites in various states.	11
3.6	Loading screen presented to the user when setting up a construction site. .	13
3.7	Construction view showing options to interact with a construction site. . .	13
3.8	Making an employee work.	14
3.9	Team overview to select between employees.	14
3.10	Hiring applicants at the start of the game.	15
3.11	Applicant pair only being different in gender.	15
3.12	Applicant pair only being different in race.	16
3.13	Statistics shown for overall hired workers.	17
3.14	Statistics that shows what was chosen when there was a difference in either gender or race.	18
3.15	Statistics for construction worker job role.	18
3.16	Screen shown after winning the game.	19
3.17	Screen shown after losing the game.	19
5.1	Distribution of arrests.	36

Listings

4.1	Method to create a construction site with a given difficulty.	22
4.2	Method to create a random person with a given sprite.	25
4.3	Method to check if a touch is on a collider.	26
4.4	Method to check if a touch is made on the UI	27
4.5	Method to set up a person pair	29
4.6	Method to send the selected applicants to the statistic class	30
4.7	Method to check if the hiring probability is significantly smaller than 50%	34

1 Motivation

The issues of racism and sexism are known problems in society. Such issues are pervasive in the context of hiring practices. A number of studies have documented discriminatory practices in hiring, with female and non-white individuals facing particular challenges in certain job sectors. Despite an increase in awareness of these issues, discriminatory decisions continue to be made by individuals because of unconsciously biased decisions. Consequently, instances of hiring bias remain prevalent.

Serious games are utilised as an educational instrument to convey information in an engaging and enjoyable manner. A number of serious games have been released that address racism and sexism. The majority of these games seek to raise awareness about these issues and their impact on affected individuals, as well as highlight ways in which these problems can be identified. However, the majority of these games do not attempt to make the player aware of unconscious decision-making processes that lead to biased decisions. One notable exception is the game “Purpose” [27], which employs a complete fictional scenario of a zombie apocalypse to detect the unconscious biases of players based on stereotypes.

It is regrettable that none of the aforementioned games addresses the impact of unconsciously biased decisions on hiring practices. The software project presented in this thesis offers a serious game in which the player assumes the role of an individual responsible for hiring new employees. In contrast to previous serious games, this game is designed to detect whether a user exhibits bias when hiring personnel. This serves to illustrate to individuals who believe they are impartial judges that they unconsciously make biased decisions.

The game is delivered via an Android application developed using the Unity game engine. The player has to choose between several candidates during the game and is shown only a picture and a “stat” for each candidate. These candidates are either different in gender or race and the player is evaluated based on the indications of bias observed in the playthrough. The game is a stealth serious game, meaning that the user is unaware that they are being evaluated at the end of the game. The aim is for the player to get into a flow while playing, which can lead to unconscious decisions such as hiring a person based on gender or race, ignoring the score that actually shows how good a candidate is at a job. In order to be able to say that a user is showing a bias, the hiring decisions are evaluated with the hypothesis test for proportions. This helps us to take into account random conditions that could also have led to an uneven distribution. Based on discrimination in real life, the game will evaluate two things about the user:

1 Motivation

- Does the user show a bias against women when hiring?
- Does the user show a bias against non-white people when hiring?

In addition to examining the presence of bias in the hiring process as a whole, the game also looks at whether there is bias when players are presented with a choice between two individuals who differ in gender or race, and whether there is bias associated with specific job roles.

A total of 31 participants were invited to take part in an evaluation of the game. The results of this evaluation were then subjected to analysis in this thesis, with a view to identifying any parallels with real-life results. In addition, the participants were invited to complete a survey, in order to indicate whether the game had made them more aware of the need to avoid judging people unconsciously in a biased manner. This allowed us to ascertain whether the players had been trained to avoid unconscious bias.

2 Related Work

The topics of sexism and racism have been extensively researched. Studies have examined the existence and causes of bias in recruitment based on gender and race. Additionally, statistical analysis of workplace demographics reveals significant disparities in the representation of different gender and racial groups across various occupational sectors. In light of these concerns, a considerable number of serious games have been developed to educate the public about the prevalence and impact of discrimination in different contexts.

2.1 Hiring Bias in Real Life

In the paper “Comparative perspectives on racial discrimination in hiring: The rise of field experiments” [23], the authors discuss field studies that have identified a hiring bias based on ethnicity and race. The authors concentrated their analysis on 140 field studies that demonstrated discriminatory practices against racial minorities in 30 countries. They investigated the underlying causes of these outcomes and identified two primary explanations: taste discrimination, which occurs when individuals are denied employment based on their employers’ prejudices, and statistical discrimination, which occurs when potential employees are overlooked due to the employer’s reliance on racial averages in evaluating productivity. To illustrate the impact of statistical discrimination in a real-world context, we will examine actual job statistics to determine which roles are likely to elicit a bias in players in the hiring bias game.

The article “Meta-analysis of field experiments shows no change in racial discrimination in hiring over time” [24] investigates whether hiring bias based on racism has changed over time. Despite claims by scholars that discrimination has decreased over time, the results of the article indicate that there has been minimal change. The results demonstrate that discrimination in hiring has not undergone significant alteration. African Americans with qualifications comparable to those of a white individual received, on average, 36% fewer callbacks when applying for jobs.

The PhD thesis, entitled “Diversity and Inclusion of Women in the Construction Industry” [15], addresses the fact that construction is one of the most male-dominated fields of work. Despite efforts to promote change, the construction industry continues to lose more women than it gains. One potential explanation for this phenomenon is that the construction industry is characterised by a masculine culture, which may include attitudes and behaviours such as aggression and sexual harassment. The article examines several biases that are at the root of this unequal distribution. One of the most significant biases

2 Related Work

is the "prove-it-again" bias, which requires women in this field to repeatedly demonstrate their capabilities in order to be valued by others. Discrimination frequently originates from recruitment practices. For instance, even when projects were assigned a diversity target, women were often hired to meet this objective and subsequently dismissed at later stages of the project.

The article "Gender biases and discrimination while hiring" [12] focuses on hiring bias based on gender. The authors cite the existence of unconscious bias in the selection of personnel, which they term 'implicit bias'. They also identify a bias that occurs because people consciously or unconsciously prefer individuals who are similar to themselves. One of the aims of the hiring bias game is to identify these unconscious processes when deciding on a new employee. The article also notes that in recent years companies have started to hire more women than before, because it creates favourable publicity that improves their image.

2.2 Serious Games about Gender and Race Bias

In their conference paper, "A Review of Serious Games to Educate on Gender Equality" [4], the authors conducted research on a number of serious games that address gender equality. Based on their findings, they developed a prototype in which the player assumes the perspective of a woman navigating everyday situations. As with the majority of the games they reviewed, the objective is to foster empathy and provide direct educational experiences for players. While several of the games they examined addressed gender inequality, none of them did so in the context of recruitment.

The "Anti Sexism Game" [18] is a serious game that addresses the issue of sexism in the workplace. The game comprises two sub-games. In the first, the player is required to complete a questionnaire about their own workplace, with the results indicating whether their work environment is sexist. In the second sub-game, bubbles containing comments are displayed on the screen. If a comment is identified as sexist, the player must pop the bubble. While the game is focused on the issue of sexism in the workplace, its scope extends beyond this to encompass broader issues of inequality and inappropriate comments directed at women in a work environment. It does not, however, address the specific issue of inequality in the recruitment process.

The International Institute of Human Rights has released a serious game, entitled "Interactive game - Fight against racism and discrimination" [13]. This serious game provides an educational experience concerning racism in various contexts of life. Furthermore, the player must select which individuals they wish to travel with on a train. Additionally, there is a recruitment scenario in which the player must determine whether the situation constituted discrimination. In this game, the player receives an immediate response to their choices, thereby becoming immediately aware of being evaluated. This differs from the hiring bias game in this thesis, in which data is collected over the course of a playthrough and then assessed at the end.

2.2 *Serious Games about Gender and Race Bias*

The game “Purpose” [27] is an example of a game that collects data in the background and evaluates the player at a later stage. In this game, the player assumes the role of a squad leader in the context of a zombie apocalypse. At the outset, the player is required to select four members for their squad, subsequently assuming responsibility for their roles, which include gatherer, scout and warrior. The objective is to traverse the map and reach a designated camp on the opposite side. The distribution of ten points between the attributes of agility and strength determines the roles that should be optimally be assigned to the members of the party in order to successfully navigate the terrain and reach the camp. It is inevitable that members of the party will perish over time. However, when traversing the map, pairs of individuals will be encountered, and the player must select one person from each pair to accompany them. Upon completion of the game, the player will be evaluated on their selection of characters based on gender and race. This principle is analogous to the hiring bias game. The “Purpose” feature provides a form of stealth assessment similar to that employed in the hiring bias game. The hiring bias game differs from the hiring bias concept in that it presents a more realistic scenario of hiring. Additionally, the hiring bias game provides a more comprehensive view of the employed population, displaying not only the overall hired individuals but also the distribution of hired people across different job roles. Furthermore, it illustrates the distribution of instances where the player had to choose between a pair of individuals differing in gender or race.

There are numerous serious games that address issues of sexism and racism. The majority of these games do not employ a stealth approach, whereby the player is unaware that they are being evaluated or educated. Instead, they attempt to educate the player directly. “Purpose” employs stealth learning and stealth assessment, which constitutes an essential element of the hiring bias game. Although hiring was briefly referenced in “Interactive game – Fight against racism and discrimination,” none of the games had a primary focus on hiring, which is the focus of the game in this thesis. Therefore, although racism and sexism are already addressed in numerous serious games, a stealth serious game about hiring bias appears to be a novel concept.

3 Design

The objective of the hiring bias serious game is to assess whether the player exhibits a bias based on gender or race when hiring an individual. It is essential that the player is unaware that they are being evaluated while playing the game. Furthermore, the game should be engaging to ensure that the player is fully immersed and more likely to make unconscious decisions. Game design plays a pivotal role in this thesis project and required a significant investment of time.

3.1 Important Aspects for Stealth Serious Games

The conference paper, entitled “A Model for the Development of Stealth Serious Games” [7] presents a model of a stealth serious game. It is recommended that stealth serious games should have a low complexity of rules in order to avoid deterring potential players who may be reluctant to learn rules that are perceived as too complex. Furthermore, the paper posits that enjoyment is a crucial aspect of a serious game. To this end, the game should not be as complex as a simulation, as this may prove overwhelming for novice players. However, it should not be overly simplistic, as this may fail to engage experienced players. One of the most crucial elements of a stealth serious game is stealth learning, which occurs when players engage in a game without realizing that they are actually learning about a subject. Stealth assessment is a valuable technique for facilitating this type of learning. It involves collecting data in the background while players are engaged in a game, which is then assessed later on. This approach is also utilized in the hiring bias game.

3.2 Basic Concept

In the game, the player assumes the role of an entrepreneur who owns a construction company. The objective is to construct a city that encompasses a designated percentage of the total area, which is calculated based on the selected difficulty level at the outset of the game. The difficulty levels are as follows: “very easy” for 30%, “easy” for 50%, “medium” for 70% and “hard” for 100% of the available area. In order to accomplish this objective, it is necessary for the player constantly hire new employees. These employees possess two distinct attributes, which distribute 10 points between each other. Depending on how many points an attribute has, or how evenly they are distributed, an employee can do a job more or less well. The employees also have values that can increase, and if

they reach a certain value, the player loses the worker. If the player loses all his workers, the game is lost. Since we want to observe hiring decisions, we need to create a need for new workers and also provide the opportunity to hire them.

