



**TARGEM GAMES**

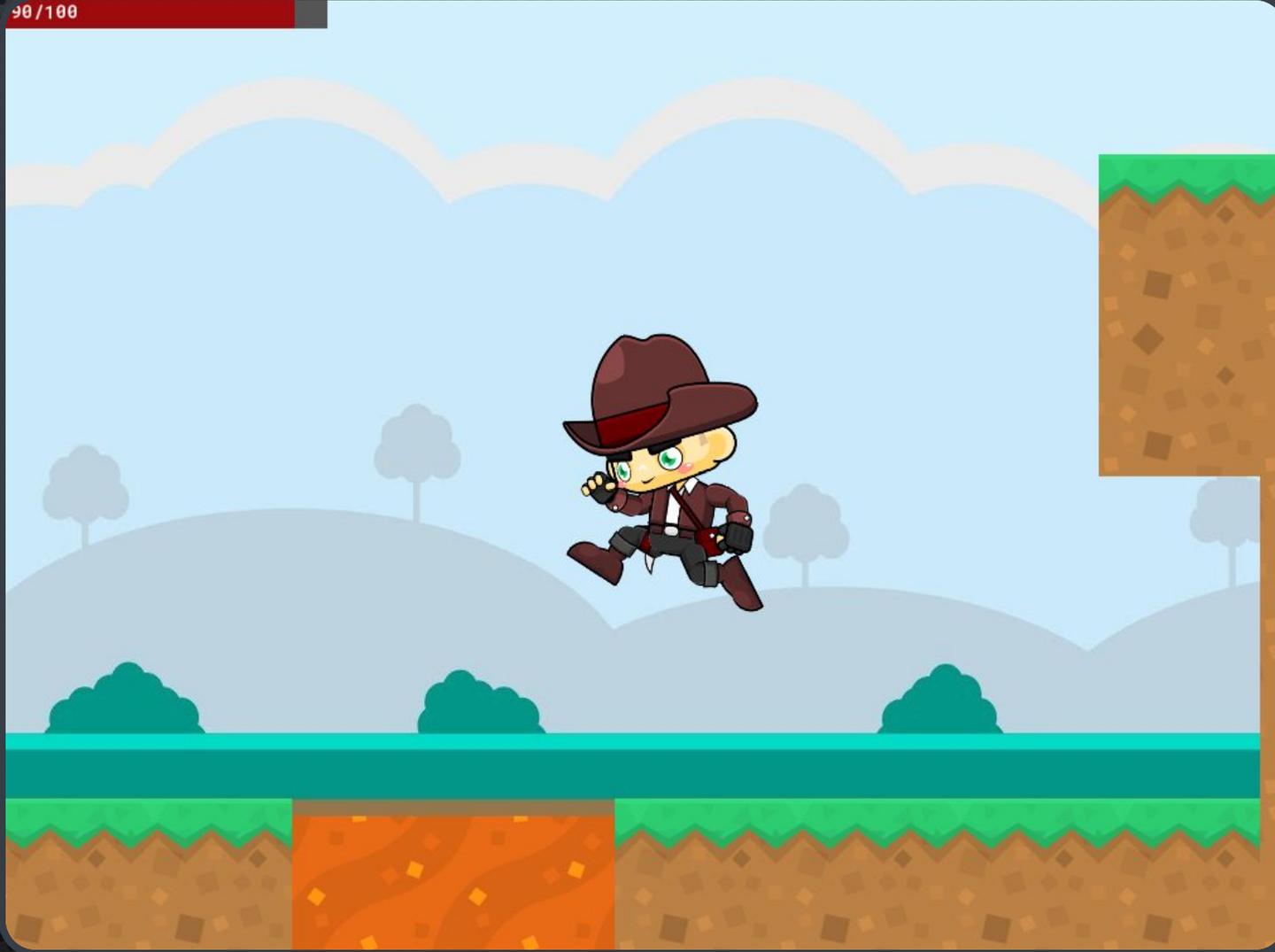
# РАЗРАБОТКА ИГР

ПЛАТФОРМЕР И ФИЗИКА ДРУГИХ МИРОВ



# ФИЗИКА В ИГРАХ

90/100



# БАЗОВЫЙ ФИЗИЧЕСКИЙ ДВИЖОК

# СИСТЕМА СТОЛКНОВЕНИЙ

- в `SIMPLEBROADPHASE` - ПРОСТОЙ ПЕРЕБОР ВСЕХ ПАР  $O(N^2)$ .
- в `AXISWEEPER3` РАБОТАЕТ В ОГРАНИЧЕННОМ ПРОСТРАНСТВЕ. ВЫ ЗАДАЁТЕ `AABB` КОРОБКУ, ВНУТРИ КОТОРОЙ ПРЕДПОЛАГАЕТСЯ БУДУТ НАХОДИТЬСЯ ВСЕ ОБЪЕКТЫ.
- в `DVWBROADPHASE` ОРГАНИЗУЕТ ВСЕ ОБЪЕКТЫ В ДВУХ ДРЕВОВИДНЫХ СТРУКТУРАХ, ОДНО ДЛЯ СТАТИЧЕСКИХ ОБЪЕКТОВ, ДРУГОЕ ДЛЯ ДИНАМИЧЕСКИХ.

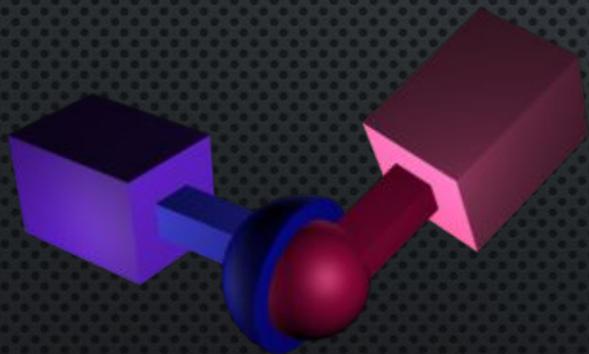


