

УДК 004.94

ДИЗАЙН КАМЕРЫ В ИГРАХ И ПРИЕМЫ ЕЕ ЛУЧШЕГО ВОСПРИЯТИЯ***В. В. Иванов, А. Т. Абраменкова***

Донской государственной технической университет (г. Ростов-на-Дону, Российская Федерация)

Статья посвящена описанию работы камеры в видеоиграх и оценке ее значимости. Рассматриваются различные подходы при работе с ней. Целью авторов статьи являлось создание удобной адаптивной игровой камеры, способной дополнить погружение в игровой процесс. Проведен анализ работы камеры в играх разных жанров. Изучена работа ракурса камеры и ее восприятия в соответствии с ее положением в пространстве. Представленные приемы адаптированы под работу с внутриигровой камерой. Описаны различные приемы для улучшения работы, сформулированы математические формулы нескольких трюков. Выполнена реализация данных приемов внутри игрового движка Unity. Предложенные решения могут быть полезными при попытках добиться большего погружения игрока в игровой процесс.

Ключевые слова: гейм-дизайн, камера, правило третей, прицеливание, угол камеры, тряска камеры.

CAMERA DESIGN IN VIDEO GAMES AND TRICKS FOR BETTER PERCEPTION***Vadim V. Ivanov, Anzhelika T. Abramenkova***

Don State Technical University (Rostov-on-Don, Russian Federation)

The article describes camera operation in video games and assesses its significance. Various approaches are considered when working with it. The authors' objective was to create a convenient adaptive gaming camera that can complement the immersion in the gameplay. The paper provides the analysis of the camera operation in games of different genres. The work of the camera angle and its perception in accordance with its position in space is studied. The presented techniques are adapted to work with an in-game camera. Various techniques for improving work are described; mathematical formulas for several tricks are formulated. The implementation of these techniques inside the Unity game engine has been performed. The proposed solutions can be useful when trying to achieve greater immersion of the player in the gameplay.

Keywords: game design, camera, rule of thirds, targeting, camera angle, camera shake.

Введение. Камера — один из главных гейм-дизайнерских инструментов при создании игр. В каком-то смысле можно сказать, что это главный прибор, через который происходит интеракция пользователя с самой игрой, поэтому разработчики всегда уделяют особое внимание тому, что и как игрок должен видеть. Практически каждая созданная игра в том или ином виде старается использовать приемы и трюки внутри игровой камеры для достижения соответствующего опыта и ощущения для игрока. Создание удобной адаптивной игровой камеры, которая дополняет погружение в игровой процесс, является целью данной работы.

Материалы и методы. Перспектива и положение. В играх камера представлена не в одном виде, есть несколько различных перспектив. Банальное расположение в пространстве передает совершенно разное восприятие и атмосферу для игрока. Например, в трехмерном пространстве камера может быть статичной, динамической, от глаз персонажа, с видом сверху. Если говорить о камере в играх от 3-го лица, то важно и то, насколько близко камера расположена

к игровому персонажу. На рис. 1 показаны примеры расположения камеры с двумя разными подходами [1–6].

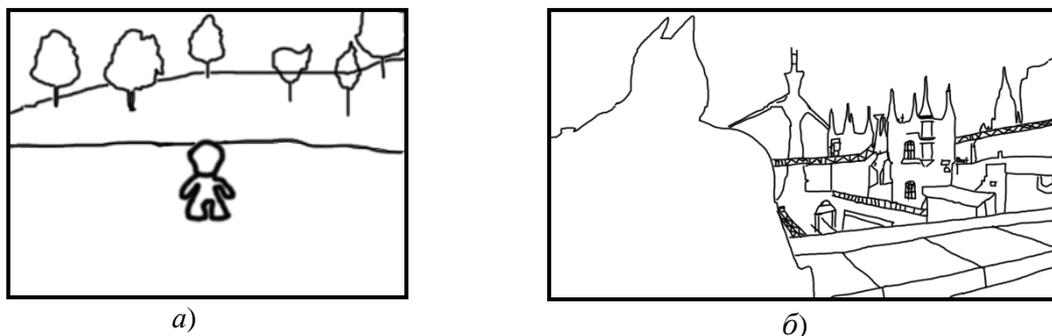


Рис. 1. Примеры положения камеры относительно персонажа в трехмерном пространстве