The objective of the game is to facilitate a state of flow for the player. The creation of a flow is contingent upon the degree of challenge presented to the player. It is essential that the challenge is not so overwhelming as to induce frustration. Conversely, the game should not be too easy, as this will result in the player becoming bored and failing to enter a flow state. When a person is in a flow state, their actions tend to become more automatic and unconscious, which could lead to unconscious bias influencing hiring decisions [19].

3.3 Game Scenario and Job Roles

A number of industries have been observed to exhibit a hiring bias. In order to investigate the phenomenon of hiring bias, we sought to identify an industry that would elicit a particular set of stereotypes. To identify an appropriate industry, we examined the distribution of gender and race in specific occupational roles. Ultimately, the construction industry was selected for further analysis. Statistical data from the U.S. Bureau of Statistics for the year 2023 indicates that only 10.8% of employees in this industry are women [32]. The article “Women in Construction: The State of the Industry in 2024” [6] uses this statistic as a reference and claim that only 24.6% of women in the construction industry are employed in construction roles. A further 72.2% of women in the construction industry are employed in sales, management and office roles. In general, it appears that office roles are more commonly associated with women. This is evidenced by the article “Administrative Professionals Day Recognizes 2% of U.S. Workforce, Mostly Women Who Earn Less Than Average Worker” [5], which indicates that roles such as secretaries and administrative assistants are predominantly occupied by women. These roles account for 93-97% of all jobs in these categories.

One of the roles selected for analysis is that of the construction worker. The role necessitates a considerable degree of physical strength. A common stereotype is that men are stronger than women [10], which also contributes to the generation of biased results. The available statistics indicate that women are more frequently associated with managerial roles in construction than with other positions. This may be attributed to the stereotype that women possess superior communication skills [11]. Consequently, the second role selected for the game was that of manager. The third role was that of engineer. Engineering is a discipline within the STEM (Science, Technology, Engineering, and Mathematics) fields, which is currently dominated by men, with twice as many males as females [9].

In the game, the aforementioned roles are represented by symbols, the meaning of which is elucidated to the player in the game’s tutorial. The objective is to enable the player to rapidly ascertain the specific job roles in diverse scenarios, obviating the necessity for textual exposition. The symbols in question are illustrated in Figure 3.1. From left to

right, these are a calculator and moneybags for managers, a hard hat for construction workers, and a checklist for engineers.

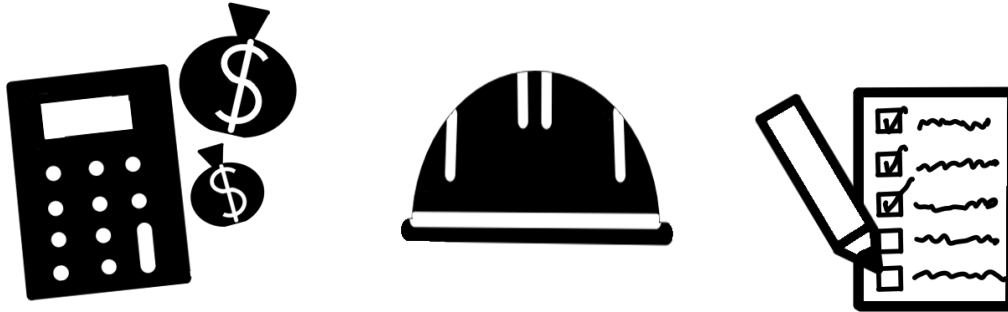


Figure 3.1: Symbols in the game for different job roles.

3.4 Employee Attributes

Each employee is characterised by two attributes, namely university experience and work experience. A total of ten points are shared between them. The distribution of these points is illustrated in a bar comprising ten blocks, as depicted in Figure 3.2. The blocks on the left represent university experience, while those on the right represent work experience. An employee must possess at least one experience point on either side, as this is necessary for the player to ascertain the value represented by the colour of the blocks in the bar.



Figure 3.2: Bar showing the experience distribution of a person.

In addition, the workers possess two further attributes that the player is required to observe: The stress value and a talkative value. The stress value of an employee is increased with each instance of work or relocation to a new site. Both activities result in an increase of five points to the stress value, which represents a significant increase given that the maximum stress value a worker can have is ten. The player may reduce the stress by utilising “bonus payments”, which can be accrued through the construction of houses on construction sites. These payments are distributed to the employees upon the completion of a construction site. The reduction in stress is contingent upon the number of employees within the company and the quantity of bonus payments. In the event that there are more bonus payments than employees, the stress is entirely eliminated. Conversely, if there are

3 Design

more employees than bonus payments, the stress is reduced proportionately, based on the ratio of bonus payments to the number of workers. To illustrate, if the player has eight workers and only four bonus payments, the stress is reduced by five points instead of ten. The stress value of an employee is represented by a bar with a smiley face, which indicates the employee's level of satisfaction with stress. This is demonstrated in the Figure 3.3.



Figure 3.3: Bar showing the stress level of a worker.

The talkative value is a measure of the probability that a worker will report an individual to the relevant authorities. The rationale provided within the game context is that the construction company is managed in a mafia-like manner, wherein the tenets of law and safety regulations are not strictly adhered to. Should the talkative value reach the maximum value of ten, the worker will be automatically eliminated following the completion of a construction site. This is done to prevent the potential informant from disclosing information to the relevant authorities. This value is increased if the worker's stress value is already at its maximum and the worker is either still working or setting up a new site. Furthermore, the possibility of a police investigation exists when a site is being completed. The talk value increases in accordance with the percentage of the site that is secured in case of an investigation. If the site is 0% secure, the talk value increases by ten points. If the site is 50% secured, it increases by five points, and if the site is completely secured, the talk value remains unchanged. The talking value of an employee is represented by a bar with a talking head symbol adjacent to it, as illustrated in Figure 3.4.



Figure 3.4: Bar showing the talkative value of a person.

The distribution of experience values influences the proficiency of an employee in a specific occupational role. Managers attempt to identify new construction sites in close proximity to the existing site on which the team is currently working, with the objective of enabling the player to select them in the future. It is not always the case that they are successful. Their success rate depends on the number of university experience points that the employee has accumulated. The higher the university experience of the employee the higher is the success rate. Construction workers are able to construct houses on construction sites.

The construction of each house results in the receipt of bonus payments, which serve to mitigate the stress levels of all employees. The number of houses that a worker is able to construct is based upon their experience points. The more experience they have, the more houses they can build. Engineers play a pivotal role in ensuring the security of the construction site. A well-secured site can effectively deter any unexpected police raids, thereby reducing the likelihood of employees talkative values increasing. The level of site security is contingent upon the size of the team and the distribution of university and work experience points. In situations where the experience points are evenly balanced, a greater proportion of the site will be secured by engineers.

The concept of interdependence is fundamental to the functioning of the roles within the game. Each role is reliant on the others for progress to be made [25]. The managers are essential for advancement, the construction workers are vital for maintaining a stress-free environment and preventing the loss of employees as a result, and the engineers are crucial for preventing immediate employee loss due to police investigations. Therefore, the player must utilise all roles equally in order to succeed.

3.5 Gameplay

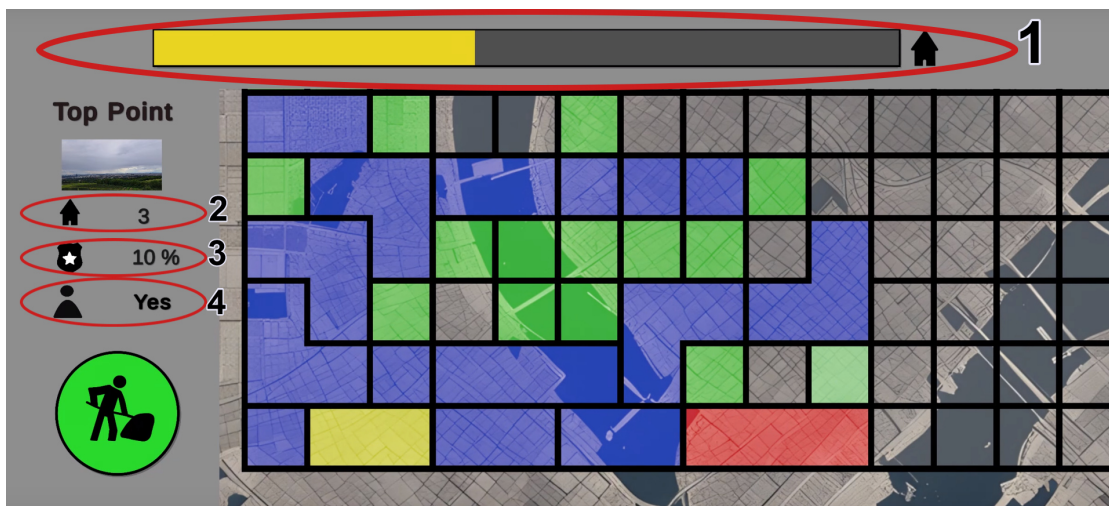


Figure 3.5: The map of the game with construction sites in various states.

In the hiring bias game, the objective is to construct a specified percentage of the area in question. The area is represented on a map divided into tiles. On highlighted tiles the player can start a construction site. As illustrated in Figure 3.5, the fields exhibit varying dimensions and chromatic characteristics, representing construction sites. The blue fields indicate which sites are already under the player's ownership. The green, yellow and red fields are construction sites the play can select to build a new site on. Red sites provide the most houses that can be build, but also has the highest probability of

3 Design

a police investigation. Green fields on the other site provide the least houses that can be build, but also have the lowest probability of a police investigation. The yellow fields are in both categories in the middle. The objective is to instill in the player a sense of territoriality, thereby stimulating the drive to expand their domain [21]. At the top of the map overview, a progress bar (1) indicates the percentage of the required area that has already been built on. On the left, an overview of the current construction site displays a house icon (2), which denotes the number of houses that can be constructed on that site, and a police icon (3), which signifies the probability of a police investigation following the completion of the site. Additionally, a small people icon (4) indicates whether the player can recruit new applicants at that location.

The construction sites on the map are generated procedurally, thereby ensuring that each time the player commences a new game, it is a novel experience. This enhances the game’s replayability [26]. The progress bar is increased to a greater extent by larger fields than by smaller construction sites (red: three tiles, yellow: two tiles, green: one tile). In light of the prevailing tendency towards immediate gratification, it is reasonable to posit that players will be inclined to attempt to complete the game in a shorter timeframe and will be more likely to select a larger field [20]. The larger the field, the greater the number of houses that can be constructed. However, this also increases the likelihood of being investigated by the police. This presents the player with a risk-reward dilemma, whereby the player has the opportunity to progress more rapidly in the game and collect more resources, but also a high risk of losing everything if the player is unable to secure the site sufficiently in the event of a police investigation [34].

The number of houses that can be constructed on a construction site is randomly generated but constrained to a specified range. These ranges are selected with the intention of placing the player in a situation of persistent stress, which is designed to exert pressure on the player. Furthermore, the constant threat of a police investigation will also exert a considerable degree of psychological pressure on the player. However, if the player has engineers who are able to secure a site with absolute certainty, and if sufficient bonus payments for the whole team have been accumulated from a red site, the player will experience a notable reduction in stress, which will enhance their overall enjoyment of the game. This concept is known as the ‘tension and release’ principle, which is based on the catharsis theory, which posits that pleasure is amplified when tension is released [36].

Activation of the button on the left will direct the player to the construction view. Prior to reaching the construction site and upon departure from the same, the player will be presented with a loading screen, as illustrated in the Figure 3.6. This has not to do with a loading process in the background. The view could be loaded instantaneously. The rationale behind utilising a “fake” loading screen is that time is a crucial factor in event segmentation for humans. Therefore, the construction view and map overview should be perceived as two distinct events, separated by the loading screen’s completion time [33].