RAGDOLL

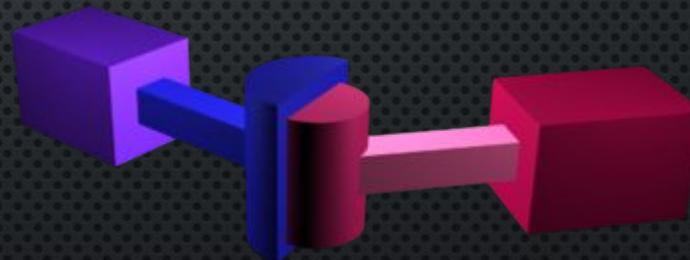


# JURASSIC PARK: TRESPASSER

# ТИПЫ СОЕДИНЕНИЙ В RAGDOLL



Шариковое

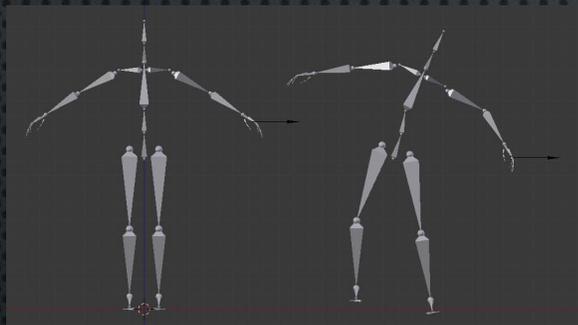


Шарнирное

# RAGDOLL

Технология ragdoll работает, основываясь на принципах алгоритма Физерстоуна и пружинно-амортизаторных контактов

Инверсная кинематика  
(Halo: Combat Evolved)



Метод Верле  
(Hitman: Codename 47)

$$Pos_{Новая} = Pos_{Текущая} + (Pos_{Текущая} - Pos_{Старая}) * \Delta t + A * \Delta t^2$$

$$Pos_{Старая} = Pos_{Текущая}$$

Смешанный Ragdoll  
(Halo 2)