Как можно заметить, персонажи на рисунках занимают совершенно разное положение на экране. В трехмерных платформерах, таких как Mario 64 и Sonic Adventure (рис. 1 а), персонаж занимает не более 2 % от общего размера экрана. Это сделано для того, чтобы игрок видел большее пространство перед собой и мог легче проходить препятствия. В таких играх, как «God of war (2018)» и «Batman Arkham Asylum» расположение камеры другое (рис. 1 б), и персонаж может занимать до 50 % пространства на экране. Это сделано потому, что в данных произведениях делается большой упор на сюжетную часть игры, а такой ракурс помогает вжиться в роль протагониста и видеть мир его глазами.

В каждом проекте разная перспектива предназначена для разных целей и вызывает разные эмоции. Хорошая камера является довольно гибкой и всегда старается адаптироваться под условия, в которых находится игрок.

Тряска. Различные движения и смена ракурса камеры способствуют большему погружению игрока в процесс. Тряска камеры является одним из таких приемов. Данный прием часто используется в экшен-ориентированных играх, где зачастую нужно показать наибольшую отдачу от произошедших действий. В игре «God of war (2018)» удары персонажа сопровождаются небольшой тряской камеры, чтобы показать силу, приложенную к удару, иначе игрок хуже ощущал бы важность совершенного действия. Это для экшен-игр имеет существенное значение. В другой игре — «Tomb Rider (2013)» — можно заметить тряску камеры в эпизоде, где главная героиня плывет по реке и избегает препятствия. В данной ситуации тряска используется для того, чтобы показать силу течения реки, иначе игрок не понимал бы насколько опасна и быстра река.

Для того, чтобы реализовать этот прием математически нужно задать поворот камеры в градусах, силу тряски (в промежутке между -1 и 1) и случайное число 1 или -1 . После этого сделать такое же вычисление, только вместо градусов указать смещение камеры в пространстве. Формула для смещения выглядит следующим образом:

$$\text{OffsetX} = \text{maxOffset} * \text{SHAKE} * \text{RandomFloatNegOneToOne}(), \quad (1)$$

где MaxOffsetX — максимальное значение смещения по оси x , SHAKE — сила, с которой происходит смещение (значение указывается в промежутке от -1 и 1 включительно) $\text{RandomFloatNegOneToOne}()$ — метод, возвращающий -1 или 1 .

Формула для поворота выглядит следующим образом:

$$\text{Angle} = \text{maxAngle} * \text{SHAKE} * \text{RandomFloatNegOneToOne}() \quad (2)$$

где $MaxAngle$ — максимальный угол, на который повернется камера, остальные переменные такие же, как и в формуле выше.

Для трехмерного пространства поворот угла вычисляется для каждой из трех осей, в то время как для двумерного пространства достаточно вычислить угол лишь для одной оси [7–9].

Для более плавной тряски можно использовать алгоритм «Шум Перлина». Суть его заключается в том, что он генерирует псевдослучайный градиент для узлов сетки на основе принимаемых им чисел. В рамках этого алгоритма используется принцип преемственности — градиент каждого узла основан на значениях соседних для него узлов, то есть выходящие числа будут случайными, однако зависимыми от предыдущих.

Этот принцип работы приведет к тому, что тряска камеры будет плавней, нежели при использовании обычного алгоритма случайных чисел. Формула для смещения камеры с использованием шума Перлина выглядит следующим образом:

$$OffsetX = maxAngle * SHAKE * PerlinNoise(seed, time), \quad (3)$$

$$OffsetY = maxAngle * SHAKE * PerlinNoise(seed + 1, time), \quad (4)$$

где $Seed$ — случайное число; $time$ — время, за которое происходит генерация [10–13].

Прицеливание на врага. Одним из наиболее важных и часто применяющихся приемов для работы с камерой является возможность заострить внимание на одной цели. Впервые такая игровая механика была использована в игре «Zelda Ocarina of time», после чего стала находкой для многих экшен-ориентированных игр. Главной особенностью данной системы является то, что она делает процесс проведения боя более удобным для игрока, поскольку ему не нужно направлять удар в сторону противника, ему достаточно просто нажать на одну кнопку удара. До этого для трехмерных игр это было большой проблемой. Вдобавок это решение лучше влияет на погружение игрока, так как зачастую в настоящих боях люди заостряют внимание на одной цели за раз. Кроме того, данный прием делает действие кинематографичней из-за ракурса и акцентирования. На рис. 2 показан пример того, как выглядит прицеливание в экшен-ориентированных играх.