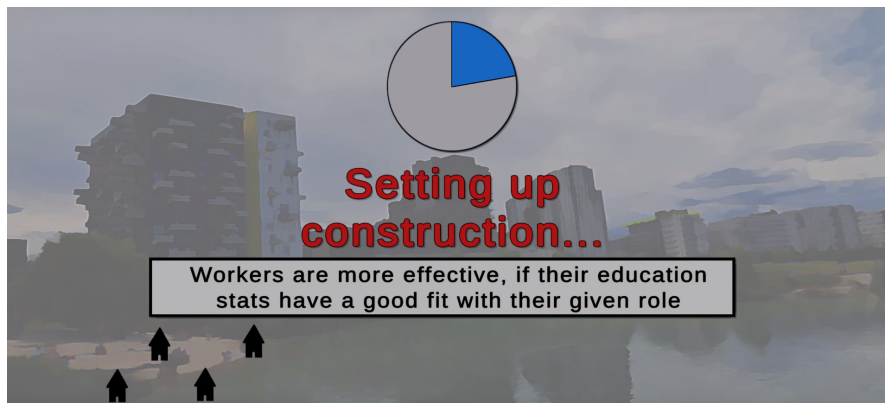


Figure 3.6: Loading screen presented to the user when setting up a construction site.

Figure 3.7 depicts the potential number of houses that could be constructed on the site (1), along with the probability of a police investigation (2). In the upper right corner, an icon representing a moneybag indicates the number of bonus payments (3) that the player's enterprise has. The central element is a button labelled "Finish Construction!" (4), which will only become visible once at least one employee has worked. At the bottom of the screen are buttons, each of which represents a specific job role (5). By selecting one of these buttons, the player can assign the requisite workers to the corresponding job icon and initiate their work, as illustrated in Figure 3.8. Once a worker has been assigned to a task, they are unable to perform that task again until construction on the site is finished. The button on the left, which features house icons (6), will redirect the user to the map overview. This allows the user to consistently ascertain which construction sites are situated in close proximity to their current location.

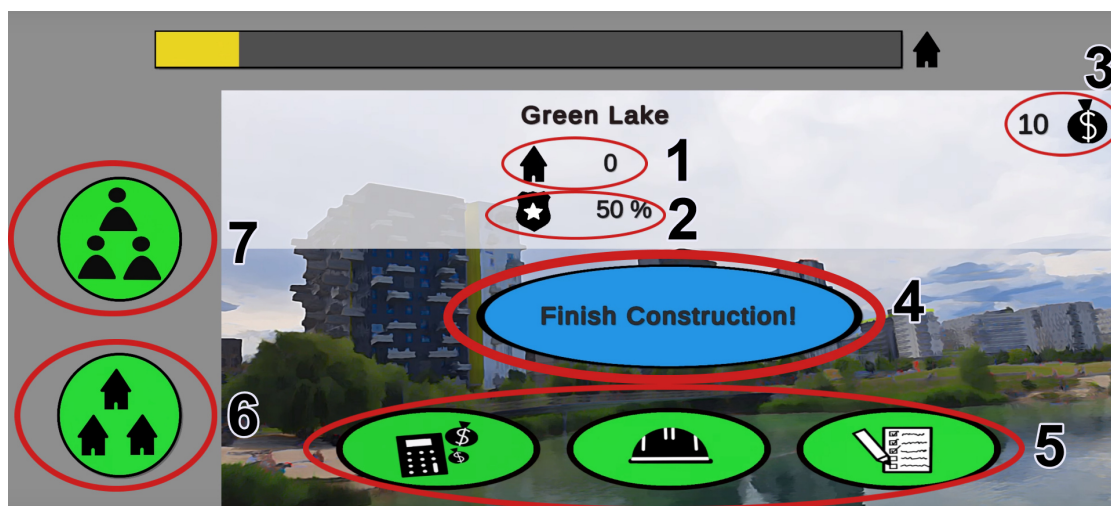


Figure 3.7: Construction view showing options to interact with a construction site.



Figure 3.8: Making an employee work.

The button located above it, which features a representation of three people (7), directs the player to the team overview, as illustrated in Figure 3.9. In this section, the player is able to oversee the management of the company's personnel, assigning them to specific roles by selecting a button with an icon representing the desired position (1). The team overview provides statistical data for each member of the staff (2). Additionally, the option to fire individual staff members is available (3), should the player deem that bonus payments are being wasted on them.

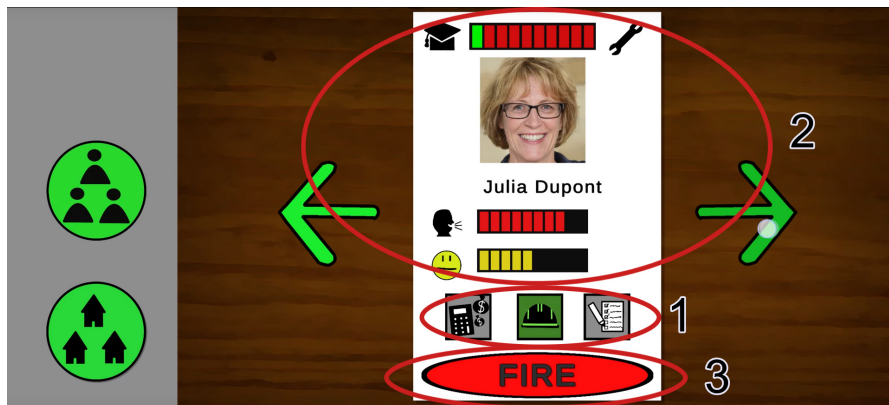


Figure 3.9: Team overview to select between employees.

3.6 Hiring Employees and Stealth Assessment

The most crucial aspect of the hiring bias game is, naturally, the hiring process. Without this, it would be impossible to assess a player's hiring bias. The initial stage of the game in which the player is required to hire personnel is at the beginning of the game. In this

3.6 Hiring Employees and Stealth Assessment

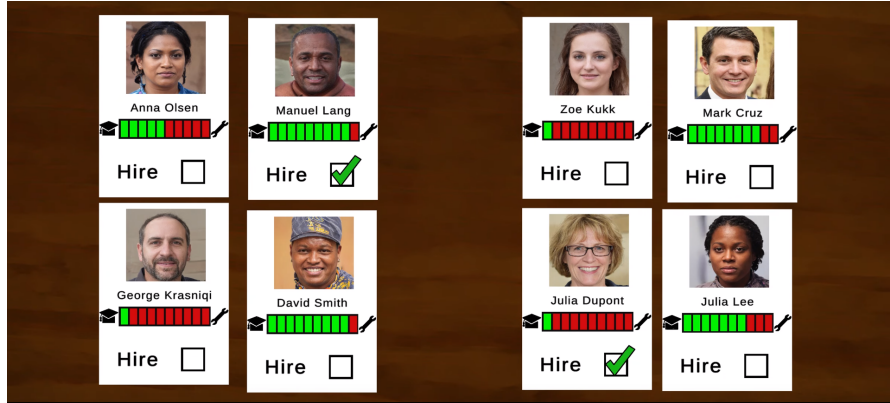


Figure 3.10: Hiring applicants at the start of the game.

phase, the player is required to select four workers from a total of eight. As illustrated in Figure 3.10, the player has the option of viewing “applications”, which provide only limited information, namely the applicant’s face, name, and distribution of the experience attributes. This limited data set does not allow for a comprehensive evaluation of the applicants. The experience distribution is randomly assigned to each candidate. The eight applicants are randomly assigned to the categories of gender and race, with equal representation across both categories. When a set of new employees is finally selected, a statistics game object receives a list of the selected individuals. This process occurs each time a new person is hired.

During the course of the game, the player will be presented with pairs of applicants who are seeking employment with the company. The pairs of applicants are not generated at random. It is a given that each pair will comprise individuals of either different genders or races, but not both. This indicates that there are two pair combinations that are of the same race but of a different gender, as illustrated in the Figure 3.11. Conversely, there are two pair combinations that have the same gender but a different race, as depicted in the Figure 3.12.

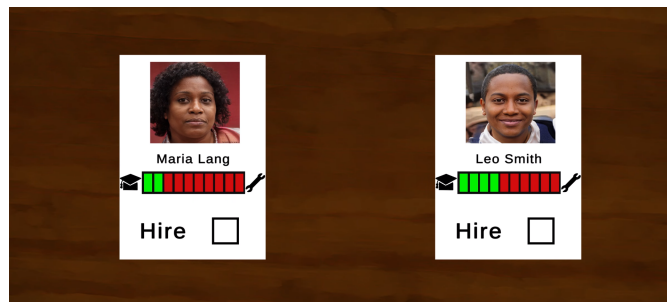


Figure 3.11: Applicant pair only being different in gender.

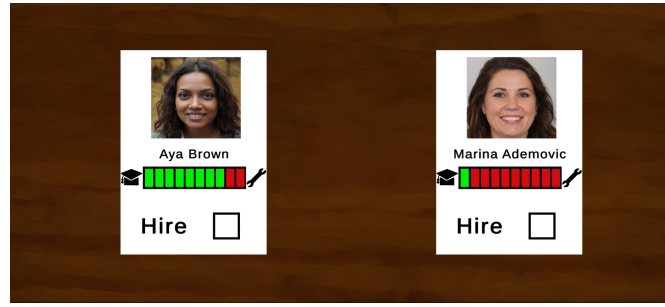


Figure 3.12: Applicant pair only being different in race.

The generation of pair combinations is not arbitrary. The sequence begins with a non-white pair that differs only in gender, followed by a female pair that differs in race, then a white pair that differs in gender, and finally a male pair that differs in race. Subsequently, the process is initiated anew. As with the initial stage of the game, the statistics game object is apprised of the employees who were selected for hire. On this occasion, the statistics object is also informed of the specific employee who was chosen and the distinction between the pair when this individual was selected. This approach enables a more precise evaluation of whether the player’s decision was influenced by gender or race. In the absence of this differentiation, it would be challenging to ascertain whether a choice was predominantly based on one category or the other.

Information regarding the player’s decisions is not solely gathered upon the hiring of an employee. Additionally, the selection of employees for specific roles is monitored. The statistics game object is not informed when the player assigns a person to a job, as it could easily happen that the user accidentally presses the wrong button. Instead, the data is collected only when an employee first works in a particular role.

Upon completion of the game, a screen will be displayed indicating the gender and racial distribution of the employed individuals. The “Hired Workers” category, illustrated in Figure 3.13, depicts the gender and racial composition of all hired personnel, encompassing those recruited at the start of the game and ignoring what the difference was between the applicant pairs. Figure 3.14 illustrates the direct selection bias. On the left, it depicts the number of individuals selected from each gender in pairs of applicants where the only difference was gender. On the right, it illustrates the number of individuals selected from each ethnicity in pairs of candidates where the only difference was race. There is also a gender and race distribution statistic for each job role, as shown in the Figure 3.15 for the role of a construction worker.