Процедурная анимация  
NaturalMotion



# НЕДОСТАТКИ СИСТЕМЫ RAGDOLL В ИГРАХ

- Пальцы и мелкие составляющие не анимируются.
- Упрощенные суставы
- Упрощенная модель столкновений
- RAGDOLL часто является виновником многих багов

# КАК РЕАЛИЗОВАТЬ ГАЗ В ИГРАХ



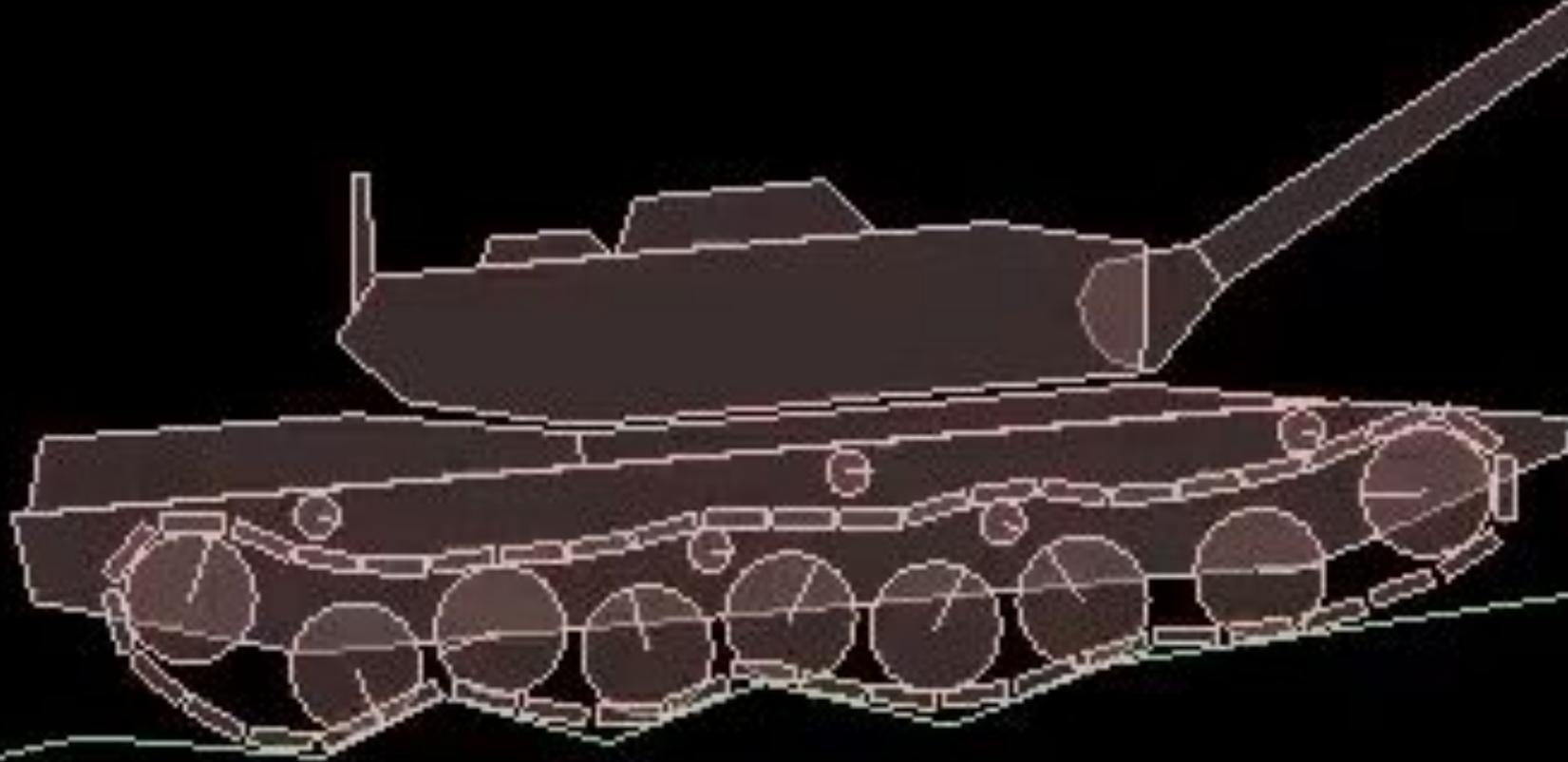
СИСТЕМА ЧАСТИЦ



Цельные текстуры



DAYS GONE. ЗОМБИ, КАК ЖИДКОСТЬ  
ИЛИ ГАЗ



GMS 2 И BOX 2D

# ВАЖНЫЕ АСПЕКТЫ

- Подсистема физики отделена от основной структуры **GMS**, то есть она не обязательно завязана на скорости комнаты и порядке шагов, как всё остальное.
- Физика заменяет многие обычные функции экземпляров. Например, вместо скорости и направления у экземпляров есть физическое тело, на которое нужно воздействовать силой или импульсом, чтобы перемещать его по игровому миру. К этому нужно привыкнуть, поэтому, прежде чем интегрировать физику в свой проект, поэкспериментируйте с ней, чтобы понять, как всё работает.
- Следите за количеством экземпляров, их столкновений и групп столкновений. Обработка физики требует интенсивных расчётов, поэтому не ждите, что вы сможете создавать тысячи экземпляров, и всё будет работать нормально.
- При определении столкновений используйте родительские объекты насколько это возможно. Например, если у вас пять разных объектов-стенок, не проверяйте столкновения с ними по отдельности, а создайте один абстрактный объект, назначьте его родителем всех стенок и проверяйте столкновения с родителем.