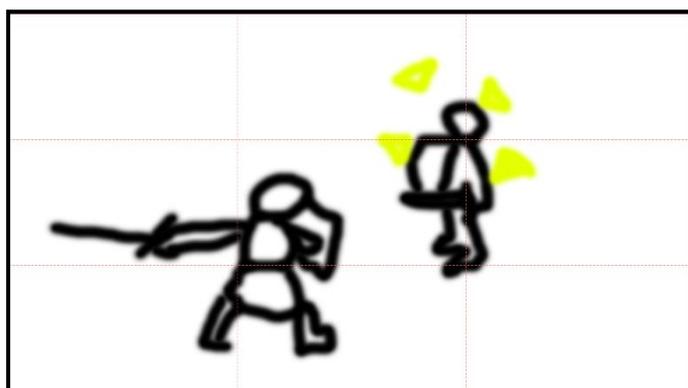


Рис. 2. Положение камеры во время прицеливания

Однако, как можно видеть на рис. 2, игра использует еще один не менее важный композиционный прием — правило третей. Суть его заключается в том, чтобы придать бóльшей красоты кадру путем разбивания его на квадраты 3×3 таким образом, чтобы на их пересечении находились объекты, на которых фокусируется взгляд человека. Это решение несет не только эстетический характер, но и гейм-дизайнерский. Такое положение камеры дает бóльший обзор для игрока, так как управляемый им персонаж находится в левом углу экрана, в то время как все остальное пространство свободно для взора [14–16].

Математически реализация этого приема выглядит следующим образом:

$$V = B + BA1 * (|BA| + Distance), \quad (5)$$

где B — это координаты цели; A — координаты объекта, который нацелен на B ; $BA1$ — направленный вектор из точки B в $A1$; $|BA|$ — длина вектора BA .

$A1$ это сумма векторов A и $OffSet$, равная:

$$A1 = A + OffSet, \quad (6)$$

где $OffSet$ — отступ от вектора A ; $Distance$ — дополнительное расстояние от управляемого объекта.

После этого строится вектор VB . Положение координат камеры зависит от положений как управляемого персонажа, так и конечной цели. При их смещении происходит и смещение положения камеры.

Результаты исследования. Далее нужно протестировать реализацию на явном примере внутри игрового движка. На рис. 3 и 4 показана реализация формулы прицеливания внутри игрового движка Unity на языке C#.

```

cameraOffSet = player.transform.position + OffSet; // A1
fromEnemyToPlayer = cameraOffSet - enemyTarget.transform.position; // BA1
fromEnemyToPlayerMagnitude = (player.transform.position - enemyTarget.transform.position).magnitude; // BA

cameraPosition = enemyTarget.transform.position + (fromEnemyToPlayer).normalized
                * ((player.transform.position - enemyTarget.transform.position).magnitude + distance); // V

if (lockOn == true)
{
    transform.position = cameraPosition;
    transform.LookAt(enemyTarget.transform);
}

```

Рис. 3. Пример реализации формулы захвата цели в виде кода внутри Unity.

На рис. 4 показано как эта реализация выглядит на игровой сцене внутри движка Unity.