The text displayed beneath each pie chart indicates whether the probability of being hired by the player is less than 50% for women and non-white individuals. These two groups are checked for hiring probabilities significantly lower than 50% because papers such as “Comparative Perspectives on Racial Discrimination in Hiring: The Rise of Field Experiments” [23] and “Gender biases and discrimination while hiring” [12] show that

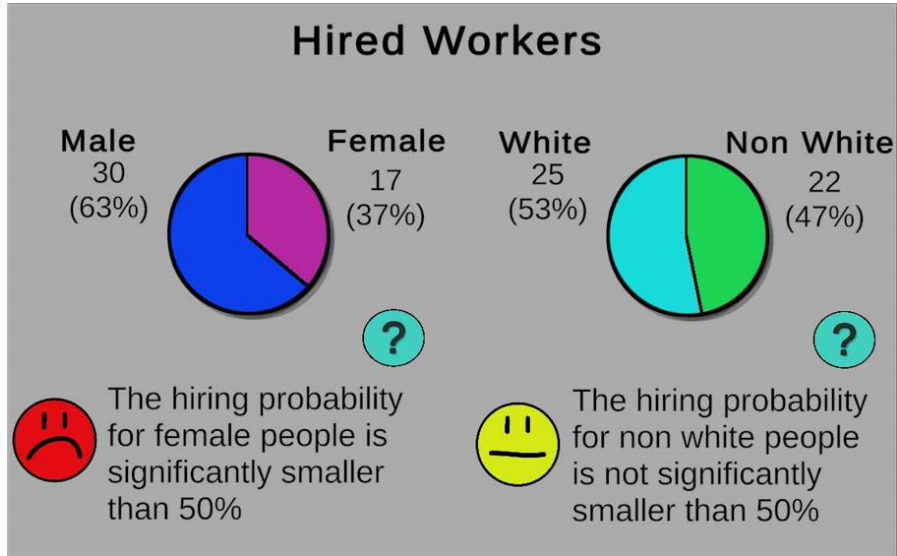


Figure 3.13: Statistics shown for overall hired workers.

these groups are most likely to have a hiring disadvantage due to bias. In the event that the percentage of women or non-white individuals in a given category exceeds 50%, a green smiling emoji is displayed. Conversely, if the percentage is less than 50%, a yellow neutral emoji is displayed. It is important to note that the experience levels of the applicants were randomly generated, and that other random factors may have played a role in recruitment. Therefore, it is insufficient to conclude that the player is biased solely based on the selection of less than 50% women or non-white individuals.

In order to identify any potential bias, a hypothesis test for proportions is employed. This test enables us to ascertain, at a specified level of significance, whether the assertions “I hire people regardless of gender” or “I hire people regardless of race” can be rejected. If this is the case, it can be demonstrated that the probability of the player hiring is significantly less than 50% [8]. In order to test whether the probability of being hired is significantly lower, a left-tailed test is employed for both categories, with the hypotheses selected accordingly. In order to ascertain whether the probability of being hired is significantly less than 50% for women, the following hypotheses are used:

$$H_0 : \text{Hiring Probability for female} \geq 50\%$$

$$H_1 : \text{Hiring Probability for female} < 50\%$$

If we want to check if the hiring probability is significantly smaller than 50% for non-white people then we use following hypotheses:

$$H_0 : \text{Hiring Probability for non white} \geq 50\%$$

$$H_1 : \text{Hiring Probability for non white} < 50\%$$

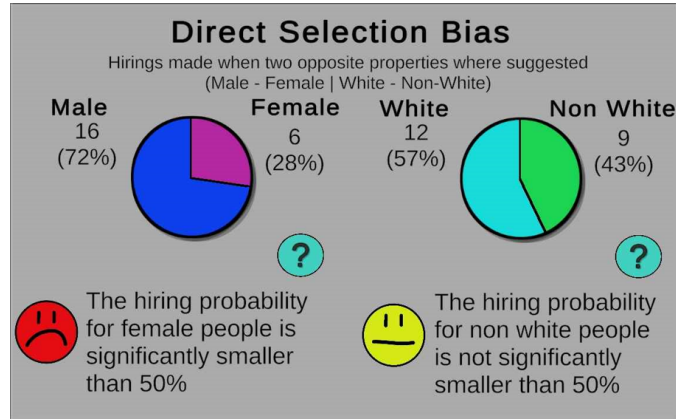


Figure 3.14: Statistics that shows what was chosen when there was a difference in either gender or race.

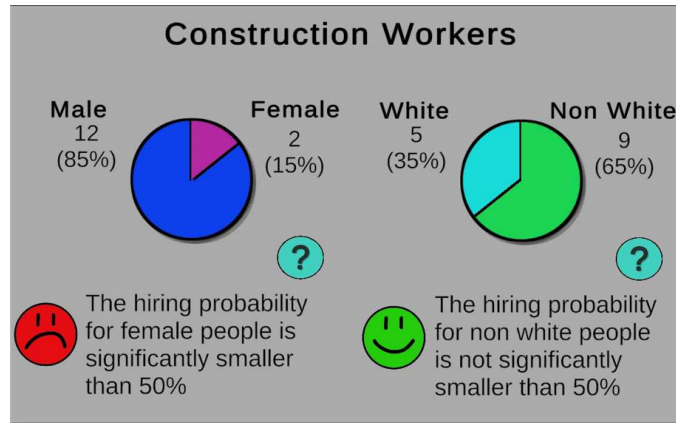


Figure 3.15: Statistics for construction worker job role.

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} \quad (3.1)$$

As can be seen in (3.1), the z-score is computed with the formula for the hypothesis test for proportions. The variable $\hat{p}$ is the percentage of the group specified in the null hypothesis (in our case either women or non-white people) in the total data. p_0 is the value we expect, in our case 50%, because the null hypotheses claim that the hires are at least equally distributed in either gender or race. The variable n is the size of the data. For the significance level we choose the most commonly accepted value which is 95% and gives us $\alpha = 0.05$ [16]. The null hypothesis can be rejected if the p-value of the calculated z-score is less than α . An alternative approach would be to convert the p-value of α to a z-score, and then conclude that if the z-score derived from the formula is less

than that of α ., the null hypothesis can also be rejected. In the event of a rejection of the null hypothesis, it can be stated with 95% confidence level that the individual playing the game exhibited a bias against women or non-white individuals with regard to hiring. Should this be the case, a red sad emoji will appear within the statistics window.

3.7 Winning and Losing

The objective of the game is straightforward. Upon completion of each building site, the game performs a verification process to ascertain whether the requisite number of tiles has been reached. In the event of a successful outcome, the player will be presented with a win screen (see Figure 3.16), from which they can access the relevant statistics. A loss occurs when the player has lost all of their employees, which happens when their talkative value reaches its maximum. When this occurs, the player is presented with a losing screen, as illustrated in the Figure 3.17.

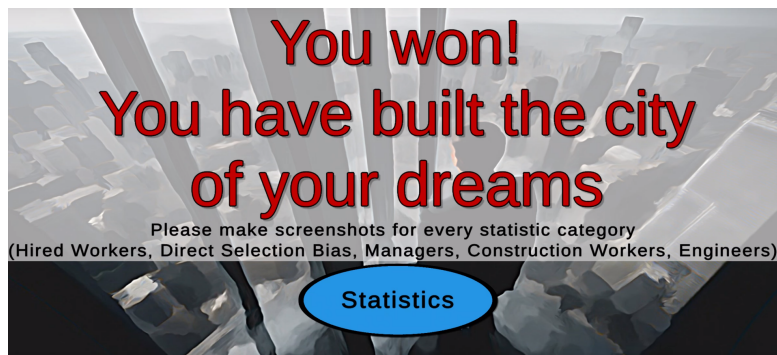


Figure 3.16: Screen shown after winning the game.



Figure 3.17: Screen shown after losing the game.

3 Design

Following a loss, the player is presented with the option of returning to the main menu, from which they may initiate a new game. Additionally, there is the option to “Bail Out”. The “Bail Out” option permits the player to resume gameplay without forfeiting any territory previously acquired. A disadvantage of this approach is that it requires the player to hire new employees and the statistics screen will indicate the number of times the player has been arrested. The player will be subject to a penalty, namely the hiring of a new set of employees. Furthermore, the statistics screen will indicate the number of times the player has been arrested at the conclusion of the game. The “Bail Out” option was originally absent from the game. However, when the game was first played by a small number of individuals, they expressed frustration when they played for an extended period and had to restart the game from the beginning. This discouraged them from continuing to play. The implementation of punishment and frustration can enhance the enjoyment and engagement of a game. However, if the degree of these elements is excessive, it may lead to a sense of stagnation and ultimately result in a loss of interest [3]. For this reason, punishing the player only with the time it takes to create a new team and showing an indicator of how well the game has been played seemed like a good alternative.

4 Implementation

The hiring bias game has been created using mainly the Unity game engine with the long term support version 2022.3.26f1 [28]. The game is an Android application compiled for the minimum API level 22 (Android 5.1 Lollipop [2]) and targets the highest available API level. In order to develop a Unity project for the Android platform, an Android Source Development Kit (SDK) is required. When using the Unity Hub [29], you can directly choose to install components for an Android build, where the SDK is automatically downloaded. For the purposes of debugging, it is possible to utilise the pre-installed emulators, but it is recommended to use the Unity Remote application [31] to be able to test the project on a real mobile device. Make sure that USB debugging is enabled on the phone and that “Any Android Device” is selected for the device in the Unity editor settings. The code can be found on following GitLab Repository: <https://git01lab.cs.univie.ac.at/pesakp00/hiring-bias>. After cloning you can add the folder “Hiring Bias” as a Unity project via Unity Hub. When opening the project with Unity go to files and then build settings. Make sure Android is chosen as the build platform and click on “Build” and you compile an APK of the game. The APK can be found inside the “Hiring Bias” folder. It can then be installed on any Android device that has at least the minimum API Level.

In general, the majority of variables that impact the game and should be easy modifiable by developers are designated as serialized fields. This enables the variables to be altered within the Unity editor without the necessity to modify the code and without making them public. This facilitates the straightforward modification of parameters for generation, work processes or, in general, text and sprites that are to be displayed by a script. Serialized fields also permit the interconnection of game objects within the Unity editor.

4.1 Generation of Construction Sites and People

Construction sites are a significant factor in the hiring bias game. The sites are randomly generated each time a new game is commenced. In order to implement this, it is first necessary to spawn a grid of tiles. The tiles are game objects, comprising a number of semi-transparent squares of varying colours, four black rectangles utilised to delineate a border that forms a square, and a 2D collider. Additionally, a mono behaviour script is included, which is responsible for modifying the colour, processing input and exchanging data with the construction site to which a tile belongs. The semi-transparent squares serve to indicate whether a given tile is owned by the player’s company, is unavailable,

4 Implementation

or is available, and also to convey the difficulty level associated with that tile. The semi-transparent nature of these squares permits the placement of an image within the game world that is visible through the tiles, thereby enabling the representation of a city from an aerial perspective. One of the semi-transparent squares should be white and have a sorting layer with a higher rendering priority than the other squares. The white square is used to highlight when the player selects the tile and is displayed in combination with another coloured square. However, if the highlight has the same sorting layer as the other squares, a graphical anomaly may occur. This could manifest as three squares belonging to the same construction site having different colours, despite them all sharing the same colour designation. To counteract this, the white highlight must be given a sorting layer with a higher rendering priority than the other squares. Only one square of a given colour can be active next to the white highlight square. The other remaining squares of the tile are inactive and therefore not visible to the player.

A base tile game object is saved in Unity as a prefab and assigned to a grid manager, a game object solely responsible for generating the grid with tiles, with a `serialize` field. This enables the method `Instantiate(BaseTile original, Vector3 position, Quaternion rotation)` to be called, where `original` is the prefab and `rotation` is set to `Quaternion.identity`. The original prefab, together with the vector representing its position and the quaternion representing its rotation, can be instantiated. In order to spawn the grid, it is first necessary to decide the number of tiles that are to be placed on the x and y axes. In the hiring bias game, there are 16 tiles on the x-axis and 9 tiles on the y-axis, resulting in a total of 144 tiles. However, these variables can be modified according to the developer's preferences (it is necessary to adjust the camera's bounding box to accommodate this change). Additionally, a starting coordinate for the initial tile on the map must be specified. Subsequently, two for-loops are employed, one for each axis, to generate a tile for each integer value of x and y between zero and the desired number of tiles on that axis. In world position, the tiles are created with the space of the scale value (there is an x and y scale value) of a tile between them. This allows the developer to modify the scale and even the ratio of a tile. Each instantiated tile is stored, with its integer coordinates as the key, in a dictionary. This dictionary contains all the tiles in the grid and is then passed to a construction site generator game object.