- Старайтесь не перемещать экземпляры, минуя физические функции. Изменить координаты экземпляра вручную возможно и в некоторых случаях даже необходимо, но этого следует избегать, потому что результаты могут быть непредсказуемыми, особенно если пытаться разрешить столкновения.
- На разных платформах поведение физики может незначительно отличаться из-за различной точности операций с плавающей запятой, однако каждая версия сохраняет постоянство от запуска к запуску.
- Чтобы предотвратить нестабильность симуляции физики, **Box2D** ограничивает предельные значения, на которые тело может переместиться и повернуться за одно обновление.

# ПОЛЕЗНЫЕ ССЫЛКИ ДЛЯ ТЕХ КТО ХОЧЕТ БОЛЬШЕ УЗНАТЬ:

- [HTTPS://GAMEDEVELOPMENT.TUTSPUS.COM/TUTORIALS/HOW-TO-CREATE-A-CUSTOM-2D-PHYSICS-ENGINE-THE-BASICS-AND-IMPULSE-RESOLUTION--GAMEDEV-6331](https://gamedevelopment.tutsplus.com/tutorials/how-to-create-a-custom-2d-physics-engine-the-basics-and-impulse-resolution--gamedev-6331)
- [HTTPS://WILDBUNNY.CO.UK/BLOG/2011/04/06/PHYSICS-ENGINES-FOR-DUMMIES/](https://wildbunny.co.uk/blog/2011/04/06/physics-engines-for-dummies/)



# ДОМАШНЯЯ РАБОТА

- Гейм-дизайн:
  - Придумать и реализовать боевую систему и врагов
- Программирование:
  - Создать платформер.
  - ТРЕБОВАНИЯ:
    - 1) Должны быть реализованные базовые физические механики (Прыжки, столкновения, отталкивания)
    - 2) Наличие врагов
    - 3) Возможность сражаться
    - 4) Наличие начала и конца
    - 5) Наличие бонусов в игре

СПАСИБО ЗА ВНИМАНИЕ)  
ВЫ ПРЕКРАСНЫ)