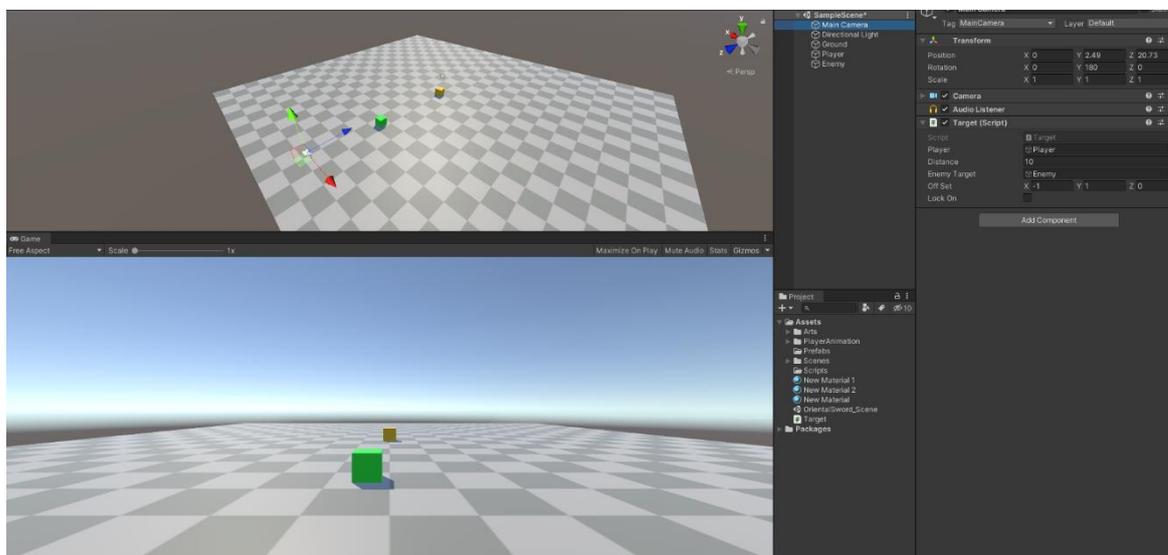


Рис. 4. Пример работы прицеливания внутри игрового движка Unity

Положение и ракурс камеры были выбраны ближе к варианту на рис. 1 а, поскольку таким образом открывается больше пространства для осмотра сцены с вражеским объектом. Зеленый квадрат — это управляемый персонаж, а желтый — вражеский. При включении нужного триггера камера акцентирует свое внимание на враге, при этом держит игровой объект в поле видимости. С помощью настраиваемых параметров `offset` и `distance` можно сделать такой ракурс камеры, чтобы

он следовал правилу третей и это непосредственно позволит проверить действительность этого правила на деле. При соприкосновении с вражеским объектом происходит тряска камеры, реализованная с помощью обозначенной выше математической формулы, которая показывает, что игровой объект получил урон.

Заключение. Все вышеперечисленные приемы работы с камерой составляют лишь незначительную часть от большого количества различных трюков с камерой. Описанные в статье примеры рассматривались на примерах трехмерных проектов. Однако для двухмерных вариаций существуют свои уникальные гейм-дизайнерские приемы, например, плавное прокручивание камеры при перемещении и другие. Для каждой игры существуют свои собственные правила того, как должна работать камера. Приемы, которые работают в одном проекте, напрочь могут сломать восприятие другого. Поэтому при работе с камерой разработчики четко осознают какой опыт они хотят передать и приходят к методам и приемам, дополняющим этот опыт. Тем не менее, представленные выше решения по работе с камерой можно использовать в большом количестве проектов для непосредственного улучшения игрового опыта.

Библиографический список

1. Espineli, M. The Challenges Of Remaking A Horror Classic Like Resident Evil 2, 28.06.2018 / M. Espineli // gamespot [сайт] — URL : <https://www.gamespot.com/articles/the-challenges-of-remaking-a-horror-classic-like-r/1100-6459995/> (дата обращения: 14.05.2022).
2. Haigh-Hutchinson, M. Fundamentals of Real-Time Camera Design, 1.4.2009, Retro Studios inc / M. Haigh-Hutchinson // ubm-twvideo01.s3.amazonaws.com [сайт] — URL : https://ubm-twvideo01.s3.amazonaws.com/o1/vault/gdc05/slides/GD_Haigh-Hutchinson_FundamentalsReal-TimeCameraDesign2.pdf (дата обращения: 14.05.2022).
3. Naftis, M., Tsatiris, G., Karpouzis, K., How Camera Placement Affects Gameplay in Video. University of Otago, 8.09.2021, SAE Creative Media College, National Technical University of Athens, Panteion Univ. of Social & Political Sciences / M. Naftis, G. Tsatiris, K. Karpouzis // arxiv [сайт] — URL : <https://arxiv.org/pdf/2109.03750.pdf> (дата обращения: 14.05.2022).
4. Sivers, Yu. Design Tips for In-Game Character Proportions, Kongregate, 16.05.2016 / Yu. Sivers // blog.kongregate [сайт] — URL : <https://blog.kongregate.com/design-tips-for-in-game-character-proportions/> (дата обращения: 14.05.2022).
5. Евгений Кучерявый, Камера в играх: как рассказать историю и показать мир, SkillBox Media, 27.9.2019 / skillbox [сайт] — URL: https://skillbox.ru/media/gamedev/kamera_v_igrakh_kak_rasskazat_istoriyu_i_pokazat_mir/ (дата обращения: 14.05.2022).
6. Tyler, D. Tips for In-Game Sizing & Scale: Consistent Sizing System, Character Sizing and more, Gamedesigning, 24.6.2021 / D. Tyler // gamedesigning [сайт] — URL: <https://www.gamedesigning.org/learn/game-scale/> (дата обращения: 14.05.2022).
7. Squirrel (Brian) Eiserloh, Math for Game Programmers: Juicing Your Cameras With Math, Game developer Conference, 14.3.2016 / Squirrel (Brian) Eiserloh // mathforgameprogrammers [сайт] — URL: http://www.mathforgameprogrammers.com/gdc2016/GDC2016_Eiserloh_Squirrel_JuicingYourCameras.pdf (дата обращения: 14.05.2022).
8. Strachan, D. Analysis of Screenshake types, Easy Difficulty Game Development / D. Strachan // davetech [сайт] — URL: <http://www.davetech.co.uk/gamedevscreenshake> (дата обращения: 14.05.2022).