A construction site is a game object that stores a list of tiles representing the site. It is responsible for ensuring that these tiles act as a unit when they change colour. Furthermore, it stores information regarding the number of houses that are available for construction, the probability of a police investigation, neighbouring sites, and whether there are any applicants. The building site generator game object contains a construction site game object prefab, which can be used to instantiate new building sites. The initial step is to reserve the first tile in the grid as an accessible construction site. Then the sites with the most tiles are created first, i.e. the hard sites, then the middle sites and at least the easy sites. To create construction sites with different difficulty, a method called `CreateConstructionSites(Difficulty difficulty, int constructionCount, int wantedNeighboursCount)` is called.

4.1 Generation of Construction Sites and People

```
1 private void CreateConstructionSites(Difficulty difficulty, int
  constructionCount, int wantedCenterTileNeighboursCount) {
2     int createdConstructionCount = 0;
3
4     while (createdConstructionCount <= constructionCount) {
5         //Get a random position
6         Vector2Int newConstructionCoord = CreateRandomPosition();
7         if (newConstructionCoord.Equals(startCoord)) {
8
9             continue;
10        }
11        BaseTile tile;
12        //Try getting a tile at that position
13        if (!tileMap.TryGetValue(newConstructionCoord, out tile)) {
14            continue;
15        }
16
17        //Ignore if the tile already has a construction site
18        if (tile.hasConstructionSite()) {
19            continue;
20        }
21        //Get possible neighbours of a centered tile in a construction
site
22        Dictionary<Vector2Int, BaseTile> neighbours =
GetAvailableNeighbours(newConstructionCoord);
23
24        //If there aren't enough possible neighbour tiles, try the whole
process again
25        if (neighbours.Count < wantedCenterTileNeighboursCount) {
26            continue;
27        }
28
29        //Only chose the needed amount of neighbours
30        neighbours = RestrictNeighboursToSize(neighbours,
wantedCenterTileNeighboursCount);
31
32        //Generate a site with a given difficulty
33        ConstructionSite generatedConstructionSite =
GenerateConstructionSite(difficulty);
34
35        //Connect the random selected tile with its selected neighbours
and assign them a construction site
36        ConnectTilesWithConstructionSite(generatedConstructionSite, tile,
neighbours);
37
38        //Increases the count for a created Construction Site with given
attributes
39        createdConstructionCount++;
40    }
41 }
```

Listing 4.1: Method to create a construction site with a given difficulty.

4 Implementation

In this method, the parameters determine the difficulty level of the construction sites, the number of neighbours a central tile (the tile from which the whole construction site is created) should have, and the number of sites of this type that should be present on the grid. Subsequently, a tile with a randomly generated coordinate is selected from the grid. In the event that the selected tile is the start tile, does not exist on the grid (that shouldn't happen, but it's a safety measure), or has already been assigned to a construction site, the process is restarted from the beginning. Subsequently, the function `GetAvailableNeighbours(newConstructionCoord)` will select all neighbouring tiles of the chosen tile that have not been assigned to a construction site. In the event that the number of available neighbours is not equal to or greater than the number of neighbours required, the process must be initiated anew from the initial stage. In the event that the number of neighbours exceeds the requisite number, only the necessary number of neighbours will be randomly selected from the list of neighbours in the method `RestrictNeighboursToSize(neighbours, wantedCenterTileNeighboursCount)`. Subsequently, a construction site game object is created. To the construction site then will be assigned the number of houses, a percentage chance of a police investigation and a boolean indicating whether there are applicants on the site. The number of houses assigned to a construction site game object is a random value within a range that is dependent on the difficulty of the site. Additionally, each site is assigned a name and an image, selected at random from a list of scriptable objects. The scriptable objects contain both a name and an image, as the names of the sites are related to the images. Subsequently, the tiles selected for the site are stored in a list of the game object. Additionally, the border rectangles where the selected tiles intersect are also stored, as when a site is discovered, these rectangles can be deactivated, and the player will see a connected site. This method is used for hard and middle sites.

For easy sites the process is analogous, albeit with a key distinction. Rather than randomly selecting tiles, the algorithm iterates over each tile in the grid. In the event that a tile has not yet been assigned a site, it is designated a construction site for easy difficulty. Once all tiles have been assigned a site, the process is repeated for all tiles in the grid. In this iteration, the construction sites are assigned their immediate neighbours, thus facilitating the identification of neighbouring construction sites at a later stage. For each tile that is iterated over, the construction sites of the neighbouring tiles are passed to the construction site that represents it. These neighbouring sites are stored in a set, ensuring that no site is stored more than once. Additionally, it is necessary to ensure that the site does not add itself as a neighbour, which could occur due to adjacent tiles belonging to the same site.

A person game object stores university experience (work experience is just ten minus university experience), the stress level and the talkative value. It also stores an image and a name of the person. The images of the people are not real people, but have been generated on the website thispersondoesnotexist.com [22], which uses styleGAN 2. StyleGAN 2 is a generative adversarial network developed by NVIDIA that is used to create data-driven images [14]. Every time the website is reloaded, `thispersondoesnotexist`

4.1 Generation of Construction Sites and People

uses this technology to generate artificial images of people. During development of the hiring bias game, it became a problem to generate African-American looking people, as they seemed to be generated with a significantly low frequency. Studies have shown that generative AIs such as Midjourney, Stable Diffusion and DALL-E 2 have a gender and race bias when generating images of people [37]. Consequently, it is plausible that `thispersondoesnotexist` also displays this phenomenon. As the website in question is a Uniform Resource Locator (URL) for an image, a simple Python script was employed to facilitate the downloading of multiple images from the site, thus streamlining the process of selecting the desired images. The names were generated by taking a list of a few first names and a list of surnames and creating all possible options for a full name.

The individual game objects are created in a person generator game object with the assistance of a person prefab. Additionally, the generator stores four scriptable objects, each containing sprites of people classified into distinct categories. These categories encompass non-white women, non-white men, white women, and white men. To determine the appropriate category for person generation, a sprite is selected, and the `GenerateRandomPersonWithSprite(Sprite randomSprite)` method is invoked with it.

```
1 private Person GenerateRandomPersonWithSprite(Sprite randomSprite) {
2
3     //Get gender for the sprite
4     Person.Gender gender = GetGenderForSprite(randomSprite);
5
6     //Get race for the sprite
7     Person.Ethnicity ethnicity = GetEthnicityForSprite(randomSprite);
8
9     //Get name depending on gender
10    string name = GetRandomNameForGender(gender);
11
12    // Create the game object
13    Person generatedPerson = Instantiate(personPrefab, Vector3.zero,
14    Quaternion.identity);
15    generatedPerson.name = "Person: " + name;
16
17    //Set the values
18    generatedPerson.SetImageSprite(randomSprite);
19    generatedPerson.SetName(name);
20    generatedPerson.SetGender(gender);
21    generatedPerson.SetEthnicity(ethnicity);
22
23    //Generate random experience, stress and talkative value
24    SetRandomizedBaseStats(generatedPerson);
25
26
27    return generatedPerson;
28 }
```

Listing 4.2: Method to create a random person with a given sprite.

4 Implementation

The gender and race variables, which are both enumerations, are set according to the sprite passed in the parameter. The name is a random full name dependent on the gender. The names gets generated by having two list of strings containing first names and last names and storing every potential combination in a separate list. In `SetRandomizedBaseStats(generatedPerson)`, the university experience is randomly generated. A stress and talkativeness score is also randomly generated, but below five points.

4.2 Touch Controls

The game utilises the new input system package [30], which must be installed from the Unity package manager. The system permits the straightforward interchange of diverse input types, including touch, keyboard, and gamepad. In order to utilise this functionality, it is necessary to create an input actions asset within the designated assets folder and select the appropriate checkbox to enable the generation of a C# class. Subsequently, three distinct actions were incorporated into the action map for the initial and subsequent touches on the touchscreen. The first is the initiation and release of a touch, which is signalled by two different callback functions, which is possible by choosing a button action type. The second reads a two-dimensional vector for the position of the touch. The third reads the delta, which is the difference vector between the two positions of the last and current frame, of a touch as a two-dimensional vector. All input-related operations are handled by a touch input game object, that allows to modify the controls without the need to modify the script of any other game objects.

The primary elements that are influenced by touch interactions are the tiles and the applications that are selected when an employee is chosen. Each tile and application is equipped with a dedicated 2D collider. In order to ascertain whether the player has made contact with a collider, it is necessary to convert the position of the touch from screen space to world space. Subsequently, the position in world space is evaluated to ascertain whether it is within the confines of a collider. In the event that the touch was applied to a collider, a comparison can be made between the collider in question and that of a game object. Should the collider belong to a game object, the process associated with that particular object can be initiated. This is indicated by the touch input game object, which has a method entitled `IsTouchHittingCollider(Collider2D collider)`. This method can be called by any other game object, with the aforementioned collider acting as a parameter.

```
1 public bool IsTouchHittingCollider(Collider2D collider) {  
2     Collider2D touchCollider = Physics2D.OverlapPoint(  
3         GetFirstTouchPositionInWorldSpace());  
4     return (touchCollider == collider);}  
5 public Vector3 GetFirstTouchPositionInWorldSpace() {  
6     return Camera.main.ScreenToWorldPoint(GetFirstTouchPosition());}
```

Listing 4.3: Method to check if a touch is on a collider.

Upon the activation of a given tile, the pertinent information regarding the site in question is then loaded into an overview user interface. Additionally, the Game Manager, a game object that facilitates the interconnection of all processes within the game, will store the selected site as a variable so that it can load the site overview UI when the player presses the button to create a new site. Additionally, all tiles associated with the selected site will be highlighted by the semi-transparent white squares of its stored tiles until a different site is selected. Given that the construction view is merely a user interface window on a canvas that is initiated by pressing a button, the game world with its grid is situated behind it. In order to prevent the player from selecting a tile inadvertently when in another view, it is necessary to implement a method for checking whether there is a UI covering the tile. If there is, the input can be ignored by the tile. The touch input game object provides the method `IsGUIInFrontOfTouch()` for this purpose.

```

1 public bool IsGUIInFrontOfTouch() {
2     return EventSystem.current.IsPointerOverGameObject();
3 }

```

Listing 4.4: Method to check if a touch is made on the UI

As Unity issues a warning if this method is employed in the release callback of a touch, it is invoked in the update method, where the return value is assigned to a Boolean variable. If the assigned variable is determined to be true in the release callback, the process of handling a touched tile will not commence. Regrettably, this approach is ineffective when applied to the application game object, given that it consists of UI elements that are also identified by this method. To avoid selecting applications accidentally while being in the tutorial, a static boolean variable stores if the tutorial is active. Consequently, when the tutorial UI elements are displayed within the game, the process responsible for handling a touch on the application is unable to resume until the static variable is changed by the tutorial elements when tutorial has concluded. Unity handles touches on GUI objects such as buttons. The only thing needed to be change is within the `EventSystem` `GameObject`, which is automatically generated by Unity when you create a UI object, is to replace the “`StandaloneInputModule`” with the “`InputSystemUIInputModule`” to make it compatible with the new input system.

On the grid, the player can move and zoom by finger movement. This is handled by a mono behaviour script attached to the camera. To distinguish whether a player is moving or zooming, it looks at whether only one or two fingers are touching the screen. One finger indicates moving around the screen, two fingers indicate zooming. The movement is calculated by subtracting the current touch position in world space from the touch position in world space when the first touch was made. This results in the finger being at the same position in the game world as it was at the start of the touch, even if it is moved around the screen, which provides the effect as if the player was scrolling through the world. To ensure that the player does not accidentally select a tile while simply moving around the grid, the tiles count down from a given time from the start of the touch. If the touch is released before the given time has elapsed, this indicates that the touch was made to press on something, otherwise the tile will ignore the touch, because if the player

4 Implementation

touches the screen for a long time it indicates that movement is in progress. In order to zoom, it is necessary to calculate the length of the distance between the two touch positions of the current and the touch positions of the previous frame (can be calculated by taking the current position and subtracting it with delta). Then the difference of the length of the distances is calculated. This value, multiplied by a factor to make it not too sensitive, is then subtracted from the orthographic size of the camera, giving the effect of zooming.

Since we do not want the player to be able to scroll away from the predefined play area of the grid, the camera has a boundary within which it is allowed to move. We want the player to be able to see only the edges of the grid, and no further. The problem is that the game can be played on multiple phones with different aspect ratios, and we only have the x and y coordinates of the camera, which represents the centre of the screen. So we take the x and y coordinates of the borders in the grid, maybe adjusting them a bit so that the GUI is between the screen and the grid borders. Then we try to calculate the distance from the centre of the screen to the edge of the phone screen. To do this, we convert the viewport coordinate at $x = 1$ and $y = 1$, which represents the top right corner of the screen, into a world coordinate. Subtracting this coordinate from the camera coordinate gives us the x and y distance from the centre of the screen to the edge in world coordinates. With this knowledge, the corresponding span width is added to the x value of the left edge and the y value of the bottom edge, while the x value is subtracted from the x value of the right edge and the y value of the top edge. Unity sometimes calculates the correct world coordinate from a viewport coordinate only after a few frames the game scene was started. To avoid a wrongly calculated border, the calculation is, instead of being called in the start method, called after a few frames (less than 30) have passed in the game. This is not a big problem, as the player will most likely still be in the tutorial or application view, where there is no movement anyway.

4.3 Hiring and Working

In the context of the hiring process, the game object designated as “table” assumes a pivotal role. The table is comprised of a background image that evokes the visual characteristics of a table surface, along with eight distinct application game objects. The aforementioned application game objects are initialised with a person game object, which displays the name and image of the individual in question. Additionally, the experience is indicated in the block bar. A block bar is a user interface (UI) game object comprising a dark background square and ten smaller squares. The bar displays a two different colours, which can be selected within the Unity Editor through a serialized field for either the left or right side of the bar. A method may be invoked in which a parameter is passed indicating the number of blocks to be allocated to the left side, with the blocks subsequently coloured according to the colour assigned to the left side and the remainder coloured according to the colour assigned to the right side. The bar is also employed in the worker overview. A “superbar”, derived from the block bar, is capable of modifying the

color of the blocks on the left side in accordance with the magnitude of the passed value. This is utilized in the talkative bar of the worker overview, and its derived bar, “emojibar”, exhibits a corresponding emoji in accordance with the value, which is employed in the stress bar.

When an application gets touched, the game object is translated to the center of the table and scaled up. Additionally, a grey background is displayed to indicate to the player that the other applications are not in focus. Subsequently, if the player then presses on a location other than the application, the application is relocated to its original position and scaled down by a factor of one divided by the previous scale value. In the event that the application is touched while in focus, a check mark is displayed on it by activating a check mark sprite. Additionally, the stored person is transferred to the table, where it is stored in a list. Upon subsequent application of pressure, the check mark is removed and the table is duly informed that the individual is no longer selected. Consequently, the table proceeds to remove the individual from the list. Once the requisite number of individuals has been recorded in the list, the button game object is designated as active. Upon pressing the button, the camera is translated to the position of the grid and the chosen individuals are conveyed to the game manager game object, which adds them to a company game object. Additionally, all applications game objects except two are deactivated. These two are positioned at the location of two empty game objects that are also present in the table game object, facilitating easier modification of their positions in the editor. Furthermore, they are scaled up slightly. Upon initiating a new construction site with applicants, a new pair from the person generator is generated, differing only in either race or gender. When generating with the method, it is stored in an enum variable in which category the pair is different, so that the table can send the chosen people depending on the category to the statistics game object.

```

1 private void ApplicantPairSetUp() {
2     List<Person> applicants = new List<Person>();
3     Tuple<Person, Person> personPair = new Tuple<Person, Person>(null,
4         null);
5
6     switch (personSelectionPhase % SELECTION_VARIANT_COUNT) {
7         case 0:
8             personPair = personGenerator.
9             GenerateRandomNonWhiteMaleAndNonWhiteFemalePair();
10            statsState = StatsState.GENDER;
11            break;
12         case 1:
13             personPair = personGenerator.
14             GenerateRandomWhiteAndNonWhiteFemalePair();
15            statsState = StatsState.RACE;
16            break;
17         case 2:
18             personPair = personGenerator.
19             GenerateRandomWhiteMaleAndWhiteFemalePair();
20            statsState = StatsState.GENDER;
21            break;
22         case 3:

```

4 Implementation

```
19         personPair = personGenerator.  
GenerateRandomWhiteAndNonWhiteMalePair();  
20         statsState = StatsState.RACE;  
21         break;  
22     default:  
23         personPair = personGenerator.  
GenerateRandomWhiteMaleAndWhiteFemalePair();  
24         break;  
25     }  
26     personSelectionPhase++;  
27     applicants.Add(personPair.Item1);  
28     applicants.Add(personPair.Item2);  
29  
30     SetUpApplications(applicants);  
31 }
```

Listing 4.5: Method to set up a person pair

```
1 private void SendTeamToStatistics() {  
2     foreach (Person worker in team) {  
3         //Send in general the chosen worker  
4         OnWorkerChosen?.Invoke(this, new Statistics.  
OnPersonChangeEventArgs {  
5             changedPerson = worker  
6         });  
7         //Send the chosen worker with difference in race in the pair  
8         if (statsState == StatsState.RACE) {  
9             OnRaceStatWorkerChosen(this, new Statistics.  
OnPersonChangeEventArgs {  
10                 changedPerson = worker  
11             });  
12         }  
13         //Send the chosen worker with difference in gender in the pair  
14         if (statsState == StatsState.GENDER) {  
15             OnGenderStatWorkerChosen(this, new Statistics.  
OnPersonChangeEventArgs {  
16                 changedPerson = worker  
17             });  
18         }  
19     }  
}
```

Listing 4.6: Method to send the selected applicants to the statistic class

Subsequently, the hired employees can be assigned job roles in the worker overview by pressing a button that represents a category. This action results in an enum variable being changed in the currently selected person game object, which represents a job role. Upon the player then pressing a button for a job in the site overview, the employees in the company are iterated, and those with the job represented by the button are displayed in the window where they can be selected for work.

Upon the activation of a work button, a JobExecutionManager script is initiated, which oversees the management of the individual's assigned tasks in accordance with their stored

work enum. Additionally, a stress value is augmented by a predefined increment, and upon reaching its maximum, the employee's talkative value is also increased. Furthermore, a Boolean attribute within the person game object signifies that the player is unable to engage in further work within the current construction phase.

When a manager executes a job, the job execution manager initiates an iterative process to ascertain whether the selected site has neighbouring sites that are not yet visible. This is achieved by traversing the stored neighbouring sites of the current site and evaluating whether they have been marked as visible through the use of a Boolean variable. In the event that this is the case, the value of the university experience value of person game object is then taken, which is a value between zero and nine. Additionally, a random number between zero and ten is generated using the `Random.Range(int minInclusive, int maxExclusive)` function. In the event that the randomly generated value is less than the university experience value, a randomly selected neighbour from the list of neighbouring sites associated with the currently selected site will be made visible. This enables the site to be visible on the grid and allows the player to click on it. The person object that has worked is also sent to a statistics game object that stores the person in a set for individuals who have worked as managers.

In the case of a construction worker's work being performed, the work experience value of the person game object is initially calculated and then divided by nine in order to obtain a float value. Subsequently, the minimum and maximum values for building houses are multiplied by the aforementioned float value and the result is converted to an integer. Subsequently, a random value is selected from the range defined by the two mentioned values using the `Random.Range()` function. In the event that this value exceeds the number of available houses on the site, it is subjected to a clamping operation, whereby it is reduced to the available number of houses on the construction site game object. The value is then subtracted from the number of houses available on the site and added to a bonus payment integer stored in the company game object. Additionally, the statistics game object maintains a record of all individuals who have worked as builders.

For the engineer job, the absolute difference between university experience and work experience is calculated. The result is then subtracted from ten and then multiplied by a random value between the minimum and maximum number of people an engineer can be responsible for, which are variables given by a serialized field. The result of this calculation is then added to a security points variable in the company game object. The person object that has worked is then sent to a statistics game object and stored in a set for people who have worked as an engineer.

4.4 Finishing a Construction Site

Upon the conclusion of a construction site, the game manager game object is required to undertake a series of prescribed procedures. Firstly, a random value between zero and one is generated to ascertain whether a police investigation is underway. In the

4 Implementation

event that the randomly generated value is less than the percentage chance stored in the construction site object, a police investigation is initiated. In this instance, the company multiplies the number of employees by ten (the maximum value a chat value can have). The security points are then divided by this result, resulting in a float that indicates the percentage of ten points secured for each individual. This float is then subtracted from one, multiplied by ten, and converted to an integer. The resulting number is then added to each employee's talking score. The safety points are then set to zero.

Subsequently, the number of tiles on the currently selected construction site is added to the total number of tiles the company has built on, which is stored in the company game object. In the event that the construction site was previously visited by the player, the number of tiles will not be added. To identify this, the site has a Boolean variable that is set to true when a construction is completed and is then checked by the company game object before adding the value.

Subsequently, the company eliminates any individual whose talkative value exceeds ten. The eliminated employees are then placed on an additional list, which is subsequently presented to a view that displays the individuals who have been eliminated. The stress scores of the employees are reduced through the implementation of bonus payments. The stress value is reduced by a factor of ten, unless the total bonus payments are insufficient to cover the number of employees in the company. In such instances, the quantity of bonus payments is divided by the number of employees, thereby yielding a float value. Subsequently, the aforementioned float value is multiplied by ten and converted to an integer. This integer represents the value that is subtracted from the stress value of each employee. The stress value is subsequently clamped so that it can be less than zero. The amount of bonus payments in the company is then subtracted from the number of employees and clamped so that also it can't be less than zero. The Boolean that indicates whether a worker has already worked is set to false for each worker, thus enabling them to work again in the next construction phase.

Then the status of the player is evaluated to ascertain whether they has lost the game. This is achieved by verifying that the size of the list utilized to store the employees in the company game object is equal to zero. In the event that this is the case, a game object displaying the game over view will be initiated on the canvas. Should then the "Bail Out" option be selected, the camera will teleport to the table area in order to select new team members. Additionally, the game manager sets a value for the number of times the process should be repeated. Consequently, each time the player selects a member, the method responsible for picking up the hired members list and placing the camera over the grid will check this value. If the value is greater than zero, it will be reduced, and the camera will be placed back at the table game objects position.

In the event that the player has not yet been defeated, the game proceeds to ascertain whether they have emerged victorious. This is achieved by checked whether the variable designated as "number of tiles wanted" is equal to or greater than the number of tiles that the company has already constructed on. If this is indeed the case, the victory screen is

then activated. Once the board has been created, the number of desired tiles is calculated by multiplying the number of tiles on the board by the percentage of desired tiles. This percentage is a static float variable in the company, set when the player selects a difficulty level at the start of the game. As the difficulty selection occurs in a separate scene, the variable must be static to be passed to another scene, as only static variables are saved independently of scenes.