9. Taskas, S. What makes for good visual game juice?, 24.4.2020 / S. Taskas // medium [сайт] — URL: <https://medium.com/swlh/what-makes-for-good-visual-game-juice-e63cb8ba2068> (дата обращения: 14.05.2022).
10. Gustavson, S. Simplex noise demystified, Linköping University, Sweden (stegu@itn.liu.se), 2005.03.22 / S. Gustavson // weber [сайт] — URL: <https://weber.itn.liu.se/~stegu/simplexnoise/simplexnoise.pdf> (дата обращения: 14.05.2022).
11. Shiffman, D. Khan Academy, Perlin noise, 2012 / D. Shiffman // khanacademy [сайт] — URL: <https://www.khanacademy.org/computing/computer-programming/programming-natural-simulations/programming-noise/a/perlin-noise> (дата обращения: 14.05.2022).
12. Biagioli, A. Adrian's soapbox, Understanding Perlin Noise, 9.8.2014 / A. Biagioli // adrianb [сайт] — URL: <https://adrianb.io/2014/08/09/perlinnoise.html> (дата обращения: 14.05.2022).
13. Tulleken, H. How to Use Perlin Noise in Your Games, Dev.mag, 25.4.2009 / H. Tulleken // devmag [сайт] — URL: <http://devmag.org.za/2009/04/25/perlin-noise/> (дата обращения: 14.05.2022).
14. Rowe, C. The Golden Ratio, ChasRowe fine art, 13.2.2014 / C. Rowe // chasrowe [сайт] — URL: <http://www.chasrowe.com/howTo/golden.html> (дата обращения: 14.05.2022).
15. Doming, M. G. The Rule of Thirds: Know your layout sweet spot, 2021 / M. G. Doming // interaction-design [сайт] — URL: <https://www.interaction-design.org/literature/article/the-rule-of-thirds-know-your-layout-sweet-spots#:~:text=What%20Is%20the%20Rule%20of,be%20used%20for%20the%20design> (дата обращения: 14.05.2022).
16. Правило третей и золотое сечение — как это работает / Фотосклад [expert]: [сайт]. — URL: <https://www.fotosklad.ru/expert/articles/vse-cto-vam-nuzno-znat-pro-zolotoe-secenie/> (дата обращения : 14.05.2022).

Об авторах:

Иванов Вадим Вячеславович, студент кафедры «Медиатехнологии», Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), maskanda54@yandex.ru

Абраменкова Анжелика Тамерлановна, старший преподаватель кафедры «Медиатехнологии», Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), [ORCID](https://orcid.org/a.t.abramenkova), a.t.abramenkova@mail.ru

About the Authors:

Ivanov, Vadim V., Student, Media Technology Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), maskanda54@yandex.ru

Abramenkova, Anzhelika T., Senior Lecturer, Media Technology Department, Don State Technical University (1, Gagarin sq., Rostov-on-Don, 344003, RF), [ORCID](https://orcid.org/a.t.abramenkova), a.t.abramenkova@mail.ru