4.5 Statistics Assessment

As seen in the hiring and working processes, there is a statistics game object that collects people objects who have been hired or worked in a job. This happens as a result of events that are fired when a hiring or working process happens. These events contain a person object as an argument. The individual is then stored in a set for all hired workers, workers selected from a pair with a gender difference, workers selected from a pair with a race difference, construction workers, managers and engineers, depending on which event was fired. When you click the button to view the statistics after winning, a game object is initialised that displays the statistics. For each category, except the “direct selection bias”, it gets the same set from the statistics object twice. This is due to the fact that distinct sets are employed when assessing the number of individuals from each gender and race who have been selected. The rationale for this approach is that the “direct selection bias” category employs two distinct sets: one comprising individuals when selecting from a pair exhibiting a gender discrepancy and another comprising individuals when selecting from a pair exhibiting a racial discrepancy. Subsequently, the sets are iterated for each category in order to ascertain the number of individuals within each racial and gender group. Subsequently, a circle image each is filled radial with the percentage of female and non-white individuals. The construction of a pie chart is achieved by the addition of a second circle of a different colour behind each circle. In order to ascertain whether the probability of being hired is significantly less than 50%, a method, `CheckIfIsSignificantlySmallerThanFiftyPercent`, has been developed. This method returns true if the hypothesis test for proportions indicates that the z-score is below a z-score of a left-sided test with a 95% confidence interval.

The set `people` is the set from which the number of females or non-white people was counted. `countOfPeopleOfCategory` is the number of women or non-white people. The variable `p_value` is an out variable, which means that the caller of this method gets this variable as well as the return value. Basically the method is the hypothesis test for proportions, where p_0 and the z-score are hard coded to reject the null hypothesis. Since neither the Unity math classes `Math` nor `Mathf` provide a hypothesis test, the algorithm has to be selfmade or implemented through an external library. In this project the method is selfmade. Since the player should be able to see the p-value, but Unity’s math classes do not provide a conversion from a z-score to a p-value, a dynamic linked library had to be added for this task. The `MathNet.Numerics` [17] library provides a conversion method with `Normal.CDF(0, 1, z-score)` that returns a p-value for a given z-score. To use it,

4 Implementation

you need to download the library, which is downloadable as a nupkg date format. Simply rename the file extension to dll. In Unity, create a folder called “Plugins” in the asset folder and put the file in there. Now you can use the methods of `MathNet.Numerics` in your Unity scripts.

```
1 private bool CheckIfIsSignificantlySmallerThanFiftyPercent(HashSet<
    Person> people, int countOfPeopleOfCategory, out double p_value) {
2     //The percentage the nullhypothesis says that something is bigger or
    equal to it
3     float p_0 = 0.5f;
4     // The percentage of the category in the list
5     float p_d = (float)countOfPeopleOfCategory / (float)people.Count;
6     // Z-score for 5%
7     float failValue = -1.645f;
8     // hypothesis test for proportions
9     float z = (p_d - p_0) / Mathf.Sqrt((p_0 * (1f - p_0)) / people.Count
    );
10    // Get p-value for calculated z-score
11    p_value = Normal.CDF(0, 1, z);
12    return z < failValue;
13 }
```

Listing 4.7: Method to check if the hiring probability is significantly smaller than 50%

5 Evaluation and Discussion

A user evaluation was conducted with the objective of testing the hiring bias game, with a total of 31 participants. The gender distribution of testers is as follows: 13 (41.9%) are female, 17 (54.8%) are male, and 1 (3.2%) identifies as transgender. The majority of participants are from Austria and Germany, with each country contributing 14 individuals (45.2%). The remaining participants were from Italy, Poland, and Switzerland, with each country contributing one individual (3.2%). The majority of participants (25, representing 80.56% of the total sample) were between the ages of 18 and 25, indicating that the tested population is relatively young.

5.1 Setup

A survey was administered to the participants, wherein the initial task was to play through the game for the first time and to emerge victorious. The participants were presented with a version of the game in which only the “easy” difficulty setting (having to start a construction site on 50% of the available tiles on the grid) was available. Upon reaching the win screen, participants were prompted to take a screenshot of each of the statistics categories. Subsequently, they were presented with a series of questions employing a Likert scale (1 = strongly disagree, 5 = strongly agree). These questions pertained to the existence of hiring bias based on gender or race, the tendency to judge individuals based on race or gender, and the extent to which the game had influenced a more cautious approach to judging others based on their appearance.

To assess the presence of hiring bias among users, we employ the hypothesis test for proportions. The statistical overview at the end of a playthrough already shows whether the hiring probability of a user for female and non-white individuals is significantly lower than 50% across all job roles. Additionally, we aim to ascertain whether there are instances where male or white individuals were hired at a significantly lower rate than 50%, just to be able to make more precise observations and to use the full volume of the collected data. For that we need new hypotheses. To investigate whether males were subjected to a significantly lower hiring rate, we utilize the following null and alternate hypotheses:

$$\begin{aligned} H_0 &: \text{Hiring Probability for male} \geq 50\% \\ H_1 &: \text{Hiring Probability for male} < 50\% \end{aligned}$$

5 Evaluation and Discussion

In order to ascertain whether white individuals were employed at a rate that was significantly below 50%, following hypotheses were formulated:

$$H_0 : \text{Hiring Probability for white} \geq 50\%$$

$$H_1 : \text{Hiring Probability for white} < 50\%$$

To test the aforementioned hypotheses, a left-tailed test is employed, with a 95% significance level. Additionally, the distribution of jobs within male, female, white, and non-white groups is examined to ascertain whether there are any underrepresented categories. For that we would use a $p_0 = 33.33\%$ and following hypotheses:

$$H_0 : \text{Proportion of people in the group that work for a specific job} \geq 33.33\%$$

$$H_1 : \text{Proportion of people in the group that work for a specific job} < 33.33\%$$

Furthermore, it is of interest to ascertain whether a particular role is significantly higher represented within the groups. To this end, a right-tailed hypothesis test for proportions is employed, with the following hypotheses being posited:

$$H_0 : \text{Proportion of people in the group that work for a specific job} \leq 33.33\%$$

$$H_1 : \text{Proportion of people in the group that work for a specific job} > 33.33\%$$

5.2 Results

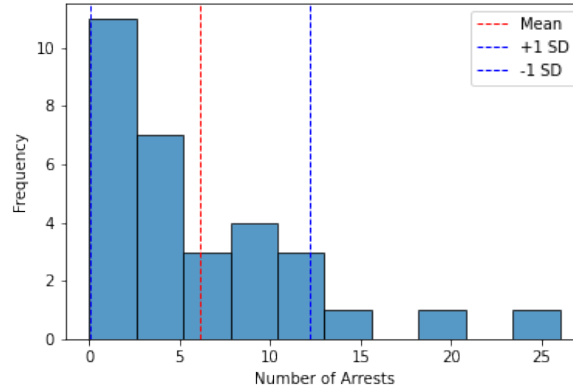


Figure 5.1: Distribution of arrests.

The statistic screenshots also include the number of times the player was arrested while playing the game. The mean for all participants was $\mu = 6.13$ arrests, with a standard deviation of $\sigma = 6.06$. The significant deviation indicates that the number of arrests varies considerably between different participants. Additionally, there are two outliers, one with 26 arrests and another with 19 arrests, as illustrated in Figure 5.1.

The data indicates that the majority of participants did not experience a high number of arrests. However, it is notable that a subset of individuals were able to play the game without an arrest. This suggests that the majority of participants may have been in a state of flow, which would have enabled them to make more unconscious decisions [19].

Upon inquiry as to whether they believe there is a hiring bias based on gender, with a mean score of $\mu = 3.45$ and a standard deviation of $\sigma = 1.31$, the survey participants appear to concur that such a bias exists. Additionally, when queried on the existence of hiring bias based on race, with a mean of $\mu = 3.68$ and a standard deviation of $\sigma = 1.37$, the participants concurred that such bias does also indeed exist. The results of the survey also indicated that the participants exhibited a strong conviction that they would refrain from judging an individual based on their gender, with a mean score of $\mu = 4.32$ and a standard deviation of $\sigma = 0.87$, or based on their race, with a mean score of $\mu = 4.29$ and a standard deviation of $\sigma = 0.94$.

Table 5.1 illustrates how many participants hired significantly fewer individuals from the following groups: "male", "female", "white" and "non-white" in the categories "Hired Workers", "Direct Selection Bias", "Manager" and "Construction Worker". In the category "Direct Selection Bias", it is demonstrated that a gender was significantly underrepresented when the player was required to select between a male and a female applicant, and that a race was significantly underrepresented when the player was required to select between a white and a non-white person. Additionally, a participant neglected to take a screenshot of this category, resulting in a sample size of only 30 participants.

	Male	Female	White	Non-White
Hired Workers	0	4 (12.90%)	0	2 (6.45%)
Direct Selection Bias	0	7 (23.33%)	2 (6.66%)	4 (13.33%)
Manager	1 (3.23%)	1 (3.23%)	2 (6.45%)	2 (6.45%)
Construction Worker	1 (3.23%)	6 (19.35%)	2 (6.45%)	0
Engineer	1 (3.23%)	1 (3.23%)	1 (3.23%)	2 (6.45%)

Table 5.1: Participants hiring probability significantly less than 50% of a group per category.

When looking at how the jobs are distributed in the gender and job categories, we also have to take into account that a participant might have distributed overall the job roles unequally. For example if a player used significantly more than 33.33% of all the employees as a construction worker, we would not count if significantly more than 33.33% of the males work as a construction worker, since this could be the result of the general unevenly distribution of the jobs.

Table 5.2 illustrates the number of participants who assigned employees within the groups of male, female, white and non-white with a significantly lower proportion of individuals than 33.33% of the total group to a particular job role. It can be assumed that participants who distributed significantly fewer individuals from a particular group to a role may have

5 Evaluation and Discussion

a preconceived notion that the group does not align with the requirements of the role in question.

	Manager	Construction Worker	Engineer
Male	0	0	1 (3.23%)
Female	1 (3.23%)	6 (19.35%)	1 (3.23%)
White	1 (3.23%)	0	1 (3.23%)
Non-White	3 (9.68%)	0	2 (6.45%)

Table 5.2: How many participants distributed significantly less than 33.33% of a demographic group to a job.

Table 5.3 illustrates the number of participants who assigned employees within the groups of male, female, white and non-white with a significantly higher proportion of individuals than 33.33% of the total group to a particular job role. It may be posited that participants who distributed a significantly higher proportion of individuals from a particular group to a role may have preconceived notions that individuals from that group possess the requisite qualities for the role in question.

	Manager	Construction Worker	Engineer
Male	0	0	1 (3.23%)
Female	5 (16.13%)	1 (3.23%)	0
White	0	1 (3.23%)	1 (3.23%)
Non-White	0	0	1 (3.23%)

Table 5.3: How many participants distributed significantly more than 33.33% of a demographic group to a job.

The most striking finding is that the majority of discriminatory behaviour occurred when applicants were required to select between a male and a female employee, as well as when allocating individuals to construction roles. A number of applicants demonstrated discriminatory attitudes towards women, whereas no such attitudes were observed towards men. These findings are consistent with the existing literature on gender representation in construction, which indicates that women are underrepresented in this field [15]. A position such as that of a construction worker is also referred to as a role that requires strength, and may unconsciously evoke the stereotype that only men are strong [10]. A connection to such a strong stereotype appears to create a bias, as evidenced by the fact that, despite the male dominance of the engineering field [11], nearly no participants exhibited bias in this category. Furthermore, an analysis of the distribution of female applicants reveals a clear bias. This suggests that women are not typically associated by some participants with construction worker roles, with a proportion of attendees allocating them to managerial positions. This aligns with observations in the real world, where women in construction are predominantly concentrated in managerial roles [6]. It

is noteworthy that there are also instances where one or two participants exhibited a tendency to hire significantly fewer male or white applicants within a given category. One potential explanation is that some participants may have been unable to reach a state of flow, wherein unconscious decisions are made and the awareness of potential biases is heightened. Players like the one with 26 arrests give the indication, that they could have lost so often, that they got frustrated and never reached this flow [19]. Alternatively, it is possible that these applicants possess a social justice orientation, which may have influenced their decisions. This orientation could have been shaped by awareness of racism and sexism, leading to choices that challenge these systems [1].

Additionally, the survey inquired whether the game encouraged participants to exercise greater caution in the future, avoiding the tendency to form judgments based on superficial characteristics such as gender or race. With an average rating of $\mu = 3.35$ and a standard deviation of $\sigma = 1.27$, the majority of participants indicated that the game had a positive impact on their inclination to avoid bias. This suggests that the game had a discernible educational value for the participants.

6 Conclusions and Future Work

The study demonstrates that despite awareness of the presence of racism and sexism in hiring practices, some players in the context of a fictional video game, in which they were required to hire applicants that were clearly labelled with “stats” indicating their capability, exhibited a bias against women. This bias appears to be linked to stereotypes associated with specific roles, such as the perception that only men are physically capable of performing tasks that require strength. While a hiring bias against women for construction worker roles was observed, this was less prevalent in the engineering profession, which is also male-dominated in reality. Additionally, the study revealed that a higher proportion of players exhibited bias against women than against men in hiring decisions. Also the game’s educational value is evident from the majority of players’ responses, which indicated that the hiring bias game encouraged them to exercise caution in judging individuals based on their appearance.

It would be beneficial for future work to consider porting the game to multiple platforms, given that it is currently only available for Android devices. The game could be developed for iOS devices and, by modifying the controls, it could also be ported to the PC. Unity provides the developer with the ability to easily change the platform on which they want to build the game, suggesting that it would only require a few modifications. Porting to multiple platforms would facilitate the testing of the hiring bias serious game with a larger sample size.

Furthermore, the game could be modified to test for differences in bias between multiple ethnicities and races, extending the scope of the study beyond the comparison between white and non-white individuals. Additionally, further potential biases could be investigated. Another documented bias in hiring is based on age. Research indicates that older individuals are less likely to be hired due to stereotypes that associate them with decreased performance and productivity [35]. By incorporating a new age category into the game, it may be possible to discern whether players exhibit a proclivity to hire only individuals who appear younger. Additionally, other serious games could be developed to assess biases in different contexts, such as renting, education, healthcare, or representation.

Bibliography

- [1] Jule Adriaans and Marie Fourré. Basic social justice orientations—measuring order-related justice in the european social survey round 9. *Measurement Instruments for the Social Sciences*, 4(1):11, Sep 2022. doi:10.1186/s42409-022-00040-3.
- [2] Android Developers. Android lollipop, 2024. Accessed: 05-07-2024. URL: <https://developer.android.com/about/versions/lollipop>.
- [3] Batu Aytemiz and Adam M. Smith. A diagnostic taxonomy of failure in videogames. In *Proceedings of the 15th International Conference on the Foundations of Digital Games*, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3402942.3402979.
- [4] Alma Barrera Yañez, Cristina Alonso-Fernandez, and Baltasar Fernández-Manjón. Review of serious games to educate on gender equality. 10 2020. doi:10.1145/3434780.3436592.
- [5] Julia Beckhusen. Administrative professionals day recognizes 2% of u.s. workforce, mostly women who earn less than average worker, 2022. Accessed: 01-07-2024. URL: <https://www.census.gov/library/stories/2022/04/recognizing-nations-administrative-professionals.html>.
- [6] BigRentz. Women in construction: The state of the industry in 2022, 2022. Accessed: 01-07-2024. URL: <https://www.bigrentz.com/blog/women-construction>.
- [7] Victor A. Cuesta Aguiar and Masaru Nakano. A model for the development of stealth serious games. In Heide Karen Lukosch, Geertje Bekebrede, and Rens Kortmann, editors, *Simulation Gaming. Applications for Sustainable Cities and Smart Infrastructures*, pages 139–147, Cham, 2018. Springer International Publishing.
- [8] Joseph L Fleiss, Bruce Levin, and Myunghee Cho Paik. *Statistical methods for rates and proportions*. john wiley & sons, 2013.
- [9] Enav Friedmann and Dorit Efrat-Treister. Gender bias in stem hiring: Implicit in-group gender favoritism among men managers. *Gender & Society*, 37(1):32–64, 2023. arXiv:<https://doi.org/10.1177/08912432221137910>, doi:10.1177/08912432221137910.
- [10] May Ling D. Halim, Dylan J. Sakamoto, Lyric N. Russo, Kaelyn N. Echave, Miguel A. Portillo, and Sachiko Tawa. Early gender differences in valuing strength. *Archives of*

Bibliography

- Sexual Behavior*, 51(4):2169–2182, May 2022. arXiv:2022Mar28, doi:10.1007/s10508-021-02185-4.
- [11] Judith A. Hall and Amy G. Halberstadt. Sex roles and nonverbal communication skills. *Sex Roles*, 7(3):273–287, March 1981. doi:10.1007/BF00287542.
- [12] Nida Hassan. Gender biases and discrimination while hiring. *Artha Journal of Social Sciences*, 18(1):13–21, 2019.
- [13] Institut international des droits de l’Homme et de la paix. Interactive game - fight against racism and discrimination. Accessed: 01-07-2024. URL: https://2idhp.eu/nos_ressources/fight-against-racism-and-discrimination/.
- [14] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan, 2020. URL: <https://arxiv.org/abs/1912.04958>, arXiv:1912.04958.
- [15] Amy King-Lewis. *Diversity and Inclusion of Women in the Construction Industry*. PhD thesis, Oklahoma State University, 2020.
- [16] Stephen C. Loftus. Chapter 13 - hypothesis tests for a single parameter. In Stephen C. Loftus, editor, *Basic Statistics with R*, pages 137–150. Academic Press, 2022. URL: <https://www.sciencedirect.com/science/article/pii/B9780128207888000262>, doi:10.1016/B978-0-12-820788-8.00026-2.
- [17] Math.NET. Math.net numerics. Accessed: 11-07-2024. URL: <https://numerics.mathdotnet.com/>.
- [18] Dr. Caroline Mitchell. Anti sexism game. Accessed: 01-07-2024. URL: <https://www.antisexism.co.uk/>.
- [19] Jeanne Nakamura, Mihaly Csikszentmihalyi, et al. Flow theory and research. *Handbook of positive psychology*, 195:206, 2009.
- [20] Ted O’Donoghue and Matthew Rabin. The economics of immediate gratification. *Journal of Behavioral Decision Making*, 13(2):233–250, 2000. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-0771%28200004%06%2913%3A2%3C233%3A%3AAID-BDM325%3E3.0.CO%3B2-U>, arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1002/%28SICI%291099-0771%28200004%06%2913%3A2%3C233%3A%3AAID-BDM325%3E3.0.CO%3B2-U>, doi:10.1002/(SICI)1099-0771(200004/06)13:2<233::AID-BDM325>3.0.CO;2-U.
- [21] Konstantinos Papangelis, Melvin Metzger, Yiyang Sheng, Hai-Ning Liang, Alan Chamberlain, and Vassilis-Javed Khan. "get off my lawn!": Starting to understand territoriality in location based mobile games. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA ’17, page 1955–1961, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3027063.3053154.

- [22] Phillip Wang. This Person Does Not Exist, 2019. Accessed: 08-07-2024. URL: <https://thispersondoesnotexist.com>.
- [23] Lincoln Quillian and Arnfinn H. Midtbøen. Comparative perspectives on racial discrimination in hiring: The rise of field experiments. *Annual Review of Sociology*, 47(Volume 47, 2021):391–415, 2021. URL: <https://www.annualreviews.org/content/journals/10.1146/annurev-soc-090420-035144>, doi:10.1146/annurev-soc-090420-035144.
- [24] Lincoln Quillian, Devah Pager, Ole Hexel, and Arnfinn H Midtbøen. Meta-analysis of field experiments shows no change in racial discrimination in hiring over time. *Proceedings of the National Academy of Sciences*, 114(41):10870–10875, 2017.
- [25] Caryl E. Rusbult and Paul A. M. Van Lange. Interdependence, interaction, and relationships. *Annual Review of Psychology*, 54(Volume 54, 2003):351–375, 2003. URL: <https://www.annualreviews.org/content/journals/10.1146/annurev.psych.54.101601.145059>, doi:10.1146/annurev.psych.54.101601.145059.
- [26] Gillian Smith, Elaine Gan, Alexei Othenin-Girard, and Jim Whitehead. Pcg-based game design: enabling new play experiences through procedural content generation. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games*, PCGames '11, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/2000919.2000926.
- [27] Birgit Stetina, Natalie Rodax, Armin Klaps, Zuzana Kovacovsky, Serkan Sertkan, and Helmut Hlavacs. Racism and sexism approached with " purpose ": Serious games as a low-threshold way to increase awareness, 03 2017. doi:10.13140/RG.2.2.15607.62883.
- [28] Unity Technologies. Unity game engine, 2022. Version 2022.3.36. URL: <https://unity.com/de/releases/editor/whats-new/2022.3.36#installs>.
- [29] Unity Technologies. Unity download, 2024. Accessed: 05-07-2024. URL: <https://unity.com/download>.
- [30] Unity Technologies. *Unity Input System Documentation*, 2024. Accessed: 09-07-2024. URL: <https://docs.unity3d.com/Packages/com.unity.inputsystem@1.8>.
- [31] Unity Technologies. *Unity Remote 5 Documentation*, 2024. Accessed: 05-07-2024. URL: <https://docs.unity3d.com/Manual/UnityRemote5.html>.
- [32] U.S. Bureau of Labor Statistics. Table 18. employed persons by detailed industry, sex, race, and hispanic or latino ethnicity, 2024. Accessed: 01-07-2024. URL: <https://www.bls.gov/cps/cpsaat18.htm>.
- [33] Vincent van de Ven, Moritz Jäckels, and Peter De Weerd. Time changes: Timing contexts support event segmentation in associative memory. *Psychonomic Bulletin & Review*, 29(2):568–580, 2022. doi:10.3758/s13423-021-02000-0.

Bibliography

- [34] Paul Williams, Keith Nesbitt, Ami Eidels, and David Elliott. Balancing risk and reward to develop an optimal hot-hand game. *Game Studies*, 11(1):1604–7982, 2011.
- [35] Marie Wilson, Polly Parker, and Jordan Kan. Age biases in employment: Impact of talent shortages and age on hiring. *University of Auckland Business Review*, 9(1), 2007.
- [36] Amy K.M. Winters, Simone G. de Waart, and Miguel Bruns. Tension-and-release: A design principle for dynamic materials. *International Journal of Design*, 16(2):1–14, August 2022. doi:10.57698/v16i2.01.
- [37] Mi Zhou, Vibhanshu Abhishek, Timothy Derdenger, Jaymo Kim, and Kannan Srinivasan. Bias in generative ai, 2024. URL: <https://arxiv.org/abs/2403.02726>, arXiv:2403.02726.